KDD IN FM: KNOWLEDGE DISCOVERY IN FACILITIES MANAGEMENT DATABASES

Graham Goulbourne, Frans Coenen and **Paul Leng** [1]

Abstract. The KDD in FM project aims to investigate how Knowledge Discovery in Databases (KDD), and particularly data mining, techniques can be applied to the distributed, heterogeneous and autonomous data sources found in the Facilities Management (FM) environment. The problems associated with multiple disparate databases are examined as is recent research in heterogeneous database mining. Finally, we describe the architecture of a system for KDD in this environment and suggest some suitable data mining techniques.
Key words: Data mining, Heterogeneous databases, Facilities management.

1 INTRODUCTION

Knowledge Discovery in Databases (KDD) has, in recent years, generated much interest for the database community. KDD, and the related concept of data mining, are concerned with the extraction of previously unrecognised and "interesting" information contained within (usually large) data repositories. "Interest" is of course a subjective concept, and a definition of what is interesting is required: it is usually taken as an overall measure of pattern value, combining validity, novelty, usefulness and simplicity (Fayyad [10]). Given that many data mining techniques have been developed, the first question faced by the practitioner is, which of these are most likely to be effective in producing interesting information within necessary limitations of time, budget and resources. The answer to this question is context-dependent; it will depend on the nature of the data being examined, the form(s) in which it is represented, and the objectives of the organisation seeking the information.

In this paper we consider the application of KDD to the Facilities Management (FM) function of an organisation. Facilities Management is characterised by a great diversity of function, most of which is only peripherally related to the core objectives of the organisation, and elements of which are essentially unrelated to each other. This diversity of function is reflected in a corresponding diversity of information and data organisation. Further, many of these information systems are likely to be *legacy* systems in that they have evolved within the organisation over a long period of time; as such they tend to be large, developed using old technology, complex and critical to the organisation's day-to-day working. KDD in this context poses particular challenges, both in identifying objectives for information discovery and, especially, in dealing with the issues raised by this extreme heterogeneity. A second question, therefore, facing practitioners wishing to apply KDD to FM (and a necessary prerequisite to data mining) is how to address the problems associated with the distributed, heterogeneous and autonomous nature of the data sources that characterise the FM environment. These problems raise issues of database interoperability, but with the distinctive feature that, for KDD, we are concerned only with the retrieval of data; we do not wish to update it.

The KDD in FM project thus has two aims; to investigate: (1) which data mining techniques are best suited to the FM operation, and (2) what are the most appropriate mechanisms to address the problems associated with the nature of the data sources to be accessed. These two interlinked questions are considered in this paper and some answers suggested. We commence, in Section 2, with a review of the facilities management application domain and consideration of the broad features and characteristics of KDD with respect to FM. A review of data mining techniques is then presented in Section 3 including an evaluation of the techniques, with respect to characteristics of the FM function. Issues concerning the heterogeneity and distributed nature of the data sources in question are considered in Section 4, in which we present a proposed architecture of a system for KDD in FM. In section 5 we bring together the twin strands of this paper with a discussion of KDD in a multi-database setting such as that characterised by FM. Section 6 presents our overall conclusions and the current status of the KDD in FM project.

2 FACILITIES MANAGEMENT

The FM operation of a company can be defined as an integrated approach to operating, maintaining, improving and adapting the buildings and infrastructure of an organisation in order to create an environment that strongly supports the primary objectives of that organisation (Further Education Funding Council [11]). Businesses may have already considered the distinction between their core and non-core businesses, as part of a drive to achieve better value for money and customer satisfaction. There is likely to be pressure to look for savings in the non-core business. Cutting operating budgets may be a financial expedient but it does not foster the long-term development of the business. Since running a business involves complex, co-ordinated processes and activities the response has to be one of taking an integrated view. A piecemeal approach to cutting costs is unlikely to produce

[1] Department of Computer Science, The University of Liverpool, Chadwick Building, P.O. Box 147, Liverpool L69 3BX, England. Tel: 0151 794 3709 Fax: 0151 794 3715 email: graham_g@csc.liv.ac.uk

Submitted to DEXA 98
February 23, 1998

those savings and may impair a business's ability to deliver high quality services.

