A Proposal for Data Mining Management System

SK Gupta* Vasudha Bhatnagar† SK Wasan‡

Abstract

Knowledge Discovery in Databases, is an inherently iterative process requiring human interaction. The traditional model for KDD process takes a process-centric view and does not allow interaction during actual mining. The gross granularity of the KDD process discourages application development by the non-expert users on the data mining systems.

We present I-MIN model for KDD process and propose an architecture for a data mining management system. The model splits the KDD process into three phases. The schema designed during the first phase, abstracts the generic mining requirements of the KDD process and provides a mapping between the generic and (user) specific KDD sub-processes. The generic process is executed during the second phase and windows of condensed knowledge called Knowledge Concentrates are created, which abstract the intended knowledge. During the third phase, which corresponds to actual mining by the end users, specific KDD sub-processes are invoked to mine Knowledge Concentrates either using a declarative query language or by writing applications.

The architectural proposal emulates a DBMS like environment for the managers and end users in the organization. The architecture provides a set of mining operators for development of mining applications to discover and renew, preserve and reuse, and share knowledge for effective knowledge management. Complete documentation of all the KDD processes in the organization, provided by the Knowledge Discovery Schemas helps in controlling the environment.

Keywords: Intension Mining, Knowledge Management, Knowledge Concentrate, Knowledge Discovery Schema, Operators

1 Introduction

KDD technology is based on a well-defined, multi-step "KDD process" for discovering knowledge from large collections of data sets [8, 18]. The KDD process is iterative in nature, and depends on interaction for dynamic decision-making throughout as shown in Figure 1. Most data mining packages model the traditional KDD process, where the user decides the premining functions, applies them on the data repository and subsequently invokes the desired mining function using the selected algorithm [4, 12, 19]. Powerful and varied visualization methods are used to display the discovered knowledge, assisting the user in its interpretation.

*Deptt. of CSE, Indian Institute of Technology, New Delhi, India. email: skg@cse.iitd.ernet.in

†Deptt. of CS, MotiLal Nehru College, University of Delhi, Delhi, India. email: vasudha@cse.iitd.ernet.in

‡Deptt. of Mathematics, Jamia Millia Islamia, New Delhi, India. email: skwasan@yahoo.com

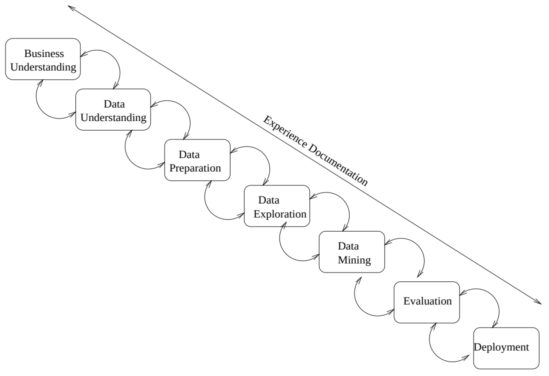

Figure 1. KDD Process model proposed by Reinartz [18]

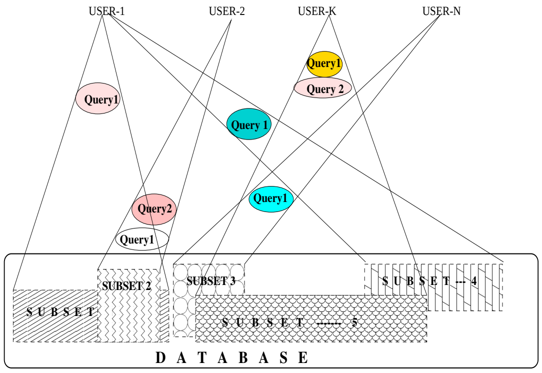

Figure 2. User-Centric Mining through simultaneous querying by multiple users

Present day KDD systems/packages require the end-user of the KDD technology to be data mining experts, since clear and complete understanding of the KDD process and the focusing solutions is essential to successfully steer the KDD process. Non-expert users need to work in close collaboration with the data miners. Subtle atomicity inherent in the traditional KDD process model, both at functional and volumetric levels, may require substantial modification in the process steps in case of any deviation from the set goal. This limits its functionality and dissuades the users to experiment creatively with the KDD process as shown in Figure 2. Autonomy in KDD systems is necessary to provide flexibility in setting mining goals to met dynamically changing knowledge needs. The autonomy must be controlled, and yet intelligently handled to drive innovation in formulating new business questions and creativity in solving them. The design of the KDD systems must consider human interaction and creativity as crucial components of the KDD process.

