# A Model for Domain Centered Knowledge Discovery in Databases

Robert Redpath
School of Computer Science and Software Engineering
Monash University
Caulfield East, Victoria
Australia
*robert.redpath@infotech.monash.edu.au*

Bala Srinivasan
School of Computer Science and Software Engineering
Monash University
Caulfield East, Victoria
Australia
*bala.srinivasan@infotech.monash.edu.au*

*Abstract* – **It is well recognised that Domain Knowledge can support the KDD process. There has been much work in particular on how pre-processing of data can be supported by Domain Knowledge. There has been less work on how Domain Knowledge can assist in the latter steps of feature selection, algorithm selection, results evaluation and results interpretation in the KDD process. This paper will review the current activity and argue that to successfully automate the KDD process a number of approaches need to be combined. In particular Domain Knowledge needs to be captured so that it can support the KDD process at a number of stages. In order to do this Domain Knowledge will need to be considered in both a broad sense to permit case based approaches and a narrower sense for application of knowledge about particular datasets and attributes. This Domain Knowledge will be able to support a partially automated KDD system where manual intervention will still be permitted as required. An architecture for a KDD process supported by domain knowledge is proposed that addresses the issues raised and future directions are indicated.**

## I. INTRODUCTION

Data Mining has gained acceptance as a term to describe the automated search for patterns of interest in large collections of data. In the research community it is considered as just one of the steps in the larger Knowledge Discovery in Databases process (KDD process). After data selection and the transformation of the data into a form suitable for analysis, the data-mining step is where an algorithm is applied to the data in order to produce a model that will describe the behavior of the data in some way and hopefully the model will be useful in fulfilling the strategic aims of those initiating the analysis. A number of people have suggested different versions for the major steps required in the KDD process. Brachman and Anand [1] provide an excellent description of the process as do Adriaans & Zantinge [2]. The points of similarity between the approaches are greater than the differences. There is general agreement on a number of the steps even if the detail of how to carry out these steps varies. Most suggested methodologies include data selection, data transformation, data-mining and results evaluation, typically all occurring in a feedback loop. In 1999 a consortium including SPSS and others published a suggested standard process known as CRoss-Industry Standard Process for Data Mining (CRISP-DM) [3]. This was done, among other reasons, in response to a perceived need to have an industry, tool, and application neutral process that would provide a point of commonality for the introduction of new ideas and applications in an environment where the KDD process is lengthy, complex and has many variations in approach to the individual parts of that process. CRISP-DM has major steps of business understanding, data understanding, data preparation, modeling, evaluation and deployment. The modeling step is where an algorithmic approach is chosen and applied and this is what is referred to as the specific step of data mining in the KDD process by others[2].

Even with some agreement on the steps to be carried out, every KDD process is, at present, a one-off exercise with many decisions being made by the expert in data mining (the data analyst) typically after consultation with the expert in the domain of interest (the user of the knowledge). It is the capturing of the domain knowledge that is one of the major factors that permits decision making at the various steps as the KDD process proceeds.

The purpose of this paper is to review the various elements that will allow a partial automation of the KDD process and show how those elements can be brought together in a coherent fashion. A model is proposed that exploits domain knowledge as the key meta-data to guide and optimize this partially automated KDD process. It is shown that to obtain that domain knowledge a step of domain knowledge acquisition must begin the KDD process. This domain knowledge can then be formally categorized for use in the subsequent steps and also to guide the path taken through the processing alternatives.

Section 2 reviews the role of domain knowledge in each step of the KDD process. Attempts to automate or partially automate the KKD process, of necessity, often involve the capture and use of domain knowledge. Some researchers take categorization of domain knowledge as the starting point in their attempts to incorporate domain knowledge into an automated KDD process. The efforts of a number of researchers to categorize domain knowledge, with the aim of capturing it, are summarized. The role of the user in the process of data analysis that is carried out in the KDD process is then reviewed. In section 3 a discussion of the use of domain knowledge in the KDD process steps of data selection, algorithm choice and results interpretation is carried out and the implications for the design of data mining tools is indicated. A model for a proposed KDD workbench that incorporates the capture and employment of domain knowledge will be described in section 4.

