Intelligent Mediation in Active Knowledge Mining: Goals and General Description

Patricia Lynch Carbone
Artificial Intelligence Technical Center
The MITRE Corporation
7525 Colshire Drive
McLean, VA 22102, USA
and
Center for Artificial Intelligence
George Mason University
carbone@mitre.org

Larry Kerschberg
Department of Information
and Software Systems Engineering
and
Center for Artificial Intelligence
George Mason University
4400 University Drive
Fairfax, VA 22030, USA
kersch@isse.gmu.edu

Abstract

Recent years have seen the increasing development of knowledge discovery or database mining systems that combine database management technology with machine learning techniques and algorithms to perform the analysis of data. Many of these systems use passive database management systems to hold the data rather than active databases. However, applications such as battle management situations or stock market trading would benefit from the use of an active database management system since the data is being constantly updated and other actions should be triggered based on database events. In this paper, we present a general description of an active knowledge mining system that combines an active database with machine learning operators. We introduce the notion of an intelligent mediator that contains knowledge about both the database and the capabilities of the learning operators. The mediator chooses which of the operators to use to achieve a learning goal, then determines how the discovered knowledge should be used and where it should be stored and maintained.

1. Introduction

With the development of database management technology, government and businesses have been developing applications to store and retain increasing amounts and types of data. However, the capability of a human analyst to keep up with the amount of data being stored is limited. Therefore, recent years have seen the increasing research and development of knowledge discovery or database mining, which is the combination of database management technology with machine learning techniques and algorithms to obtain knowledge from the data ([PF], [PS]).

While much of the knowledge mining research is based on the use of passive databases, very little research has focused on the use of active databases in the knowledge mining process. A passive database is one which merely stores data or responds to user queries but does not react to user-defined or system events in the database. An active database, however, allows the user to construct rules which respond to database events, triggering other processes to occur (see [HW] for an overview). For example, one may define rules to act as consistency constraints, notifying an administrator if data exceeds a defined threshold. Rules may also be used to monitor incoming data, triggering separate applications if the data matches defined conditions.

Current knowledge mining research assumes that the data in the database will be relatively static, not requiring the use of an active database to respond to database updates. Examples of such applications include analysis of the census data taken every ten years or market analysis of sales over a given period of time to determine how to improve performance or whether to expand into new markets. However, there are a number of applications that would benefit from the use of an active database. For example in battle management applications, the system should perform a threat assessment capability. In a stock market analysis, either trading programs should be activated or deactivated, or a stock broker should assess the impact of buying and selling on his customers' portfolios. The combination of knowledge mining with the active database could presumably aid in the refinement of the active rules or triggers, thus providing a better assessment of the situation.

Most of the "passive" knowledge mining systems currently under development assume that a user controls the session, selecting the data to be analyzed, selecting the learning operator to be employed, then determining the relevancy and validity of the discovered knowledge. However, in the active applications mentioned above, a user does not necessarily have the luxury of time to control the analysis. Rather, analysts may find that they cannot keep up with the amount of data being processed and stored in the database. Therefore, the use of a partially to fully automated intelligent control mechanism is necessary to choose which learning operator should be used next to help keep up with the analysis tasks, as well as to decide where the discovered knowledge should be stored for future use by the system.

In this paper, we present a general description of an "active" knowledge mining system. The system will contain an active database management system and several machine learning operators. Expanding on the idea of mediation in [Wie], we introduce the notion of an intelligent mediator which contains knowledge about both the database and the capabilities of the learning operators. The mediator is an interface between the machine learning algorithms (or operators) and the active database; it chooses which of the operators should be used to achieve a learning goal and then determines how the discovered knowledge should be used and where it should be stored and maintained.

2. Background

A number of knowledge mining systems are currently under development, including DBLEARN ([CC], [HC]), IMACS [Br], and INLEN ([Ka1], [Ka2], [MK]). Each of these systems merge a passive database management system with one or more machine learning algorithms to derive rules to classify the data. The user selects a subset of data and passes it to the machine learning operator. Based on the results, the user may choose either a new subset of data to be analyzed or choose a different algorithm to be used. Another approach to knowledge mining or "data dredging" uses a deductive database as the basis for the mining system [Ts]. The user creates hypotheses about the data and presents them to the system. The database then uses deduction to prove or disprove the hypotheses. All of these systems require that the user be heavily involved in the analysis process, so that the user can guide what is being discovered, rather than potentially learning uninteresting rules.