In this paper we present an architecture for a data mining management system based on a user-centric model for KDD process (I-MIN model), which abstracts the KDD process. The three level architecture facilitates development of applications to discover new knowledge from the evolving databases and explore already discovered knowledge with novel perspectives. The architecture insulates the mining applications from the details of the KDD process and provides mining operators for preservation and controlled sharing of the discovered knowledge, thereby facilitating knowledge management. Continuity in the KDD process affected by the architecture keeps the knowledge always current.

The paper is organized as follows. Section 2 describes the research related to KDD process models and KDD systems. Section 3 describes the I-MIN model and Section 4 lists the functional components of the model. Section 5 discusses the operators for mining and application development. Section 6 proposes three level architecture of I-MIN model. Section 7 reports the implementation and Section 8 concludes the paper.

2 Related Works

Significant contributions have been made by researchers toward understanding of the KDD process and design of KDD systems. Brachman and Anand [3] highlighted the human centricity of the KDD process model, which was further emphasized by Reinartz [18]. These works underscore the need for human interaction and its role in successful culmination of the KDD endeavor. CRISP-DM - CRoss-Industry Standard Process for Data Mining [5] - advocates a data mining methodology consisting of tasks described at four levels of abstraction. The methodology is based on the KDD process model that offers systematic understanding of step-by-step direction, tasks and objectives for every stage of the process. Theoretical formalization of the KDD process proposed by Williams [21], helps in differentiating and comparing alternative approaches.

The concept of second generation data mining has been proposed by Imielinski and Mannila [13]. Virmani [20] proposes a design of *Discovery Board* - a second generation data mining system. The proposal strives to provide a framework for DBMS-like environment supporting query language to satisfy basic data mining needs, and APIs for developing data mining applications to satisfy complex mining requirements. Psaila [17] uses operators to execute the KDD process in *AMORE* system exhibiting tight coupling between KDD process and SQL based database systems.

Architectures for several KDD systems have been reported by researchers. Matheus et al. in [16], present a model of an idealized KDD system and describe the way its components handle the requirements for knowledge discovery in real-life applications. DBMiner system tightly integrates On-Line Analytical Processing (OLAP) with wide spectrum of data mining functions [11]. Mineset, conceptualized by Brunk et al. [4], is based on a three tier architecture and supports complete KDD process. Mining Kernel System, designed as a set of libraries by Anand et al. in [1] embodies the interdisciplinary nature of data mining, by exploiting useful techniques from areas of statistics, machine learning, database technology, artificial intelligence and visualization.

2.1 Intension Mining

In *Intension Mining* scheme [2], mining goals are stored in form of a *Knowledge Discovery Schema* (*mining intension*) analogous to the *database schema* (*database intension*) in DBMS [6, 7, 14]. The schema in *"Intension Mining"* contains the specification of generic mining requirements (KDD process), just as the *database intension* contains the specification of all relations in the database. The ultimate goal of schema design is to facilitate a view that is of direct interest to the user, and enhance productivity, and ease of use and comprehension at the user level.

Intension Mining is fundamentally based on incremental mining concept. The incremental database is processed automatically at regular intervals with the periodicity specified in the schema. The processing consists of *pre-mining* of data followed by preliminary analysis and/or aggregation. Since the mining requirements are available in the schema, the system is capable of carrying out, premining-cum-aggregation operation in off-line mode. This periodic operation on the incremental database is termed as *Accumulation*. The resulting aggregates, called *Knowledge Concentrates*, constitute intermediate form of the intended knowledge and are preserved on secondary storage.

Intension mining is performed in three phases viz. *Planning phase*, *Accumulation phase*, *Mining phase*.

During *Planning* phase, detailed specifications of the KDD process are stored as *Knowledge Discovery Schema* (KDS). This phase involves collaboration between the data mining analyst, domain expert and end user. The schema is compiled like database schema, resulting into creation of meta-data and data structures[1] to be used during the later two phases.

Accumulation phase starts after compilation of the schema and continues until the user decides to drop the mining requirement (schema) altogether. During *Accumulation* phase the incremental database is pre-mined and aggregated in consultation with the meta-data to yield *Knowledge Concentrate* (KC). The KCs store the intermediate form of intended knowledge. They serve as *windows* of condensed knowledge for future mining.

The *Mining* phase is invoked by the user when a mining query is presented to the system or a mining application is executed. KCs are processed by the mining algorithm to discover the intended knowledge during this phase.