A review of the development of and approaches to the use of Domain Knowledge cannot be discussed in isolation from other issues, such as integration of a number of data mining tools, visualization of data and knowledge and automated support of the user via, for instance, workbench approaches. So this paper in some senses reviews the development of automation of the Knowledge Discovery

process in general with an emphasis on domain knowledge and its role in the process.

## II. APPROACHES TO THE USE OF DOMAIN KNOWLEDGE IN THE KDD PROCESS

### *A. The Use of Domain Knowledge at All Steps in the KDD Process*

It is recognised by Kopanas et al. [4] that there is a role for Domain Knowledge in all stages of the KDD process, but that other than in pre-processing and data transformation its' role is not well explored by data mining researchers. They outline, in their view, the use of domain knowledge in every stage of the KDD process. Examples demonstrate how the domain expert is needed to help *define the problem* by e.g. giving business rules on what a failed transaction is or what is considered a problem customer; to assist in the creation of the target dataset by, e.g. defining the structure of the data and the semantic value of the data attribute values. They also help in the *data pre-processing and data transformation* by; (1) eliminating irrelevant attributes; (2) inferring abstract values from multiple base values; (3) determining missing values; (4) defining suitable time scales for observation periods; (5) supporting data reduction by sampling and transaction elimination. (The use of domain knowledge in the step of data pre-processing has been explored by a number of researchers; see [5, 6].) The domain expert also assists in the *feature selection and algorithm selection* step by assessing what is suggested by the automatic techniques (such as discriminant analysis). The domain expert also at this step verifies the success of earlier steps by assessing the impact of data selection on the model produced. Considering, in particular at this step, the choice of data mining task and algorithm selection; how the algorithm is chosen is not explored. Then during the *evaluation and interpretation of the learned knowledge* step the domain expert provides criteria for assessment of the results based on the business objectives. Kopanas et al. [4] also describe a final phase called *fielding the knowledge base*. In this step the domain expert combines learned knowledge with the existing approaches to incorporate it into the operational decision support systems of the business.

Interesting work is being carried out by a research group sponsored by the Information Societies Technology Programme with links to Dortmund University and is known as the Mining Mart project [7]. As part of this large and ambitious project one of the deliverables is detailed specification of how Domain Knowledge relates to the decision making at all the stages of the KDD process [8]. The list of what constitutes Domain Knowledge is extensive and includes information not considered by other researchers. Considering the data mining stage in the KDD process they present a different and extended version of how this is done. They list a number of Mining Process Decisions (MPDs). For example the particular decision concerning the data mining stage is simply known as *Analysis Technique* and the mining process decision (MPD) is stated to be the selection of the analysis technique. This is impacted by a number of Domain Knowledge Elements (DKEs), these being the Causal Model, the Design Pattern and the Analysis Paradigm. The Causal Model explores the cause and effect relationships between events and variables and provides information that permits the separation of variables into indicative or explanatory classifiers and target or output variables. There may be more than one version of the Causal Model, as the KDD process is trying to establish such relationships in any case. A Design Pattern is a general approach to the test design or experimental setup and includes, for example, cross-validation, extrapolation, modeling and forecasting (modeling is a sub step in a number of other design patterns). The Analysis Paradigm is what is considered, by the domain expert, as the primary task of the KDD exercise. There are a number of lists of the primary data mining tasks and they generally include classification, regression, association and clustering among others [9]. How the domain expert chooses one of these is not explained [8] but it would depend on the business objectives and the strategic aims of the data analysis. So to a degree this it has all become rather circular logic and the decision as to what algorithm (e.g. neural network or decision tree) to use depends on the choice of analysis paradigm (in this case e.g. classification could be done by a neural network or a decision tree) but no guidance as to how to make either choice is given; the domain expert just decides. One must conclude that human intervention is still vital for task determination and to fully automate or, even just, to suggest possibilities in some automated way is a difficult task.

Regarding the pre-processing step, Mining Mart considers how both domain knowledge (as do many other researchers [5, 6]) and previous cases can be used to guide the pre-processing at the discretion and under the control of the user. They indicate that with real examples they have reduced pre-processing time by up to 20%, of what it would have been, by using their techniques. Given that pre-processing is one of the more difficult and slowest steps this is very significant.