The following "Learning Engineering" process model — based on the knowledge-based system development life-cycle model in ([WK1], [WK2]) — has been established [Ke] in support of the INLEN system. (See Figure 1 for a diagram of this process model.) The data or knowledge administrator or user must first plan for the learning process, including accessing and combining data from multiple knowledge sources and databases and reconciling data and knowledge incompatibilities. Goals for the learning process should also be established. Hypotheses regarding the data must be generated either by an administrator or automatically by the system. These hypotheses are then tested against the currently stored data and knowledge. At this point, the knowledge discovery process may begin by integrating the learning algorithms, the data, and potentially any previously discovered knowledge. The newly discovered knowledge must be validated and judged as to its relevancy against the domain and the existing data. Finally, there will be an evolution of the knowledge schema and potentially the data schema as the knowledge is incorporated back into the system and background knowledge is perhaps revised. The entire process may become iterative, as the goals for learning may change, based on the discovered results.

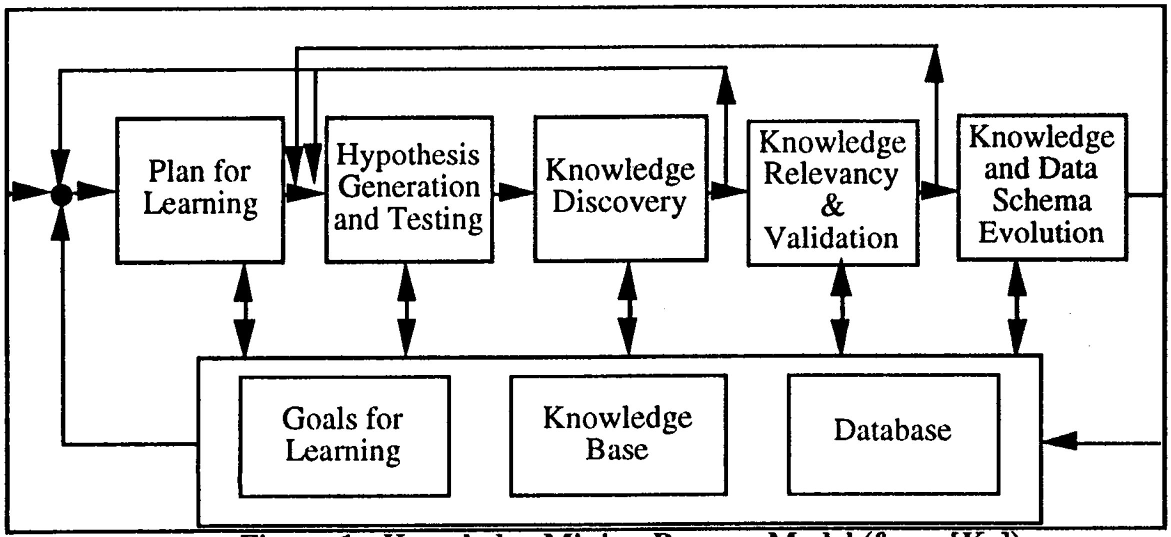

Figure 1 - Knowledge Mining Process Model (from [Ke])

While the process model was initially developed with the thought of using passive databases, we intend to build a system based on the process model using an active database management system. Active databases merge a production rule system with database management capabilities into one system. Rules are triggered by database events, such as updates, and queries. The rules are stored and maintained in the same manner as the data. Systems such as POSTGRES ([SR], [SJ]) and Starburst [Wid] have been developed and have achieved some success as research systems. Recently, some commercial systems have begun to include the active component in their architectures (e.g., INGRES).