An important characteristic of *Intension Mining* is that it perceives KDD as a continuous process. Periodic *Accumulation* of incremental database at regular intervals gives rise to a sequence of *Knowledge Concentrates* providing non-overlapping *windows* in the database. These *windows* form the basis of the ongoing knowledge renewal and knowledge sharing, which are two important issues in knowledge management. A *Knowledge Discovery Administrator* (KDA) is responsible for the overall KDD operations in the organization, analogous to DBA. The overall approach allows systematic and complete documentation of the KDD operations in an organization, and helps in proficient management of knowledge and enforcement of standards in the organization. For details of the *Intension Mining* scheme, please refer to [2].

3 I-MIN Process Model

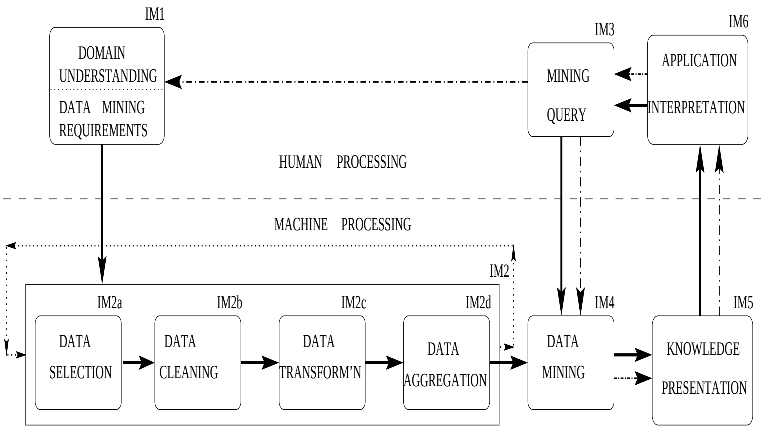

Figure 3. I-MIN model for Knowledge Discovery Process. Solid lines indicate data flow; Dotted lines indicates periodic repetition and Dash-Dot lines indicates optional repetition.

We present a user-centric model for the KDD process, which is based on the concept of *Intension Mining* [2] and is designed to support interactive exploration and experimentation with the KDD process. The model called "I-MIN Model" is shown in Figure 3. The model is downward compatible with the traditional KDD process model and provides full functionality for it. It can be realized by designing and integrating the agents for each of the process steps. The steps of KDD process are numbered IM$x, x = 1, \ldots 6$.

The KDD process begins with data understanding and formalizing the mining requirements during Step IM1. This corresponds to the *Planning phase* of *Intension Mining*. During this step discovery goals are identified and specified in terms of *Knowledge Discovery Schema*. The

[1] These data structures store the knowledge aggregated during *Accumulation* phase, and constitute *Knowledge Concentrates*.

schema is compiled and the resulting meta-data is stored for future use during *Accumulation* and *Mining* phases.

The second step IM2 is premining-cum-aggregation step and corresponds to the *Accumulation* phase. Step IM2 is a compound step in which steps IM2a - IM2c can be mapped to appropriate steps in the traditional KDD process model. Step IM2d is responsible for analysis/aggregation of pre mined data, which is carried out during data mining step of traditional KDD process model. Since the functions for pre-mining and aggregation operations are already specified in KDS, they are performed automatically without any human intervention. The outcome of this process step is a *Knowledge Concentrate.* This step is periodically repeated on incremental database as per the frequency specified by the user in the schema.

Step IM3 signifies initiation of the *Mining* phase. Mining queries are formulated and applications developed by the end users during this step. This user initiated step is asynchronous as it commences with either formulation of mining query or invocation of an application. KCs extracted during step IM2 can be restrictively shared for experimentation and monitoring of desired subsets of database. The discovered knowledge can be preserved and reused by developing applications to meet complex knowledge needs.

IM4 is the actual mining step during which the mining algorithm specified in the schema is invoked. The same algorithm is also invoked during step IM2d where aggregation is done partially. During step IM4 the intended knowledge is mined from the KCs.

The resulting knowledge is presented and interpreted/deployed in steps IM5 and IM6 respectively.

Though the proposed model subsumes the traditional KDD process model, an interesting contrast between the two models is that in traditional KDD process the functionality of the KDD process is defined at the beginning of the process by the data miner, while in I-MIN model it is decided dynamically by the end user at the time of actually mining the database. This model also naturally allows sharing of the KDD process by multiple users.

A KDD System based on I-MIN model is referred as I-MIN system in the remaining part of the paper.