### *B. Classification of Domain Knowledge for KDD*

Donhoe and Rendell[10] provide a classification of what is essentially Domain Knowledge that they refer to as Fragmentary Knowledge. As it is a thoughtful and comprehensive classification, each of the types is defined below.

Relevance Knowledge: A group of features is relevant to a particular goal class or intermediate concept; e.g. features such as body style, make, no-of-doors are relevant to the intermediate concept family car; horsepower, body style, make and aspiration relate to the intermediate concept sporty car. Constructed features composed of a group of relevant (and related) features are more likely to be useful.

Support Knowledge: A feature is known to support a particular goal class or intermediate concept when it has a particular value; e.g. auto style is relevant to the concept family car and it also supports family car when it has the value station wagon.

Correlation Knowledge: A continuous version of support knowledge; e.g. cash level is correlated (positively) to company success; long term debt is correlated (negatively) to company success; so cash subtracted from long term

debt is meaningful. If cash is added to long term debt the result is not meaningful. Constructed features correlated to something important (e.g. company success) should be searched first.

Contingency Knowledge: The effect of one feature is contingent upon the value of another feature. It augments other types of knowledge; e.g. clouds are negatively correlated to temperature if it is day time and clouds are positively correlated to temperature if it is night time. (So whether it is day or night is the contingent knowledge). If contingency is absent then the features are independent and this is useful knowledge also.

Normalization Knowledge: (Similar to correlation knowledge), a single feature is correlated to multiple intermediate concepts (some relevant and some not). The goal is to normalize a feature to eliminate correlations to irrelevant intermediate concepts; e.g. net income is correlated to success and size (e.g. for a large company $1 million is low but for a small business $1 million is high). So net income should be normalized to eliminate correlation to size.

Proximity Knowledge: Indicates a group of features that represent objects or events that are in close physical, temporal or functional proximity (the logic is that this will increase the likelihood they will interact); e.g. The questions a physician asks a patient and the test then performed on the patient are likely to be related. Features derived from closely linked components are more likely to interact with each other than with other components.

Inheritance Hierarchies: Shows features that can be grouped based on similarities. A grouping of features, under a class label drawn from the hierarchy, is more likely to yield interactions. Constructed features composed of features closely related in the inheritance hierarchy should be favoured over arbitrary combinations; e.g. city-miles per gallon and highway-miles per gallon can be grouped as fuel usage.

Dimensional Analysis: Knowledge of a features' dimensions can make certain feature combinations illegal. (If the units in which the feature is measured do not match it is illegal to combine them); e.g. a feature with unit dollars should not be added to a feature with unit dollars per year.

Anand et al. [11] suggest a catergorisation of information provided by the user into Domain Knowledge and Bias Information. They separate Domain Knowledge into three classes being Hierarchical Generalisation Trees, Attribute Relationship Rules and Environment Based Constraints. Bias information includes attributes of interest, which attributes are interesting as antecedents and which as consequents (in discovered rules). Bias information is included under the general heading of Domain Knowledge by some other researchers, see [8]. Anad views domain knowledge as being useful in generalizing attribute values to reveal patterns not otherwise visible and for constraining the search space as well as the rule space. The categorization of domain knowledge is demonstrated with a rule discovery approach without consideration as to how domain knowledge would be used with other modeling approaches or why rule discovery has been chosen as the modeling approach.

### *C. User Centered KDD Using Domain Knowledge*

Yoon et al. [12] put the case that due to the large, and growing, volumes of data the user needs assistance in focusing their search before initiating discovery. They propose an outline for a system that captures three kinds of domain knowledge in order to constrain the search space and to modify queries so that they may run more efficiently. The three types of domain knowledge are (1) Interfield domain knowledge; e.g. If type="four wheel drive" then price > 30000; (2) Category domain knowledge; e.g. If (age >= 45) and (age < 65) then age=mid-old; (3) Correlation domain knowledge; e.g. If education, position, and income are correlated they can be represented as a set {education, position, income}. By considering only query driven approaches to analyse the data the three types of domain knowledge provide a limited but practical method to demonstrate the retrieval of domain knowledge and the modification of queries so that they can be processed more efficiently.