Essentially, therefore, FM attempts to bring together under one organisational heading a number of operations which are largely functionally independent. These operations, for an organisation whose core business is (say) retailing of goods and services, might include: planning and design of premises; estate management; building maintenance; catering; staff travel and accommodation; car fleet management; printing services; and any other activities which are not intrinsic to the principal business function. In practice, within most large organisations, it will be necessary for each such operation to be managed separately. As an example, the Fleet Unit within an FM operation would be responsible for all aspects of vehicle policy within the company including types of vehicle available, delivery times, breakdown and recovery service, and maintenance; in fact, every aspect of running a fleet of perhaps 5000 or more vehicles. The scale and scope of this operation demands in itself a substantial organisational infrastructure which may be quite separate from that of the parent business. This tends to militate against holistic approaches to customer service, financial management, and other issues which are orthogonal to the functional organisation of the business. It is this problem which Facilities Management seeks to address.

Inherent in the nature of the FM operation is the likelihood that the information being considered will be:

1. *Distributed* among a number of separate databases each belonging to an individual area within FM: e.g. fleet DB, catering DB, print DB, etc. In many cases these databases will also be geographically distributed.
2. *Heterogeneous* in that they are likely to have been developed and be maintained independently; possibly using different platforms, DBMS, schemas and query languages, and to exhibit both structural and fundamental semantic heterogeneity (Colomb [6]).
3. *Autonomous* in that they are all managed and maintained independently of each other.

Note that; Structural semantic heterogeneity refers to differences in the representation of equivalent information in different databases: for example, differences in terminology used. Fundamental semantic heterogeneity occurs when the representations used in different databases do not allow identification of common information.

3 DATA MINING TECHNIQUES

The essential aim of KDD is the extraction of new and interesting knowledge from a database or databases. In this section we present a review of a number of data mining techniques. To provide a framework we have broadly categorised the techniques available under the following headings:

1. Meta-learning strategies
2. Probablistic approaches
3. Inductive learning techniques
4. Neural Networks
5. Genetic algorithms

Each is discussed in further detail in the following sub-sections. In sub-section 3.1 we evaluate the techniques overviewed in the context of the FM environment.

3.0.1 *Meta-learning strategies*

Meta-learning strategies have been employed in [5] to search for patterns held within local and remote sites. The example used is that of credit card transaction fraud. Each bank has a database of customer purchases and would not release this information to other banks, for obvious reasons. If fraud patterns are identified within the constituent databases then the patterns can be encoded as executables and treated as "black boxes". The main thrust of the research is to enhance the performance of locally learned models by merging them with models obtained from remote sites.

The approach is to meta-learn a set of new classifiers whose training data is based on the predictions of an underlying set of base classifiers. A base classifier is the outcome of applying a learning algorithm to raw data. An arbiter or combiner is a program generated by a learning algorithm that is trained on the predictions produced by a set of base classifiers and the raw data. The arbiter/combiner is also a classifier and therefore other arbiters/combiners can be computed from the set of predictions of other arbiters/combiners. The new classifiers generated from one data set can be imported to work with other classifiers generated from other data sets. The results obtained from this research suggest local meta-learning techniques can significantly raise the accuracy of local classifiers.

3.0.2 *Probabilistic approaches*

Probabilistic work has been undertaken (for example by McClean and Scotney [16], to provide the user, (typically a business manager) with high level summaries of the interesting contents of distributed databases. McClean and Scotney [16] use the Dempster-Schafer (DS) representation of evidence to describe categorical data and provide a framework for the extraction of data. DS representation is particularly suited to data that may include missing values and be held at different levels of aggregation. It also enables samples in a distributed database to be efficiently combined using the weighted sum operator. This means that without resorting to the raw data, new rules can be induced at a finer level of granularity than was available in any of the contributing data sources.