4 Functional Components of I-MIN System

Implementation of I-MIN model for KDD process essentially requires developing components to accumulate, mine, experiment and monitor. These components need to be developed for each type of knowledge, e.g. *Association rules, Classifications, Clustering etc.*[2]. Each component effectuates either one step or a functionality of the I-MIN model. However, a combination of more than one components may be required to accomplish diverse functionality. We propose five components necessary to achieve desired functionality of the I-MIN model.

$$\left\langle \alpha^{\mathcal{IM}}(K_A),\ F_{acc}^{\mathcal{IM}}(K_A),\ F_{min}^{\mathcal{IM}}(K_A),\ F_{exp}^{\mathcal{IM}}(K_A),\ F_{mon}^{\mathcal{IM}}(K_A)\right\rangle$$

Where K is the type of knowledge discovered using algorithm (say) A, $\alpha^{\mathcal{IM}}$ is the "merge" operator required to engineer the user specified subset of the database, $F_{acc}^{\mathcal{IM}}$ is the accumulation component, $F_{min}^{\mathcal{IM}}$ is the actual mining component. $F_{exp}^{\mathcal{IM}}$ and $F_{mon}^{\mathcal{IM}}$ support experimentation and monitoring respectively and may use $F_{min}^{\mathcal{IM}}$ and $\alpha^{\mathcal{IM}}$. The core of a fully implemented I-MIN system is a collection of components for different type of knowledge, for different mining algorithms as shown in Figure 4. We discuss below each functional component in detail.

[2]Further, by supporting different mining algorithms for each type of knowledge, I-MIN system can offer a wide choice to the users for knowledge discovery.

1. **Accumulation Component:** This component performs analysis and partial aggregation on pre-mined data. This component performs step IM2d shown in Figure 3. The aggregation function is defined at the time of schema design, when the mining algorithm is specified. The *Accumulation* component is automatically invoked by the I-MIN system to construct *windows* for the incremental database. It is noteworthy that an end user is transparent to this component.

2. **Merge Component:** *Intension Mining* scheme allows user to dynamically decide the target subset of database for mining, which is specified in terms of the time span for the growing database. In order to prepare the desired window in the database at the time of mining, KCs for the designated period need to be merged to create a temporary wider window. Merge component provides facility to merge two or more *windows* to derive the target subset of database for mining.

3. **Mining Component:** This component consists of the actual mining algorithm used for knowledge discovery. It is invoked during the *Mining* phase, when the user executes a mining query/application. The mining parameters and constraints are supplied to the mining algorithm at this stage. There may be more than one executable function in the mining component for an algorithm. Each function may discover intended knowledge with a different flavor or format. For example, in *Classification* task there may be different sub-components for mining; one for inducing *Classification Tree* the other for *Classification Rules*. This component forms the basis for $F^{\mathcal{IM}}_{exp}$ and $F^{\mathcal{IM}}_{mon}$.

4. **Experimentation Component:** This component of I-MIN model supports user-centric data exploration and experimentation. Repeating experiments with different constraints, subsets of data repository, focus or other relevant parameters provides functionality for experimentation with the KDD process. By meaningfully embedding the desired functionality with the basic services provided by $F^{\mathcal{IM}}_{min}$ and $\alpha^{\mathcal{IM}}$, it is possible to design new experiments in form of user applications to meet specific requirements. As evident from Figure 4, some experimenting sub-components may provide functionality for monitoring also.

5. **Monitoring Component:** Monitoring component of I-MIN system facilitates auditing of data characteristics by comparing and contrasting the knowledge discovered in different *windows*. Multiple sub-components may be tailored to meet user-specific monitoring requirements. The execution of this component is subject to authorization checks[3] at the time of invocation. This component is very powerful and has tremendous potential for revealing the patterns of change. The *windows* created by the KCs naturally accommodate FOCUS framework developed by Ganti et al. [9], for quantifying the deviation in the patterns discovered from two windows or data sets.

The strength of the model stems from the last two components, which are instrumental for user-centric nature of the model. Note that the omission of $\alpha^{\mathcal{IM}}, F^{\mathcal{IM}}_{exp}, F^{\mathcal{IM}}_{mon}$ reduces I-MIN model to traditional KDD process model.

[3]Since monitoring activity has privacy aspects associated with it, only authorized user with proper access can use this component.