Han et al. [13] make a persuasive argument for human involvement in the knowledge discovery process, suggesting that the computer should do what it does best, for example, handling large data volumes and searching that data or aggregating it; and let the user do what they do best, for example, specifying the current mining session focus. To achieve this, an approach termed constraint based mining is outlined. The constraints can be considered to embrace some of the strategic objectives of the analysis as well as domain knowledge. Five categories of constraint are put forward

- knowledge constraints which express strategic objectives of the analysis expressed by them as the *knowledge to be mined.* (e.g. concept description, association, classification, prediction, clustering, anomalies)
- data constraints- a selection by the user of the data to be mined
- dimension/level constraints – these confine the dimensions(attributes) and level(in a hierarchal classification of the data value classes)
- rule constraints – concrete constraints on the rules to be mined
- interestingness constraints which make use of statistics to put measures on discovered patterns and specify ranges for these measures which indicate usefulness or interestingness.

Another area where domain knowledge can have an impact is in the choice of model. Domingos [14], in discussing the application of Occam's razor to model choice, takes the position is that it is correct (and uncontentious) to choose the simpler model if they have the same generalization error. Often, though, the incorrect application of Occam's razor is made and that incorrect application can be defined as, *if training-set error is the same for two models then the simpler model should be preferred because it is likely to have a lower generalization error.* This leads to over-fitting of models to the training-set data and reduces the flexibility of the model with new data. He suggests a better way to avoid over-fitting and retain flexibility is to

constrain the knowledge discovery process (and thus the choice of model) by the use of domain knowledge. The user provides the knowledge that can be useful even if it is only weak knowledge or general assumptions. Domingos [14] comprehensively reviews the literature that indicates how domain knowledge can be used to constrain knowledge discovery; typically by constraining the form of the discovered rules. Examples of how domain knowledge can constrain discovery are cited for neural networks, rule induction, inductive logic programming, propositional rule learning, and association discovery. Domingos also points out how domain knowledge can improve accuracy and speed by reducing the search space and the comprehensibility of the model by making the results of the induction consistent with previous knowledge. This is a strong argument for the use of domain knowledge, and thus for the user, to assist in model selection and it is consistent with the approaches suggested by others for example Han et al. [13].

Ho et al. [15] suggest a user centered KDD process where the Domain Knowledge is not held in some persistent form but rather the user employs their Domain Knowledge to make decisions at various points in the process. With regard to choice of technique the architecture of their discovery system D2MS has available both rule based approaches and association analysis. These are both unsupervised techniques. Within this restriction the user decides on a plan (defined as an ordered list of algorithms) that when executed produces a model. The user is able to generate a number of plans, and thus models. They then use their Domain Knowledge to select the most interesting models supported by appropriate visualizations of the models generated.

Another system of interest called RuleViz [16] proposes an interactive model that visualizes the entire KDD process. The user assists in the navigation through the search space and is supported by visualizations of the original data, the reduced data, data pre-processing, rule discovery and rule presentation.

A commercially available system developed by Accrue Decision Series[17] partially automates a number steps in the KDD process and in particular Bechers et al. [18] describe how in the exploratory data analysis automated assistance is given to attribute selection through four steps (1) identification of inappropriate and suspicious attributes; (2) selection of the most appropriate representation; (3) creation of derived attributes; (4) choice of the optimal subset of attributes. This is done by a combination of heuristic approaches and algorithmic approaches supported by manual intervention by the analyst who can provide domain specific information. The domain knowledge is not persistently held by the system.

## III. CONSIDERATIONS WHEN APPLYING DOMAIN KNOWLEDGE TO THE KDD PROCESS

Whilst the emphasis of this paper is on Domain Knowledge for support of the KDD process a holistic approach is required. Debate occurs on how interactive or automated data mining tools should be from a human computer interaction point of view [19] and also on other aspects of the task such as the use of visualization. All these issues are important and must be considered to achieve the objective of improved data mining tools with less need for the expert in knowledge discovery.