The DS representation allows the coarsening or refinement of the data by changing to a higher or lower level of granularity. This is achieved by reallocating the masses[2] using the coarsening and refining operators respectively. This is an essential feature for systems that manage uncertainty as part of KD, since in the search for rules ways to aggregate or disaggregate the data must be available. If a rule can be generalised at a higher level of aggregation then it becomes a more powerful statement of how the domain behaves; *cf.* Ockham's razor. By calculating the belief and plausibility of a given proposition

---

2

$$m(\phi) = 0$$

i.e. the mass function of the null proposition is always zero.

$$\sum_{A \subseteq \Omega} m(A) = 1$$

i.e. the sum of the masses of all possible propositions in the frame of descernment is one from the combined mass function, a probability function may be assigned and hence a quantifier to the proposition, thus inducing an appropriate rule. The ability to view the data at various levels of granularity and to calculate accompanying beliefs in associated propositions is useful when combining samples with different schemas.

3.0.3 Inductive learning techniques

Inductive learning techniques are a feature of INLEN, developed at George Mason University ([17]). The name INLEN is derived from **in**ference and **le**ar**n**ing; the KDS (Knowledge Discovery System) contains the AQ algorithm which uses inductive inference to learn decision rules from examples. INLEN has been used to provide insights into characteristics and relationships that may exist in the data but are unknown to the user. In order to perform knowledge discovery across multiple databases the primary and foreign keys must be included, since these serve as links across the databases.

Knowledge generated by INLEN from a particular database is stored in "segments"; each one contains information that includes database name, decision variable, rule head, rule body and a number of covered examples. The knowledge segments for one database are grouped together into a knowledge base. The approach is to combine information about primary and foreign keys from the database schemas with the knowledge segments from the multiple databases. This allows knowledge discovery from multiple databases, without having to construct new databases, (e.g. universal relations) and perform knowledge discovery within them.

3.0.4 Neural Networks

Two types of neural network have been used recently in data mining research: the back propagation net and Kohonen self organising maps. Although a powerful classification method back propagation networks can be criticised for their "black box" nature, i.e. they provide answers but without an explanation of how these are arrived at. A large training set can be required making the classification of large data sets computationally intensive.

A hybrid neural/expert system has been developed [4], to find significant events in a market research database. A neural network is trained by back propagation to classify trends in the time series data. A rule system then uses these classifications to infer the significance of the data. The system was able to analyse six months data in 5 minutes, compared with 4 days for the manual analysis of the same data.

Self Organising Maps (SOM) in conjunction with a WWW type browser interface have been developed to provide a method for reviewing textual information, [14]. The document collection is organised into a map in an unsupervised manner utilising the statistical information of short word contexts, i.e. word catagories rather than individual words. The resulting ordered map shows where similar documents lie near each other and presents a general view of the document space. This allows the user to navigate between similar documents. A visualised map of the landscape may reveal surprising connections between different areas of knowledge.

3.0.5 Genetic Algorithms

Genetic Algorithms (GAs) offer a robust method of finding a solution; if a solution exists then it is likely that a well coded GA will find it. Conceptual clarity is one of the GA's main advantages; many non-specialist clients have difficulty grasping computing concepts, whereas the Darwinian theory of evolution and survival of the fittest is simple to understand. The output from a GA is usually in a reasonably readable form so the client can see how the computer has arrived at its conclusion.

The Dutch airline KLM uses a GA-based data mining technique to predict the career paths of pilots based on learning rules from historical data [1]. Pilots have a complex career structure and being able to predict accurately which pilots will "bid" for which "seat" has allowed better support of strategic investment decisions. The implementation of an integrated strategic planning system, (including the use of the GA) has led to a reduction of the total manpower costs of 2%, giving a pay-back time of less than 1 year.

