Integrating KDD algorithms and RDBMS code

María C. Fernandez-Baizán*, **Ernestina Menasalvas Ruiz**,
José M. Peña Sánchez, **Borja Pardo Pastrana**
{cfbaizan, emenasalvas}@fi.upm.es, {chema, borja}@orion.ls.fi.upm.es

Departamento de Lenguajes y Sistemas Informáticos e Ingeniería del Software,
Facultad de Informática, Campus de Montegancedo, Madrid

Abstract. In this paper we outline the design of a RDBMS that will provide the user with traditional quey capabilities as well as KDD queries. Our approach is not just another system which adds KDD capabilities, this design is aimed to integrate these KDD capabilities into RDBMS core. The approach also defines a generic engine of Data Mining algorithms that allows easy enhancement of system capabilities as a new algorithm is implemented.

1 Introduction

Most of the KDD systems that have been implemented up to the present moment apply just one particular methodology or implement a particular algorithm (rough sets[9], attribute-induction[5], apriori[1,2]). When designing this architecture we wanted a system that integrates data mining capabilities within the RDBMS. We wanted the system to be extensible, that is, we wanted to build a system in which adding new algorithms would be easy. This goal is achieved dividing KDD algorithms into basic operations that will be implemented as particular instance of a structure that has been called *operators*.
The paper is organized as follows: The division of KDD algorithms into basic operations is explained in section 2 as well as the main structure that operators must have in order to be included in the system. Extension of main modules of traditional database systems to handle new operations is discussed in section 3.

2 Algorithms

It is easy to observe that many KDD algorithms have similar behaviour during, at least, an important part of their execution. This fact has led us to consider the division of the algorithms into several parts, in order to achieve reusability of code. Moreover, as one of the goals of the design is the integration with a RDBMS, we have tried to make each of those parts as similar to RDBMS basic operations as possible. In our desing the basic operations will be performed by

* This work is supported by the Spanish Ministry of Education under project PB95-0301

the **operator** structure. We will call operator any operation that is made up of: Relational tables both as **Input** and **Output**, **Auxiliar Structures** that will be used to keep input/output information of the operation and **Parameters** than guide their behaviour. The main result of this process could be represented as **Extracted Information**. In figure 1 the basic structure of an operator is depicted. A data mining query will then be defined as the secuence of operators

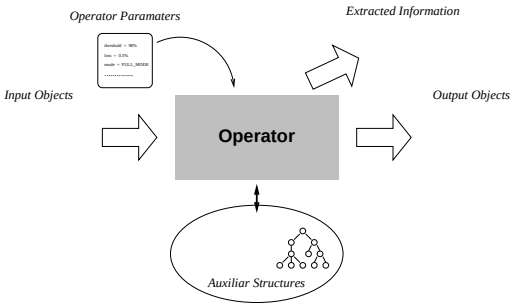

Fig. 1. Operator interface

that, given a particular table containing certain objects, gets as a result the set of patterns to describe the knowledge asked by such a query.

3 Architecture

The decomposition of algorithms into operators guided the design of the system we plan to build. The identification of these simple components could be used to split most KDD queries into atomic elements that could be managed directly by a new RDBMS. This system may be named as **RDBMAS** (**R**elational **D**ata**B**ase **M**anagement and **A**nalysis **S**ystem) and would be designed (see figure 2) as an extension of traditional RDBMS with new capabilities.
In figure 2 we show how a generic traditional RDBMS would have to be modified. As a result next generation of database systems would provide the user with traditional queries as well as the posibility to analyze data.

- **Query Analyzer**: This module parsers each submited query to the system and translates it into a internal representation. In traditional RDBMS this module analyzes only SQL commands, for RDBMAS it will have to analyze also new KDD sentences.
- **Optimizer**: This component gets internal sentence representation supplied by the previous module and optimize the order of execution of its clauses. As a result returns a specific execution plan for the user query. If new operations provide the optimizer with the same measures traditional operations do (weights, restrictions, ...) this module would not have to be changed significantly.