### *A. Interestingness of Data*

The process needs to allow a feedback loop through the steps. For example interestingness of data can guide the analysis and identify the interesting portions of a dataset for subsequent closer study[20, 21]. Palpanas in particular discusses how to find data instances that are interesting to the user and those that are not. Interestingness could also be generated by the discovery system by identifying trends then recognising data instances not following the trends as interesting. Interestingness can also be thought of as the distance of data from the beliefs of the domain knowledge expert[22].

### *B. Domain Knowledge to Support Result Interpretation*

Pazzani[23] notes that comprehensibility and interestingness are approached in a very loose fashion. Counter-intuitive results or non-sensical results are often presented for consideration. He suggests that insights could be gained from cognitive psychology into how to make results more comprehensible and useful to the user. He suggests three particular aspects of learning that have an impact on the user. These are; (1) consistency with prior knowledge, i.e. if a number of correct models present themselves then one should be chosen that is consistent with the users prior knowledge; (2) consistent contrast, i.e. people prefer category representations that use attributes that have values clearly different between the categories; (3) global biases, i.e. there is a bias for categories based on rules that have a number of attributes each of which individually contain attribute values (when considered alone) that could place an instance in that category. It is noted that tree and rule learning systems do not respect this and thus sometimes lead to counter-intuitive rules.

### *C. Domain Knowledge as Strategic Direction*

The definition of domain knowledge needs to be broader than that used by many researchers if the choice of data mining task (or technique such as clustering or classification or prediction) is to be supported at all. There is a requirement for knowledge of the goals of the analysis at a strategic level (this is what Han/Lakshaman/Ng call bias constraints[13]) as well as lower level knowledge about the behaviors of and relationships among the attributes. These different types of knowledge require different styles of employment and user interaction if they are to assist in partially automating the KDD process. Considering the choice of data mining task, the impact of domain knowledge on the choice of task and consequently the choice of algorithm depends first of all on deciding upon the strategic requirements of the analysis, making this part of the domain knowledge base, and from this, judgments on the appropriate task can then be made. This particular issue is at a very early stage of resolution. There is some work on how domain knowledge can support

particular algorithmic approaches, once the task is chosen[14], but little indication of how it can assist in choice of general approach (and thus task).

## IV. A MODEL FOR DOMAIN CENTERED KDD

A user centered workbench KDD system architecture is proposed which exploits domain knowledge to assist at every stage in the process. The architecture is based on a recognition that domain knowledge can be broadly grouped into three categories (1) that which is suitable for formal capture into predefined types; (2) that which is captured on a case by case basis based on previous analysis exercises; and (3) that which can only be applied manually in the current analysis. To successfully exploit these three categories of knowledge a partially automated workbench style tool is required that is guided by the stored domain knowledge where possible with the user having the ability to finally determine any action if they intervene.

The KDD process requires as part of pre-processing an additional domain knowledge acquisition step. To carry out this step a *Domain Knowledge Acquisition Module* would allow the user to enter the general objectives of the analysis and indications of matches to previous data analysis cases and also to enter domain knowledge under the predefined types. Specifically the user would be asked to indicate the general objectives of the data analysis exercise the KDD workbench is being used for. This information would permit two major functions. The first would be for the user to request, from a choice of tasks, for example clustering, classification and association analysis, which task is required. The second function would be to allow the case based reasoning module to have comparison criteria to select matching previous analysis cases. Based on the criteria the user would have a percentage indication of the strength of the match and details of the previous data analysis case. Based on this they would accept or reject the case as suitable to guide the current processing. The current analysis will be added to the case base when it is complete. (Note also that a data analysis case study repository is suggested to be established on the web, so cases could be accessed as a web service if the architecture permitted [24]).