3.1 Evaluation

The first three of the above methods are much more suited to dealing with heterogeneous databases all, assume that the databases being mined or manipulated contain a degree of commonality. The Dempster-Schafer mining and local meta-learning techniques offer the possibility of mining heterogeneous data sources but the examples offered are of banks which will undoubtedly have similar data within their databases. INLEN requires primary/foreign keys to be present in each of the databases. Other techniques used in KDD for identifying patterns, such as neural networks, are predicated on the assumption that the data being analysed can be presented in a uniform and consistent way. These observations are made not as criticisms, but to emphasise the gulf between an idealised view of KDD and the real problems which we perceive in the highly heterogeneous envoronment of FM. We seek a solution that will enable the application of a variety of KDD techniques within a common framework that addresses these problems. Methods used for KDD when multiple databases are involved, (such as the FM environment), may be broadly viewed under three headings:

1. Application of KDD techniques to each database separately; an example of this would be inducing a set of rules and applying some interestingness measure to them. Such work is found in Silberschatz [19], Kamber and Shinghal [13] and Dzeroski and Grbovic [8].
2. Selection of a number of related, although not necessarily homogeneous databases, such as, premises - planning and design, estate management, and building maintenance, and integrating them for the purpose of data mining. Such work is found in McClean and Scotney [16], Levy [15] and Ganesh et al. [12].
3. Creation of a data warehouse and integrating all the databases, either for the purposes of mining the warehouse or to establish data marts for independent mining activities [21].

Application of KDD to each database in turn clearly poses fewest problems in dealing with heterogeneity. Unfortunately, the problems which KDD seeks to address in FM are essentially those which involve interactions between separate operations and hence associations between the corresponding databases. The underlying hypothesis is that some aspect of (say) building maintenance has an unsuspected influence on (say) the cost of the print operation, and that it may be possible to infer this relationship from some examination of the two (or more) corresponding databases. We therefore need to consider methods involving the integration of information from multiple, disparate, sources.

4 AN ARCHITECTURE FOR MINING HETEROGENEOUS DATABASES

In seeking to apply data mining techniques in the field of Facilities Management, we require in particular to consider methods by which the relevant data can be made available for mining. This requires us to consider ways of integrating information held in a number of databases which are heterogeneous, geographically distributed, and autonomously maintained. The usual solution recommended in this situation is to establish a data warehouse, within which data extracted from all the relevant databases is stored in a common format.

There are a number of reasons why this approach may be unsuitable in the environment we are considering. The number of databases involved may be relatively large, and their structural heterogeneity very great, leading to significant problems and costs in establishing a common data representation. The costs of storing a large volume of integrated data, and of updating this when required from the diverse distributed sources, may be significantly high. Many attributes in the larger data set will be irrelevant to the current knowledge discovery task and therefore make the extraction of useful knowledge more difficult and computationally more expensive. There may also be ethical and even legal objections to integrating data which may include, in some of the databases, personal information which is held subject to the proviso that it is maintained securely. Finally, an integrated data warehouse is always liable to be undermined by the operational autonomy of the contributing databases; each organisational change may require a corresponding reorganisation of the warehouse to accommodate it.

The alternative approach is to organise the contributing databases into a multidatabase system, through which it is possible to access data from all sources using some unifying schema. Multidatabase architectures typically use a Canonical (Common) Data Model (CDM) where schemas are expressed in a common Data Definition Language (DDL). Each local schema associated with a component databases, and expressed in a local DDL, is mapped onto a component schema which represents the same information in terms of the CDM and common DDL. A query against a component schema is translated into queries against the underlying schema. The results of such queries are then processed to form an answer to the initial query. Typically component schemas are integrated into a multiple component schema referred to as a federated schema, also expressed in the common DDL, which is then the user's first point of contact. A query against a federated schema is translated into queries against the underlying component schemas and then on to the local schema. The software which co-ordinates, i.e. facilitates the integration of, a number of component databases in a multidatabase is referred to as a Federated Database Management System (FDBMS).

