INTRODUCTION

Knowledge discovery is defined as “the non trivial extraction of implicit, unknown, and potentially useful knowledge of the data” (Fayyad, Piatetsky-Shapiro, Smyth, & Uthurusamy, 1996, p. 6). According to these principles, the knowledge discovery process (KDP) takes the results just as they come from the data (i.e., the process of extracting tendencies or models of the data), and it carefully and accurately transforms them into useful and understandable information. To consider the discovery of knowledge useful, this knowledge has to be interesting (i.e., it should have a potential value for the user; Han & Kamber, 2001).

Current data mining solutions are based on decoupled architectures. Data mining tools assume the data to be already selected, cleaned, and transformed. Large quantities of data are required to provide enough information to derive additional knowledge (Goel, 1999). Because large quantities of data are required, an efficient process becomes essential.

With the idea of efficiency, intension mining was born. Gupta, Bhatnagar, and Wasan proposed the architecture and framework.

Intension mining arises as a framework that focuses on the user of the current KDP. The basic idea behind the concept of intension mining is to separate the user from the intricacies of the KDP and give him or her a single database management system (DBMS)-like interface to interactively mine for the required kind of knowledge. The user can plan the data mining needs beforehand and input them in the form of knowledge discovery schema (KDS). The system mines knowledge and presents the results in the required format. Intension mining leads to efficiency and makes the whole process more realistic, user-friendly, and, hence, popular (Goel, 1999). As a result, intension mining is a logical extension of incremental mining, with an oriented paradigm to the user, who establishes and conceives the requirements of the mining before the mining begins (Gupta, Bhatnagar, & Wasan, 2000a, 2000d).

BACKGROUND

There are countless contributions to improve and understand KDP. The concept of a second-generation data mining system (Imielinski & Mannila, 1996; Virmani, 1998) involves rule generation, data-rule management, and rule postprocessing. Another extension includes providing users with the ability to remember past mining sessions. Virmani (1998) developed a design called discovery board, which provides a framework for DBMS-like environment supporting query language and APIs to build data mining applications.

Imielinski and Mannila (1996) proposed an evolution of KDP with an SQL-like interface for ad hoc mining. Meo, Psaila, and Ceri (1998) suggested architecture strongly coupled with an SQL server. Ganti, Gherke, and Pamakrishman (2000) developed DEMON, a system based on an incremental mining paradigm; with DEMON, it is possible to mine the entire data repository or some selected subset.

The work being done in the field of structured data mining and upcoming database ideas, such as Hippocratic databases, share various levels of commonality with the I-MIN model in their core ideas (Gupta et al., 2000c). An extensive and explicative coverage with other researches can be found in Gupta et al. (2000a) and Gupta, Bhatnagar, and Wasan (2001).
Once planned, the objectives of data mining are kept in the form of a KDS. Besides capturing the requirements, the functionality of this scheme is to provide the user with a friendly interface, improving the user’s productivity due to its understanding (Gupta et al., 2000a).

As the database framework’s outlines contain the specifications of the relations, the KDS contains the specification of the mining requirements. The outline of knowledge discovery guides the selection, cleaning, transformation, and aggregation processes of the data before mining and, due to the readiness of the requirements in the KDS, the system is able to execute off-line pre-mining operations periodically. This information should be preserved in secondary storage in an appropriate form to be used to satisfy the mining queries performed by the user (Gupta et al., 2000a). Thus, the mining in the base can be carried out on the basis of demand, using this information.

An important characteristic of intension mining is that it perceives the KDP as a continuous process. Because the temporary aspect is captured by the operations of periodic premining, experimentation as well as monitoring is possible (Gupta et al., 2000c).

Intension mining, as DBMS architecture (for more detail on DBMS, see Elmasri & Narvathe, 2000), is built up in three phases:

- **Phase 1-Planning**: The aim of phase 1 is to evaluate the objectives of data mining and to formalize them as a KDS. As was previously mentioned, the user anticipates the mining requirements and specifies them in the aforementioned scheme during the planning phase.

A clear understanding of the domain and the requirements of the mining help in designing a KDS, which directs the KDP. The type of knowledge to be mined in the database, the database to be mined, the transformation, the selection, the cleaning, the specific mining algorithm to be executed, and finally the presentation tools to be used are specified by the user in the KDS (Goel, 1999). The metadata is stored to be used in the accumulation and mining phases. Just as a good database outline design can efficiently satisfy the users’ queries in most occasions, a well-thought-out KDS would be able to efficiently discover different types of knowledge in most instances (Gupta, 2000). At this level, security, backup, and recovery issues also arise in the database (Gupta et al., 2000a).

- **Phase 2-Accumulation**: Phase 2 starts after compilation of the KDS and continues until the user decides to drop the mining requirements (schema) altogether. During this accumulation phase, the incremental database is premined and aggregated in consultation with the metadata to yield knowledge concentrate (KC), which stores the intermediate form of intended knowledge (Gupta et al., 2001).

The crucial parts of this level are the interaction between the database and the KDP to get to the registers of data and the maintenance of the KC in the secondary storage. The extraction of the KCs, starting from the incremental database, represents the intensive task of I/O in intension mining and it can endure several scannings of the data. Significantly, all these tasks are carried out off-line (Gupta et al., 2000a).

In conclusion, the presence of the trade-off that the KCs imply can be observed. Although KCs allow new functionality to be added, because the mining, when working on them is speeded up, they have a cost when occupying extra space. In the ideal case, all the tuples of the database will be the same, and so will the small structures; but, in the worst case, all the tuples are different; they can occupy a great quantity of space. One should then evaluate what is convenient for the user in each case.

- **Phase 3-Mining**: Mining is the final phase of the system. In general, during this step the KDP is intensive. The phase of mining begins when a user invokes a mining query in the user’s interface (Goel, 1999). The user specifies the parameters when carrying out the query. This offers the user the freedom to explore the database and to experiment with the KDP. The query is processed, and the mining is done on the specific structures of data, which are kept in the KC (Gupta, 2000).

The mining process consults the KDS, and it executes the algorithm of the specified mining on the cumulative KCs during the accumulation phase. Finally, the results are presented by means of the presentation tool (Gupta et al., 2000a).

Those response times are also better because the I/O is avoided and is giving the possibility of carrying out an exploratory analysis, choosing the subset of data, or varying the parameters. As the mining is carried out on the KCs, it does not interfere with the operations of the database.

**I-MIN Model: An Instantiation of Intension Mining**

The pattern has been designed to support intension mining. Thus, it is developed in three layers, according to the basic ideas mentioned in Gupta et al. (2000a).

The architectural proposal emulates a DBMS-like environment for the managers, administrators, and end users in the organization. Knowledge management functions, such as sharing and reusing of the discovered