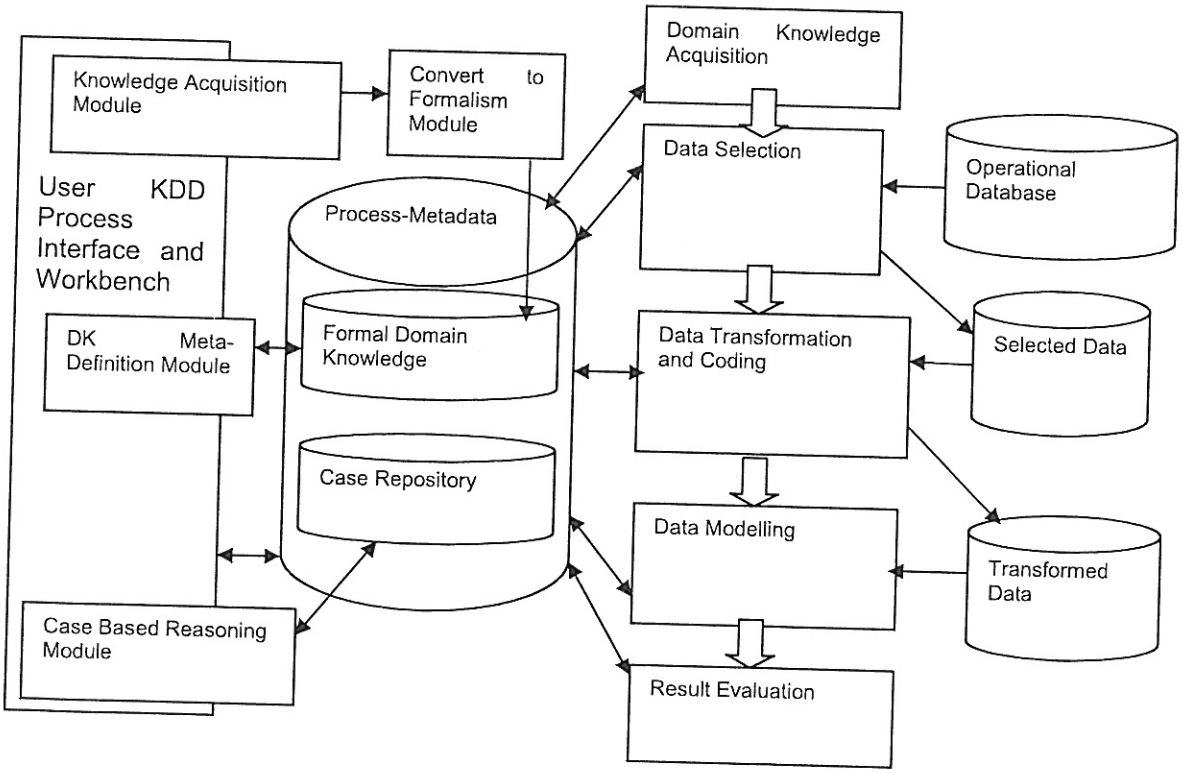


Fig. 1. Architecture for a Partially Automated KDD System

The user would then be requested to enter domain knowledge under the predefined categories through a series of wizards that are supported by the meta-data definitions of the data being analyzed. The wizards would make use of existing meta-data to offer the user a GUI interface that invited the user, via menu selection and linking of attributes, to indicate domain knowledge under each of the pre-defined categories. It would also contain a *Domain Knowledge Meta Definition Module* to allow the definition of new categories of domain knowledge if the existing formalism required extensions.

This domain knowledge is then converted into the chosen formalization, using the *Convert to Formalization Module*, so it is in a form suitable to be used for automated support of the appropriate stages in the KDD process. The module

would take as input any specific domain knowledge, relating to the attributes; converting that input into a formal language suitable for manipulation by the software. The KDD workbench software would then carry out, under user direction, the tasks of selection etc. at each step in the KDD process. Applications for each type of domain knowledge would be defined and implemented in the KDD workbench software and the user would be prompted to accept or reject the pre-defined use of the domain knowledge at each step in the KDD process.

The KDD workbench software would use the categorized domain knowledge in concert with the information from the *Case Based Reasoning Module* to take the user through the steps of the KDD process based on an appropriate pre-defined set of steps matched to the objectives of the analysis. These steps could be varied by the *Case Based Reasoning Module* taking account of previous cases, if there is a case that is a reasonable match as indicated earlier by the user in the knowledge acquisition stage. As the processing proceeds, the user would be asked to accept or reject each processing step and associated parameters and actions. The user has ultimate control of the path taken through the processing through their ability to override any suggested action and manually carry out alternative actions if desired.