Architectures for multidatabase systems are reviewed by Fahl [9]. Four alternative models are identified:

1. Global Schema Architectures: In this approach all component schemas are integrated into a single federated schema. This is in principle the simplest approach, but shares some of the disadvantages we have ascribed to data warehousing solutions. To be effective, a global schema may need to undermine the autonomy of the component systems. The most serious drawback, however, is the need for all conflicts between component schemas to be resolved. This is unlikely to a a feasible option in the highly heterogeneous context we are considering.
2. Multiple Integrated Schema Architectures: Here a number of federated schemas are supported, each of which integrates a particular set of parts of the participating component schemas. Note that overlap between groups of component schemas is allowed. The advantage is that less integration of component schemas is required, and not all information need be included. This approach is appropriate when particular needs can be identified which can be addressed by a partial integration of the available data.
3. Federated Schema Architectures: This is similar to the multiple integrated approach except that each user site has its own dedicated federated schema. The distinction between this and the previous architecture is sometimes expressed using the terms tightly coupled and loosely coupled. In the previous case the federated schemas are maintained by a "multidatabase" administrator (tight coupling), while here the integrated schemas are the responsibility of individual users (loose coupling).
4. Multidatabase Languages Architectures: This last approach is different from the foregoing in that users do not access data through an integrated schema. Instead a data manipulation language is used that provides constructs to access component schema and consequently combine data from the component databases making up the multidatabase.

Global schema integration poses what may be unresolvable problems of reconciling cases of semantic heterogeneity. Conversely, methods involving partial integration of schemas bypass these problems by recognising that in particular applications, total integration may be unnecessary. In the KDD context, however, avoiding integration of problematic data runs the risk of losing vital information: because we are (by definition) searching for relationships which are not immediately apparent, we cannot a priori eliminate any data from our search.

Fortunately, the requirements of a system to support KDD across multiple databases are less demanding than in the general case of a multidatabase architecture. First, KDD requires only read access to data, without the need to update it. Secondly, and in our view most significantly, it may be possible to draw "interesting" inferences from data even when this cannot be integrated in the strict sense usually required of a federated DBMS. For example, a Personnel database may contain records of all employees, and another database, records of payments made for staff travel and hospitality. It may be impossible reliably to match records in the two data, for various