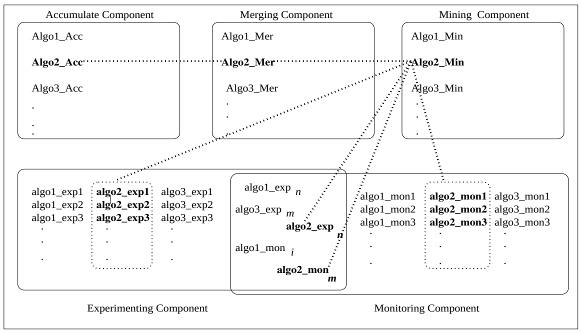

Figure 4. Functional Components of an I-MIN system supporting multiple algorithms for data mining

5 Operators for Intension Mining

Each functional component described in the previous section is a set of functions, specific to the knowledge type and mining algorithm. These functions are accessible to the users as operators in a declarative query language called *Intension Mining Query Language*, and as corresponding APIs in user applications [2]. *In all further references to the term "operator", "API" is automatically intended, unless otherwise mentioned.*

There is one operator each for $\alpha^{\mathcal{IM}}$ and $F^{\mathcal{IM}}_{acc}$, while $F^{\mathcal{IM}}_{min}$ may have multiple operators to provide diverse functionality. $F^{\mathcal{IM}}_{exp}$ and $F^{\mathcal{IM}}_{mon}$ may also map to more than one operators, each providing unique functionality. Operators are logical in nature and are mapped to appropriate functions during compilation of either schema for *Accumulation* or application/query for *Merging* and *Mining*. This mapping is required in view of the support for discovering diverse knowledge types using different algorithms.

A set of primary operators provides basic functionality for constructing windows i.e. *Accumulation*, re-sizing the windows as per the user requirement i.e. *Merging* and discovering knowledge i.e. *Mining*. The "ACCUMULATE" operator is not accessible to the user and is invoked by the system process during *Accumulation* phase. The "MERGE" operator, invoked at the system level to construct the window of size specified in the user applications is also transparent to the user. Mining operators are the only primary operators that can be explicitly addressed by the user in queries and applications, subject to authorization.

Secondary operators provide functionality for exploring/comparing/contrasting two or more subsets of the data set. Some of the secondary operators allow storage and retrieval of knowledge discovered earlier, while others provide processing capabilities. These operators are instrumental in creating the environment for user-centric mining and sharing the knowledge by multiple users simultaneously, and provide functionality for knowledge management. Like primary operators, they can be invoked either through query command or embedded in the applications using APIs. Design of *Intension Mining Query Language* for association rule mining and classification, and development of mining applications has been illustrated in [2].

Primary and Secondary operators for association rule mining have been reported in [10].

6 Three Layered Architecture of I-MIN System

The three-layered architecture proposed for I-MIN system is shown in Figure 5. The architecture is inspired by the DBMS three layered architecture proposed by CODASYL committee [6] and described in several DBMS books [7, 14]. Chief motivation for I-MIN architecture has been to abstract the complete KDD process and provide efficient environment for knowledge management. Independent of the type of underlying database, domain and platform, the architecture supports knowledge discovery, knowledge preservation, knowledge renewal and knowledge sharing, which are considered to be significant aspects of knowledge management[15].