## V. CONCLUSION

There are currently a number of limitations in data mining tools. These include limited pre-processing ability, excessive processing loads created by large volumes of data, a lack of provision for capturing and exploiting domain knowledge, limited guidance as to the most suitable analysis objectives and little indication of which results are interesting or significant. Considering these issues we conclude that an approach is needed that combines a number of techniques and that domain knowledge can take a leading role in coordinating those techniques and guiding the KDD process. The approaches suggested by these conclusions are incorporated in the domain knowledge centered model for the KDD process proposed and the model thus addresses the limitations identified. The model by including an initial step of knowledge acquisition allows the domain knowledge captured to assist in establishing strategic objectives for the data analysis, data selection, data pre-processing, data transformation, algorithm selection and results interpretation. By permitting persistent storage of domain knowledge from the current analysis and also the processing decisions from past cases, the path the user takes through the KDD process can be suggested to them while they retain their discretion to override any actions if they wish.

Data mining tools and their use are at an early stage of development. For domain knowledge to successfully contribute to automating the process its' use must be broad and comprehensive enough to be useful at all stages in the process. It will also have to be combined with other approaches; specifically visualization and case based reasoning. There is much room for improvement and many issues to be addressed.

Issues not fully resolved include

- the establishment of a suitably general classification of domain knowledge such that it is applicable across many domains
- the adoption of a formal language and notation for manipulating the established domain knowledge classes that will gain acceptance as a standard while still being extensible as domain knowledge classes grow or change
- the development of a knowledge acquisition module able to capture, via a user interface, domain knowledge from the user and translate it into the adopted formal language suitable for control of the knowledge discovery process
- the integration of the above elements into a workbench style toolkit that permits user intervention when suggested steps or transformations are not acceptable to the system user
- consideration as to the interface required to allow the domain knowledge categorizations to be extended by the user if the domain in question does not match well
- the incorporation of a case-based reasoning module, and the required meta-data, that can make use of similar data cleaning and data transformation cases to guide the current user. And also to guide the user in all steps if similar cases are held in the meta-data repository.
- the consideration of how web services could be designed to provide a case base repository, data processing and software for various steps in a coordinated fashion.

It needs to be recognized that domain knowledge will not always be able to be codified and in consequence the system should trap what domain knowledge it can and permit the user to intervene in the knowledge discovery process when they wish to override any suggested actions taken by the system in the current processing. The use of meta-data drawn from previous analysis activities should inform the current steps when the user or system recognizes similarities between the current task and past cases.

There are some common threads apparent in the research considered. A number of researchers see the need to store meta-data, to permit user interaction and involvement with the process and to incorporate data and knowledge visualization tools into the system to facilitate the user interaction that occurs. Additionally it is suggested here that the user involvement can be split into an initial knowledge acquisition task and subsequent user interventions as required in the knowledge discovery process. Thought needs to be given as to how knowledge acquisition, data selection and control of the knowledge discovery process should be designed from a human-computer interaction point of view to allow non-experts in data mining to make effective use of the system.

## VI. REFERENCES


[1] Brachman, R.J. and T. Anad, *The Process of Knowledge Discovery in Discovery in Databases*, in *Advances in Knowledge Discovery and Data Mining*, F.U. M., Editor. 1996, AAAI Press: Menlo Park, California. p. 37-57.

[2] Adriaans, P. and D. Zantinge, *Data Mining*. 1996, Harrow, England: Syllogic, Addison-Wesley.

[3] Chapman, P., et al., *CRISP-DM 1.0 Step-by-step data mining guide* www.crisp-dm.org. 2000, CRISP-DM consortium.

[4] Kopanas, I., N.M. Avouris, and S. Daskalaki, "*The Role of Domain Knowledge in a Large Scale Data Mining Project,*". Methods and Applications of Artificial Intelligence : Lecture Notes in Artificial Intelligence, 2002: p. 2308: 288-299.

[5] Lee, L.M., T.W. Ling, and W.L. Low. "*IntelliClean: A Knowledge-Based Intelligent Data Cleaner,*". in *International Conference on Knowledge Discovery and Data Mining*. 2000. Boston, Massachusetts, United Staes: ACM Press, New York,NY,USA.