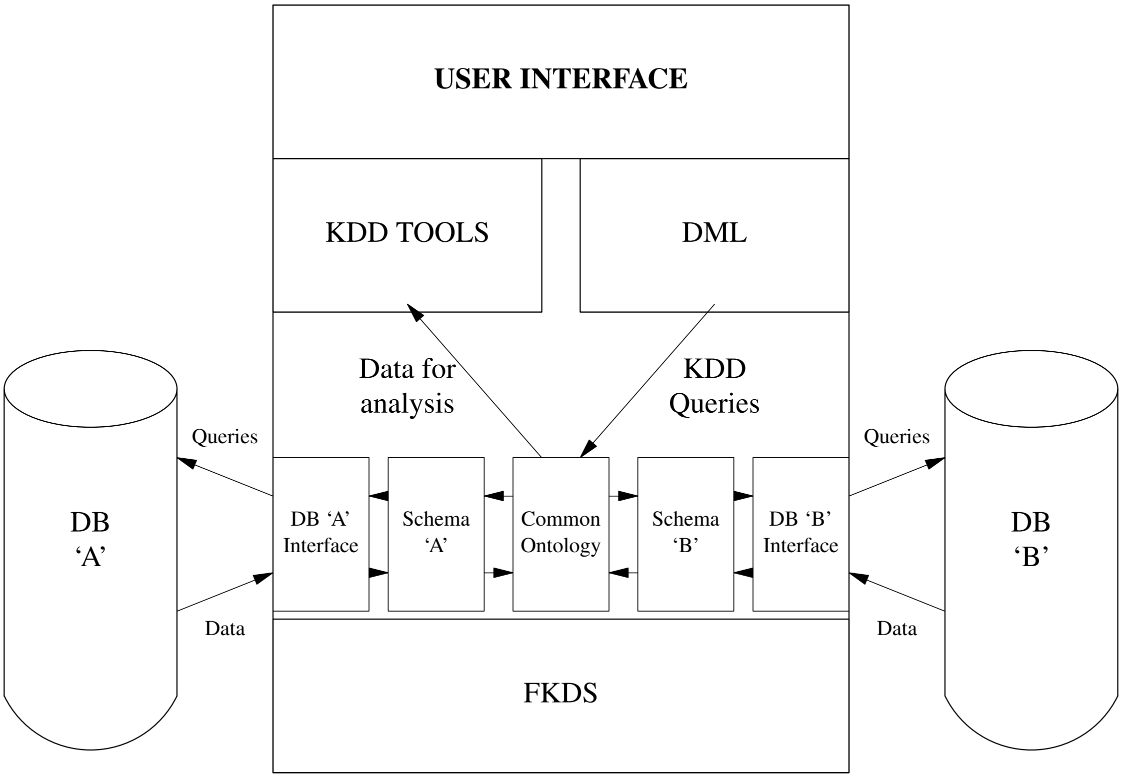

Figure 1. A Conceptual view of the FKDS architecture

reasons; for example, there may be cases of payments being made to individuals who are not employees, or the attributes of the payment records may not enable reliable identification of the payees. This is, therefore, an instance of fundamental semantic heterogeneity which would normally prevent a full effective integration of the databases. However, it may be that there is, in practice, a very significant relationship between the two databases; it may be known that in almost all cases, recipients of travel payments are current employees, so a search for interesting patterns within the two sets of data may proceed on the assumption that this is the case. For this purpose we do not require a complete integration of the component schemas; it may be sufficient to define approximate relationships between entities, sufficient to enable heuristic searches for interesting patterns. Because of the relaxation of the requirements for integration, and the absence of a requirement for data updates, it is inappropriate to refer to this kind of system as a Federated Database Management System: we will instead describe it as a Federated Knowledge Discovery System (FKDS). The appropriate model for this is the Multidatabase Language Architecture. We postulate a Data Manipulation Language which provides constructs to access and combine data from component databases, including constructs which reference data in terms of the imprecise relationships which may exist between imperfectly matched entities. The principal elements of the FKDS (Figure 1) are:

1. A common Data Definition Language, in which schemas of all the participating component databases are expressed.
2. A common ontology, within which relationships between the component databases are described. For the reasons discussed above, the requirements of this ontology are less exacting than would be the case for a federated schema.
3. A Data Manipulation Language, the semantics of which enable queries formulated for knowledge discovery purposes to be translated via the shared ontology into queries to the component databases.
4. Tools for Knowledge discovery. In the generic FKDS model, these may include tools making use of genetic algorithms, neural networks, or any other chosen methods which can be applied to data extracted and combined using the DML.
5. Interfaces to the component databases. Because of the autonomy, geographical distribution, and technological diversity of the component databases, these interfaces may need to be very specialised; their purpose is to translate queries expressed in terms of the common DDL into queries expressed in the form required by the component database, and vice versa for responses. They may also include filters to prevent access to confidential or sensitive data when necessary.
6. A user interface to enable application of the available KD tools and the formulation of KD queries using the DML.

The FKDS architecture is generic, in that it provides a framework for KDD within a heterogeneous context, without defining which specific techniques of knowledge discovery will be used. We see this as an essential attribute of a system for KDD in FM, as the diversity of the FM function may require a range of different techniques to be applied.

5 KDD IN A MULTI-DATABASE ENVIRONMENT

It is becoming increasingly common for distributed data to be used for the purposes of knowledge discovery either as part of a distributed database management system, or using autonomous databases with open database connectivity. The distributed nature of the information allows for the combination of the data and therefore provides a new and fertile area for the application of data mining techniques. Two such techniques are reviewed;

1. Entity integration techniques.
2. Query language extensions (e.g. SchemaSQL).

5.0.1 Entity integration techniques

Identification of records that represent the same real world entity is an important task in the database integration process. When a common identification mechanism for similar records across heterogeneous databases is not readily available, entity integration can be performed by examining the various attribute values among the records. The distances between attribute values can be used as a measure of similarity between the records they represent. Record matching conditions can, for entity integration, then be expressed as constraints on the attribute distances. KD techniques can be used to automatically derive these conditions, (expressed as decision trees) directly from the data, using a distance based framework [12].