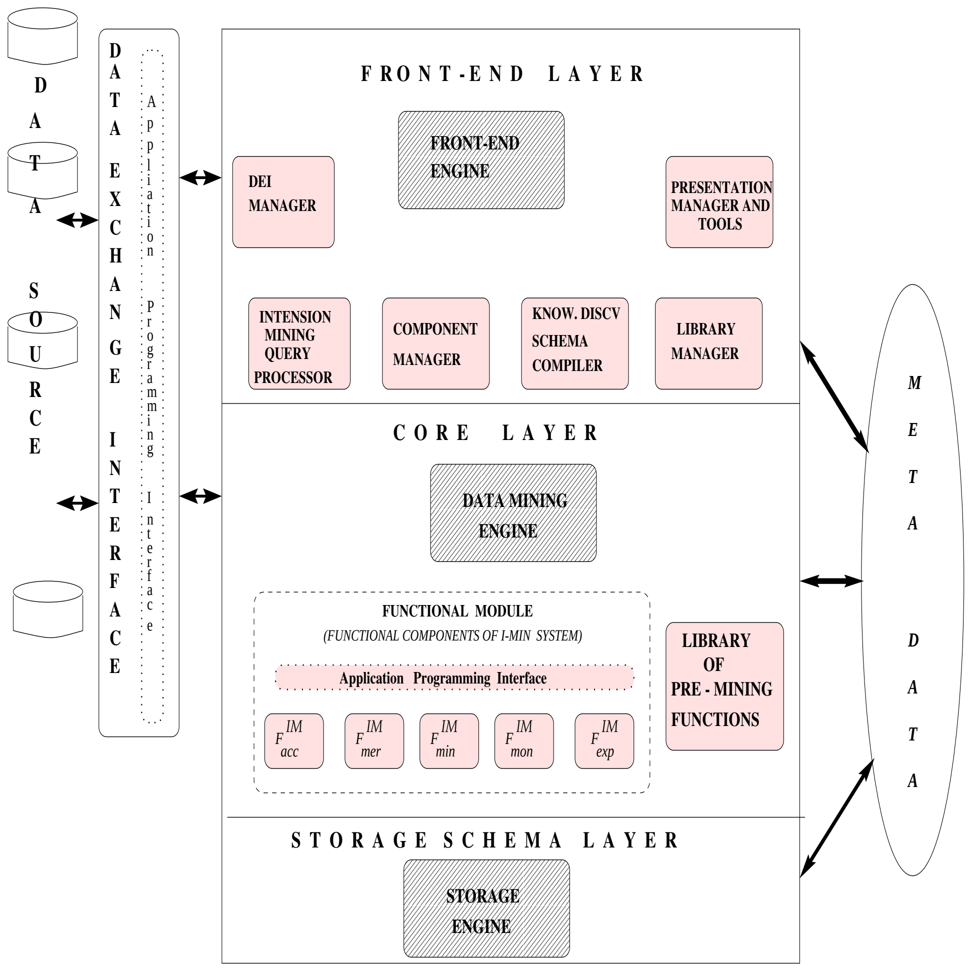

Figure 5. Architecture of I-MIN System for KDD

The top layer called the *Front-End* layer forms the user interface. It provides functionality at the *Planning* and *Mining* phases[4]. The middle layer, which is the *Core* layer is instrumental in carrying out the *Accumulation* and *Mining* phases. The functional components of the I-MIN system are located in this layer. A library of the pre-mining functions is also present. The bottom layer is *Storage Schema* layer, which takes care of the storage of KCs and mappings between KCs and the schemas. It plays an important role during *Accumulation* and *Mining* phases.

Each layer has an *Engine* which maintains the layer level database and, coordinates the other components of the layer. All the three layers access and share *Meta data* stored corresponding to each schema. The *Data Exchange Interface* provides mechanism to access the data source on which mining is sought.

The architecture provides abstraction of knowledge and the KDD process.

Abstraction of Knowledge

[4]There is no user interaction during the *Accumulation* phase.

The knowledge aggregated from the evolving database increments is stored on the secondary storage as units of condensed knowledge. The *Storage Schema* layer provides lowest level of abstraction by describing how this knowledge is stored in data structures and files. The *Knowledge Discovery Schema* at the middle level, assisted by *Storage Schema Layer* abstracts these units of condensed knowledge as *Knowledge Concentrates* (KC) or *windows* [2]. The schema provides conceptual abstraction of the knowledge by providing mapping to all the KCs. Applications that use the desired KCs provide abstraction at the highest level. The user defines query specific view of the subset of the target database to be mined in terms of these *window*s. *Data Exchange Interface* hides the database and access related details from the end user.

The ability to modify the physical data structure or the files of KC, without affecting either the mapping or the applications provides physical data independence.

Abstraction of KDD Process

Recall that each *Knowledge Discovery Schema* points to a collection of KCs and defines one generic KDD endeavor. The complex details regarding pre-mining and aggregation, storage and mapping of KCs are hidden from the user by the middle and lower layers. The user's KDD process is derived from the generic KDD process defined by the schema. Formulation of a mining query or application at the top layer describes the KDD process in the end user's context. Each mining query realizes a specific KDD (sub)process. At an instance, the generic process supports as many sub processes as the number of mining queries or applications using the schema. All the users sharing the schema share the same generic KDD process. An application completes the KDD process.

The ability to modify the KDD process by altering pre-mining functions or mining algorithm without affecting the applications provides logical data independence.