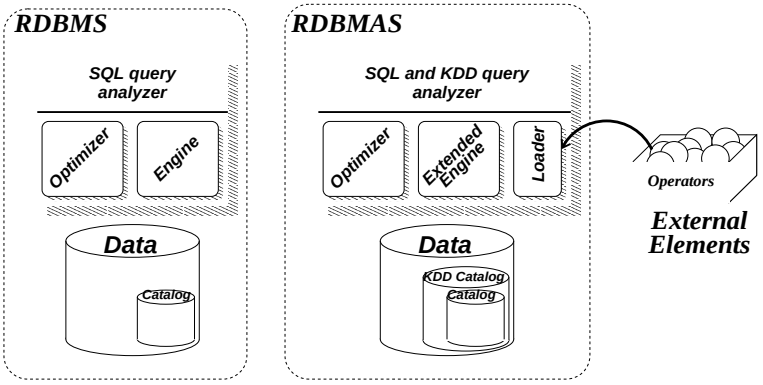

Fig. 2. Traditional RDBMS and proposed RDBMAS

- **Engine**: This module carries out execution plan. In RDBMAS this execution engine must be able to complete SQL sentences and new KDD queries.
- **Loader**: In order to achieve KDD queries some new code must be executed in the RDBMAS. Almost all KDD systems provide their functionalities with static defined code. In this system all algorithms must be split into elemental operations and may be implemented in external dynamically-loaded components (**operators**).
- **Catalog**: Information about data (metadata) is stored as aditional tables for RDBMS. This information is required for management functions, but a new information about data may also be necesary in order to support analysis (KDD capabilities) functions. So, the system catalog must be extended.

3.1 Comparison with other systems

Approaches like Data Surveyor [6] propose similar system architectures that modify RDBMS in order to achieve better performance, but in most cases KDD algorithms run outside of these enhanced RDBMS. With our design all queries (SQL queries and KDD queries) are managed by the same RDBMAS.
RSDM [4] has been conceived as an engine of KDD algorithms instead of a system that adds some particular capabilities. This approach has its advantages as well as disadvantages. On the one hand, the idea of building an engine of algorithms in contrast to all the existing Data Mining systems, will allow to add new capabilities with the only task of building the module that will execute such capability. This avoids the complex process of codifying programs for an integrated system in which you have to care not only of the coding of the algorithm but of the communication, storing of intermediate results and so on. On the other hand, the process of construction of the architecture is more complex. However, we must emphasize once again that adding any capability will be a straighforward task once the architecture has been finished.

3.2 Conclusion and future research

The design of a new generation of database systems that will provide the user with query and analysis of data has been outlined in this paper. We will called these system RDMAS.
At the present moment the design is being further studied to tackle the problems that are arising as a result of addition the new capabilities. Also implementation of a first prototype has just began. We hope to have in the near future a prototype available.
We have to remark once again that the implementation of the operation inside the core of the RDBMS is twofold on the one hand efficiency gain due to maximuzation of optimizer functions on the other next generation of RDBMS will provide users with data analysis capabilities.

4 Acknowledgements

We are very much indebted for inspiration to Dr. Ziarko and Dr. Pawlak.

References

1. Agrawal R., Imielinski T., Swami A. *Mining association rules between sets of Item in large Databases*, Proceedings of ACM SIGMOD, pages 207-216, May 1993.
2. R. Agrawal, *Mining Association Rules Between Sets of Items in Large Databases*, In Prooceedings of ACM SIGMOD Int. Conf. on Management of data, Washington DC, pp. 207 - 216, 1993.
3. R. Agrawal et al., *The Quest Data Mining System* In Proceedings The Second Int. Conf. on Knowledge discovery and Data Mining, pp. 244-249. August 1996
4. M.Fernandez Baizan, E. Menasalvas, J.M. Peña. *Integrating RDBMS and Rough Set Theory* To Appear in Fuzzy Databases in August 1998
5. J. Han et al., *DBMiner: A System for mining knowledge in Large Relational Databases* In Proceedings The Second Int. Conf. on Knowledge discovery and Data Mining, pp. 250-255. August 1996
6. M.L. Kersten, Arno P.J.M. Siebes, *Data Surveyor: Searching the nuggets in parallel* Advances in Knowledge Discovery and Data Mining, AAAI Press, pp 447-467
7. J. Komorowski, A. Ohrn, *ROSETTA: A Rough Set Toolkit for Analysis of Data* In Proceedings JCIS 97, pp. 403-407. March 1997
8. Z. Pawlak, *Rough Sets - Theoretical Aspects of Reasoning about Data*, Kluwer, 1991.
9. Z. Pawlak, *Information Systems-Theoretical foundations*, Information systems, 6, No.4, 1993, pp. 299 - 297.
10. G. Piatesky-Shaphiro, *An Overview of Knowledge Discovery in Databases: Recent Progress and Challenges*, Rough Sets, Fuzzy Sets and Knowledge Discovery, 1994, pp. 1-11
11. A. Skowron, C. Rauszer, *The Discernibility matrices and Functions in Information System*, ICS PAS Report 1/91, Technical University of Warsaw 1991, pp. 1-44
12. W. Ziarko, *Variable Precision Rough Sets Model* Journal of Computer and System Sciences, vol. 46. 1993, 39-59.
13. W. Ziarko, N. Shan *On Discovery of Attribute Interactions and Domain Classificactions*, CSC'95 23 Annual Computer Science Conf. on Rough Sets and Data Mining