The process of integration requires two distinct tasks: schema integration and entity identification. The former resolves mismatches such as homonyms (the same name for different attributes), and synonyms (different names for the same attributes). The removal of these inconsistencies allows the creation of a global schema, making it possible to treat all the records uniformly for further processing. The mapped global database may contain multiple instances of the same real world entity, therefore the identification of distinct entities is the is the purpose of the entity identification phase.

5.0.2 Query language extensions

SQL is the most common language for the extraction of information from databases. An extension of SQL, (SchemaSQL) has been developed, [19] that offers the capability of uniform manipulation of data and meta-data in relational multi-database systems. Interoperability in multiple databases is the ability to share, interpret and manipulate information in component databases. Factors influencing interoperability can be classified into semantic issues (interpreting and cross-relating information in different databases), syntactic issues (heterogeneity in schemas, data models and query processing) and systems issues (operating systems, communication protocols, security management etc.). SchemaSQL deals with the syntactic issues; its development is aimed at extracting information from relational databases storing semantically similar data in structurally different ways. Some of the key features required of a language for the interoperability of relational multi-database systems are:

1. It must have an expressive power that is independent of the schema with which the database is structured.
2. To promote interoperability, it must permit the restructuring of one database to conform to the schema of another.
3. It must be easy to use yet still be expressive.
4. It must provide full data manipulation and view definition capabilities.
5. It must be downward compatible with SQL in view of SQL's importance in the database world.
6. The language must realise a non-intrusive implementation that would require minimal additions to the RDBMS.

The architecture for implementing SchemaSQL builds on the existing database architecture in a non-intrusive manner. The SchemaSQL system is built on top of the existing SQL systems. A SchemaSQL server communicates with local databases in the federation. Local meta-information, database names, names of attributes and relations, are stored on the SchemaSQL server in the form of a table called the Federation System Table. Global queries are submitted to the SchemaSQL server which determines the queries to submit to the local databases. The server then collects the answers and executes a final series of SQL queries to produce an answer to the global query.

6 CURRENT STATUS AND FUTURE PLANS

In this paper we have considered the issues of data mining with respect to the FM environment and particularly the distributed, autonomous and hetereogeneous nature of the sources involved. We have presented a model for a generic system for Knowledge Discovery amongst such data sources, which we call a Federated Knowledge Discovery System. Work is currently in progress to develop a prototype demonstration of a system with this architecture. This is to be achieved initially by developing a "test harness" using a number of different databases founded on different platforms and using different operating systems and DBMS e.g. Ingres on a UNIX platform, MS Access on a PC platform etc.. The FKDS will then be implemented using a WWW type interface supported by CGI scripts to achieve the desired connectivity. The "test harnes" will then be used to evaluate ideas regarding common ontalogies. When established this ontology will be use as the basis to define a Data Manipulation Language which will include constructs to allow for the manipulation of data which need not be precisely matched. The aim is to develop ways of extracting potentially interesting information from data even when the fundamental heterogeneity of this makes formal integration impossible. Work is currently in progress and the research team have been greatly encouraged by the results to date.

7 ACKNOWLEDGEMENTS

The KDD in FM project is funded by Royal Sun Alliance FM. For further information relating to the KDD in FM project see *** http://www.csc.liv.ac.uk/research/kddfm.html ***.

REFERENCES

[1] Adriaans, P. and Zantinge, D. (1996). Data Mining. Addison Wesley Longman Ltd. .

[2] Augier, S. Venturini, G. Kodratoff, Y. (1995). Learning First Order Rules with a Genetic Algorithm. Proceedings of the First International Conference on Data Mining and Knowledge Discovery, AAAI Press 1995, pp 21-26.