[6] Maydanchik, A., "*Challenges of Efficient Data Cleansing,*" www.dmreview.com/article_sub.cfm?articleID=1403, in *DM Direct Newsletter September 1999*. 1999, The Thomson Corporation and DM Review.

[7] IST, "*Mining Mart - Enabling End-User Datawarehouse Mining,*": *www-ai.cs.uni-dortmund.de/MMWEB/content/annex.html*. 2003.

[8] Knobbe, A., A. Schipper, and P. Brockhausen, "*Domain Knowledge and Data Mining Process Decisions: Enabling End-User Datawarehouse Mining; Contract No. IST-1999-11993; Deliverable No. D5,*": *www-ai.cs.uni-dortmund.de/MMWEB/content/publications.html*. 2000.

[9] Cabena, P., et al., *Discovering Data Mining: From Concept to Implementation*. 1998, Upper Saddle River, New Jersey: Prentice Hall PTR. 195.

[10] Donoho, S. and L. Rendell. "*Constructive Induction Using Fragmentary Knowledge,*" in *Machine Learning: Proceedings of the Thirteenth International Conference (ICML'96)*. 1996. Bari, Italy: Morgan Kaufmann.

[11] Anand, S.S., D.A. Bell, and J.G. Hughes. "*The Role of Domain Knowledge in Data Mining,*" in *Proceedings of the Fourth International Conference on Information and Knowledge Management*. 1995. Baltimore, Maryland, United States: ACM Press New York, NY USA.

[12] Yoon, S., et al. "*Using Domain Knowledge in Knowledge Discovery,*" in *Proceedings of the Eighth International Conference on Information and Knowledge Management*. 1999. Kansas City, Missouri, United States: ACM Press New York, NY, USA.

[13] Han, J., L.V.S. Lakshmanan, and R.T. Ng, "*Constraint-Based, Multidimensional Data Mining,*" in *Computer*. 1999. p. 46-50.

[14] Domingos, P., "*The Role of Occam's Razor in Knowledge Discovery,*" *Data Mining and Knowledge Discovery*, 1999. **3**: p. 409-425.

[15] Ho, T., T.D. Nguyen, and D.D. Nguyen. "*Visualization Support for a User-Centered KDD Process,*". in *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2002. Edmonton, Alberta, Canada: ACM Press New York, NY, USA.

[16] Han, J. and N. Cercone. "*RuleViz:A Model For Visualizing Knowledge Discovery Process,*" in *International Conference on Knowledge Discovery and Data Mining*. 2000. Boston, Massachusetts, United States: ACM Press, New York, NY, USA.

[17] Accrue Software Inc, *Accrue Decision Series: /*www.accrue.com. 2004.

[18] Bechers, J.D., P. Berkhin, and F. Edmund. "*Automating Exploratory Data Analysis for Efficient Data Mining,*" in *International Conference on Knowledge Discovery and Data Mining*. 2000. Boston, Massachusetts, United States: ACM Press, New York, NY, USA.

[19] Ankerst, M., "*Report on the SIGKDD-2002 Panel: The Perfect Data Mining Tool: Interactive or Automated?,*" SIGKDD Explorations, 2002. **4**(2): p. 110-111.

[20] Palpanas, T., "*Knowledge Discovery in Data Warehouse,*" ACM SIGMOD Record, 2000. **29**(3): p. 88-100.

[21] Padmanabhan, B. and A. Tuzhilin. "*A Belief-Driven Method for Discovering Unexpected Patterns,*" in *International Conference on Knowledge Discovery and Data Mining*. 1998. New York, NY, USA: ACM Press, New York, NY, USA.

[22] Silberschatz, A.T., Alexander, "*What Makes Patterns Interesting in Knowledge Discovery Systems,*" IEEE Transactions on Knowledge and Data Engineering, 1996. **8**: p. 970-974.

[23] Pazzani, M.J., "*Knowledge Discovery From Data?,*" IEEE Intelligent Systems, 2000(March/April 2000): p. 10-13.

[24] Bohanec, M., et al. "*Describing Decision Support, Data Mining, and Text/Web Mining Studies in SolEuNet,*" in *ECML/PKDD01 2nd International Workshop Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning (IDDM-2002)*. 2002. Helsinki.