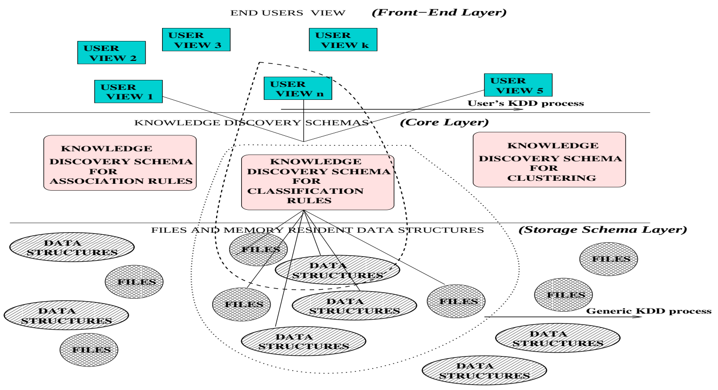

Figure 6. Data and Process abstraction provided by the three layers; The dotted closed figure describes one generic KDD process and dashed closed figure defines individual KDD (sub)process

Figure 6 shows the abstraction provided by the proposed architecture. The middle layer contains three different schemas for mining *Association rules, Classification rules* and *Clusters* from possibly different data sources. The *Storage Schema* layer is populated by various memory resident data structures and files, storing the condensed knowledge from increments of the corresponding data sources. These units are logically mapped to *Knowledge Concentrates* by each schema. The dotted closed figure represents a generic KDD process for mining classification rules. The files and data structure in the referred closed figure denote the sequence of KCs. The top layer contains the user queries and applications, each defining the users view of the KDD process. The query/application corresponding to "USER VIEW n", realizes the KDD (sub) process, involving the schema for classification rules.

6.1 Front-End Layer

The *Front-End* layer provides the user interface for the I-MIN system. The user interaction takes place on account of Schema Design during *Planning* phase, formulation and processing of user application during *Mining* phase and system administration. The layer consists of the following components: i) *Intension Mining Query Processor* : to accept a mining query/application, validate it syntactically and semantically, and construct the execution plan for mining request; ii) *Knowledge Discovery Schema Compiler*: to enter and validate the *Knowledge Discovery Schema* designed by the *Knowledge Discovery Administrator* [2], compile it and store the compiled schema as *Meta-data*; iii) *Presentation Manager*: to allow maintenance and upgradation of presentation tools; iv) *Component Manager*: to maintain the database of the functional components of the I-MIN system in the *Core* layer; v) *Library Manager* : to maintain the library of executable pre-mining functions in the *Core* layer; vi) *Data Interface Exchange Manager*: to allow maintenance and upgradation of the *Data Exchange Interface*.

A *Front-end Engine* at this layer maintains a local data base, providing support to all the components of this layer. It coordinates actions of all the components of the *Front-End* layer. The engine also supports the concept of a session and maintains a session log for each user.

6.2 Core Layer

The *Core* layer implements the *Accumulation* and *Mining* phases of *Intension Mining*.

This layer invokes and manages the generic KDD processes defined by the schemas as well as the user KDD (sub)processes. The *Accumulation* phase is executed by an *Accumulation Process* and the mining query is satisfied by a *Mining Process*. These processes are created and managed by the *Data Mining Engine*. At an instant, this layer is populated by exactly one *Accumulation Process* corresponding to each compiled schema entry and one *Mining Process* corresponding to each mining application invoked by the user.

Data Mining Engine is the core component of the system as it invokes the *Accumulation* component and responds to the user queries and application by invoking mining component. This engine is responsible for the task of creating and managing the *Accumulation* and *Mining* processes in the *Core* layer. It also communicates with *Data Exchange Interface*, on behalf of the *Accumulation* processes in order to retrieve data from the target database. Both accumulation and mining processes are independent of each other and can run simultaneously for the same schema.

The *Functional Module* present in the *Core* layer consists of five functional components of I-MIN system described in Section 4. Each component is an independent collection of executable functions corresponding to the mining algorithms supported by the system as illustrated in Figure 4. All sub-components of a functional component provide similar functionality.

The *Library* of pre-mining functions for selection, cleaning and transformation operations is available in the *Core* layer. With growth of the KDD operations in an organization, new KDD requirements may arise and new functions for data cleaning, data selection and data transformations may be added in the library.

6.3 Storage Schema layer

The main objective of *Storage Schema* layer is to provide efficient access to the data requirements of various processes in the *Core Layer*. The services of this layer are used by accumulation process for storing the KC and by mining process for retrieving KCs while merging and mining. This layer is instrumental in providing physical data independence to the user applications.

6.4 Meta-data and Data Exchange Interface

The *Meta-data* for all the compiled *Knowledge Discovery Schema*[5] entries is stored in the system. It is used for knowledge discovery and, restricted reuse and sharing of knowledge. Since *Meta-data* documents the entire set of KDD operations in the organization, it becomes an important point of control for knowledge management.

The *Data Exchange Interface* is instrumental in achieving the goal of independence of the KDD process with respect to the data source. For supporting mining of new data types, the interface can be augmented with new access methods with the help of *DEI Manager* in the front-end layer.