[3] Ciampi, A. and Lechevallier, L. (1995). Designing Neural Networks from Statistical Models: A new Approach to Data Exploration. Proceedings of the First International Conference on Data Mining and Knowledge Discovery, AAAI Press, pp 45-50.

[4] Ciesielski, V. and Palstra G. (1996). Using a Hybrid Neural/Expert System for DB Mining in Market Survey Data. Proceedings of the Second International Conference on Data Mining and Knowledge Discovery, AAAI Press, pp***DO WE KNOW THIS***.

[5] Levy, A. Meta Learning Strategies in Multi-Databases. Proceedings of the Second International Conference on Data Mining and Knowledge Discovery, AAAI Press.

[6] Colomb, R.M. (1997) Impact of Semantic Heterogeneity on Federating Databases, The Computer Journal, 40(5), pp235-244.

[7] Domingos, P. (1996). Linear-Time Rule Induction. Proceedings of the Second International Conference on Data Mining and Knowledge Discovery, AAAI Press, pp***DO WE KNOW THIS***.

[8] Dzeroski, S. and Grbovic, J. (1995). Knowledge Discovery in a Water Quality Database. Proceedings of the First International Conference on Data Mining and Knowledge Discovery. AAAI Press 1995, pp 81-86.

[9] Fahl, G. (1994). Object Views of Relational Data in Multidatabase Systems. Studies in Science and Technology, Sweden.

[10] Fayyad, U., Piatetsky-Shapiro, G. and Smythe, P. (1996). Knowledge Discovery and Data Mining: Towards a Unifying Framework. Proceedings of the Second International Conference on Data Mining and Knowledge Discovery, AAAI Press, pp 82-95

[11] Further Education Funding Council (1997) Effective Facilities Management: A Good Practice Guide. Her Majesty's Stationery Office.

[12] Ganesh, M., Sirvastava, J. and Richardson, T. (1996). Mining Entity-Identification Rules for Database Integration. Proceedings of the Second International Conference on Data Mining and Knowledge Discovery, AAAI Press 1996, pp291-294.

[13] Kamber, M. and R. Shinghal, R. (1996). Evaluating the Interestingness of Characteristic rules Proceedings of the Second International Conference on Data Mining and Knowledge Discovery, AAAI Press, pp 263-266.

[14] Lagus. K. et al. ***CAN WE PUT THE OTHER AUTHORS IN HERE*** (1996). Self-Organising Maps of Document Collections: A New Approach to Interactive Exploration. Proceedings of the Second International Conference on Data Mining and Knowledge Discovery, AAAI Press, pp***DO WE KNOW THIS***.

[15] Levy, A. (1996). Obtaining Complete answers from Incomplete Databases. Proceedings of the 22nd Conference on VLDB, VLDB 1996, pp 402-412.

[16] McClean, S. and Scotney, B. (1996). Using Evidence Theory for Knowledge Discovery and Extraction in Distributed Databases. Informatics Research Report. Number 11. University of Ulster.

[17] Ribero, J. Kaufman, K. Kerschberg, L. (1995). Knowledge Discovery from Multiple Databases. Proceedings of the First International Conference on Data Mining and Knowledge Discovery, AAAI Press, pp 240-245.

[18] Schafer, G. (1976). A Mathematical Theory of Evidance. Princeton University Press, Princeton, New Jersey.

[19] Silberschatz, A. and Tuzhilin, A. (1995). On Subjective Measures of Interingness in KD. Proceedings of the First International Conference on Data Mining and Knowledge Discovery. AAAI Press, pp 275-281.

[20] Toivonen, H. Klemettinen, M. Ronkainen, P. Hätönen, K. Mannila, H. Pruning and Grouping Discovered Association Rules. Working paper, Dept. of Computer Science, University of Helsinki.

[21] Gill, H.S., Rao, P.C. Computing Guide To Data Warehousing. Que Corp. 1996. ISBN 0-7897-0714-4.