6.5 Other Issues in I-MIN Architecture

Other issues that need to be addressed for smooth execution of the KDD processes in I-MIN system include privacy and security related policies, backup and recovery, design of languages for schema definition, query language etc..

7 Implementation of I-MIN System

Implementation of the complete I-MIN data mining management system is a gigantic task. However, feasibility of the design and data mining/knowledge management functionality can be demonstrated by designing functional module, mining operators and schema compiler. We have designed and implemented I-MIN framework for association rule mining and classification. Due to space constraint, we are unable to include the details in this paper. Interested readers may refer to [2] for design of a query language and implementation of functional module, operators etc..

8 Conclusion

In this paper we proposed a user-centric model (I-MIN model) for KDD process, and an architecture for a data mining management system based on it. Motivated by the three tier architecture of DBMS, it is an endeavor toward a mining platform extending support for knowledge management, cataloging all the KDD endeavors in the organization. Mining operators are provided to develop applications to meet ongoing knowledge needs of the organization. The architecture permits knowledge discovery in platform and domain independent manner, and knowledge preservation, knowledge renewal and knowledge sharing for effective knowledge management.

[5]Recall that the schema contains the specification of the generic KDD process.

References

[1] S. S. Anand, B. W. Scotney, M. G. Tan, et al. Designing a Kernel for Data Mining. *IEEE Expert Systems and Their Application*, 12(2):65–74, Mar 1997.

[2] V. Bhatnagar. *Intension Mining: A New Approach to Knowledge Discovery in Databases.* PhD thesis, Jamia Millia Islamia, New Delhi, India., 2001.

[3] R. J. Brachman and T. Anand. The Process of Knowledge Discovery in Databases . Chapter 2 in [8], 1996.

[4] C. Brunk, J. Kelly, and R. Kohavi. Mineset: An Integrated System for Data Mining. In *Proceedings of 3rd Int'l. Conf. on Knowledge Discovery and Data Mining*, 1997.

[5] CRISP-DM Homepage. CRoss Industry Standard Process for Data Mining. http://www.crisp-dm.org.

[6] Data Base Task Group. *CODASYL DBTG data model.* DBTG, 1971.

[7] C. J. Date. *An Introduction To Database Systems.* Addison-Wesley Longman, 1999.

[8] U. M. Fayyad, G. Piatetsky-Shaperio, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery in Databases.* AAAI/MIT Press, 1996.

[9] V. Ganti, J. Gehrke, R. Ramakrishnan, and W.-Y. Loh. FOCUS : A Framework for Measuring Differences in Data Characterstics. In *Proceedings of 18th Symposium on PODS*, 1999.

[10] S. K. Gupta, V. Bhatnagar, and S. K. Wasan. User-Centric Mining of Association Rules. In *Workshop on Data mining, Decision Support, Meta learning and ILP , PKDD'2000*, Sept 2000.

[11] J. Han et al. DBMiner: A System for Data Mining in Relational Databases and Data Warehouses. URL:http://www.cs.sfu.ca/DBMiner.

[12] IBM. Intelligent Miner. Data Mining Package; See http://www-4.ibm.com/software/data/iminer/fordata/.

[13] T. Imielinski and H. Mannila. A Database Perspective on Knowledge Discovery. *Communications of the ACM*, pages 58 – 64, Nov 1996.

[14] H. F. Korth and A. Silberschatz. *Database System Concepts.* McGraw-Hill International Editions, 1986.

[15] A. Macintosh. Knowledge Management. http://www.aiai.ed.ac.uk/ alm/kamlnks.html.

[16] C. J. Matheus, P. K. Chan, and G. Piatetsky-Sahpiro. System for Knowledge Discovery in Databases. *IEEE Trans. on Knowledge and Data Engneering*, 5(6), Dec 1993.

[17] G. Psaila. *Integration of Data Mining Techniques and Relational Databasaes.* PhD thesis, Politecnico de Torino, 1998.

[18] T. Reinartz. *Focusing Solutions for Data Mining.* LNAI - 1623, Springer Verlag, 1999.

[19] SAS Inc. Enterprise Miner. Data Mining Package; See http://www.sas.com/.

[20] A. Virmani. *Second Generation Data Mining: Concepts and Implementation.* PhD thesis, Rutgers University, NJ, USA, 1998.

[21] G. J. Williams and Z. Huang. Modelling The KDD Process. TR-DM-96013, CSIRO Division of Information Technology, CPO Box 664, Canberra, ACT 2601, Australia. email: Graham.Williams@cbr.dit.csiro.au, 1996.