We propose that an active knowledge mining system can be built using the process model as a basis. In developing a knowledge mining system with an active database, one wants to take advantage of the triggering mechanisms of the database and the changing nature of the data. Therefore, it would be advantageous to automate as much of the knowledge mining process model as possible. Obviously, some interaction with an administrator or user will be necessary. However, with the current research being carried out in the area of multistrategy learning [MT], we believe that an automated system will be possible.

A potential method for developing an active knowledge mining system is through the use of a mediator. The idea of mediation was first proposed as a method for helping to resolve differences between heterogeneous, distributed systems [Wie] and allow users to query those databases in a specific domain more easily. It was also suggested that the mediator could become an interface between the database(s) and the machine learning algorithms. We have built upon this idea to propose an intelligent mediator that acts both as the interface between the database(s) and the machine learning operators and as an intelligent agent to choose which operator is best for the analysis, given a specified domain.

We expect that the mediator will receive data from the database. The active database will trigger the mediator periodically, sending it data that has changed over some threshold or that has not been seen in a certain time period. We expect that current research into the area of quasi-caching will be incorporated into our architecture (e.g. [SK]) so that the mediator's data and information remains consistent with the database(s). Based on this data and on information stored in the mediator, a machine learning operator will be chosen and initiated. To make the decision, the mediator will have knowledge regarding the active database, the machine learning algorithms, and the goals of the learning session. In addition to the data which is passed to the particular operator, the mediator would eventually be able to provide knowledge about other heterogeneous databases that might provide additional background knowledge for the learning process. When the learning operator obtains some knowledge, the mediator must determine the relevancy and reliability of the discovered knowledge. The mediator will store the discovered knowledge, possibly suggesting changes both to the database schema and to the goals of the learning process.

We expect to incorporate into the mediator some of the research on multi-strategy learning in order to enable it to choose the algorithm as intelligently as possible. For example, analysis of the performance of the machine learning algorithms can aid in an adaptive control of the learning session ([Ho]). An introspective analysis of how the learning is progressing could help the mediator to refocus the session while it is occurring [CR]. In addition, genetic algorithms have been used to refine rules through their interaction with the environment [SG]. Providing the mediator with this type of control can aid in the learning session and can help in performing relevancy and validity checks.

3. System Description

An architecture of our proposed system is shown in Figure 2. It is made up of four major components: an Active Database, a Knowledge Discovery System, a knowledge base containing the Discovered Knowledge, and the Intelligent Mediator. Each of these will be described in turn.

The Active Database is a "typical" active database system, such as POSTGRES or Starburst. The database maintains the real world facts from the external observations in the database extension. Triggering rules and other metadata are stored in the database intension. The Production System interacts with the extension and intension, firing rules that are triggered by the database events. Users, application programs, and other systems may interact with the Active Database.

The Knowledge Discovery System holds the Machine Learning Operators or algorithms and the knowledge necessary for them to operate. The necessary knowledge may include background knowledge for the operators, domain knowledge of the application, and other metadata. Examples of possible Machine Learning Operators are discussed in [Ka1]. Because of the changing nature of the data, it is assumed that the preferable machine learning algorithms chosen to be operators will be incremental learning algorithms, allowing changes to the discovered rules as new data becomes available. As incremental machine learning algorithms can have full memory, partial memory, or no memory of the previous learning, some experimentation will have to be done to determine which version of memory storage will provide accuracy without causing undue delay in the processing time.

The Discovered Knowledge subsystem is a knowledge base that will hold the results of the learning process. We expect that the information will have been checked for relevancy. However, there may be varying degrees of reliability of the information stored within, depending on whether certain discovered information needs more substantiation.

The Intelligent Mediator provides the control for the system, becoming an interface between the Active Database and the Knowledge Discovery System. The Learning Objective contains the goal or goals of the learning process. The Meta-Data contains information about the contents of the Active Database and about the application in general. Data and knowledge administrators can interact with both the Meta-Data and the Learning Objectives, as well as the

Intensional Database and the background or domain knowledge in the other two subsystems. The Strategies/Solutions component in the Intelligent Mediator contains both information about the capabilities of the machine learning operators and any of the full or partial rules discovered by the operators.

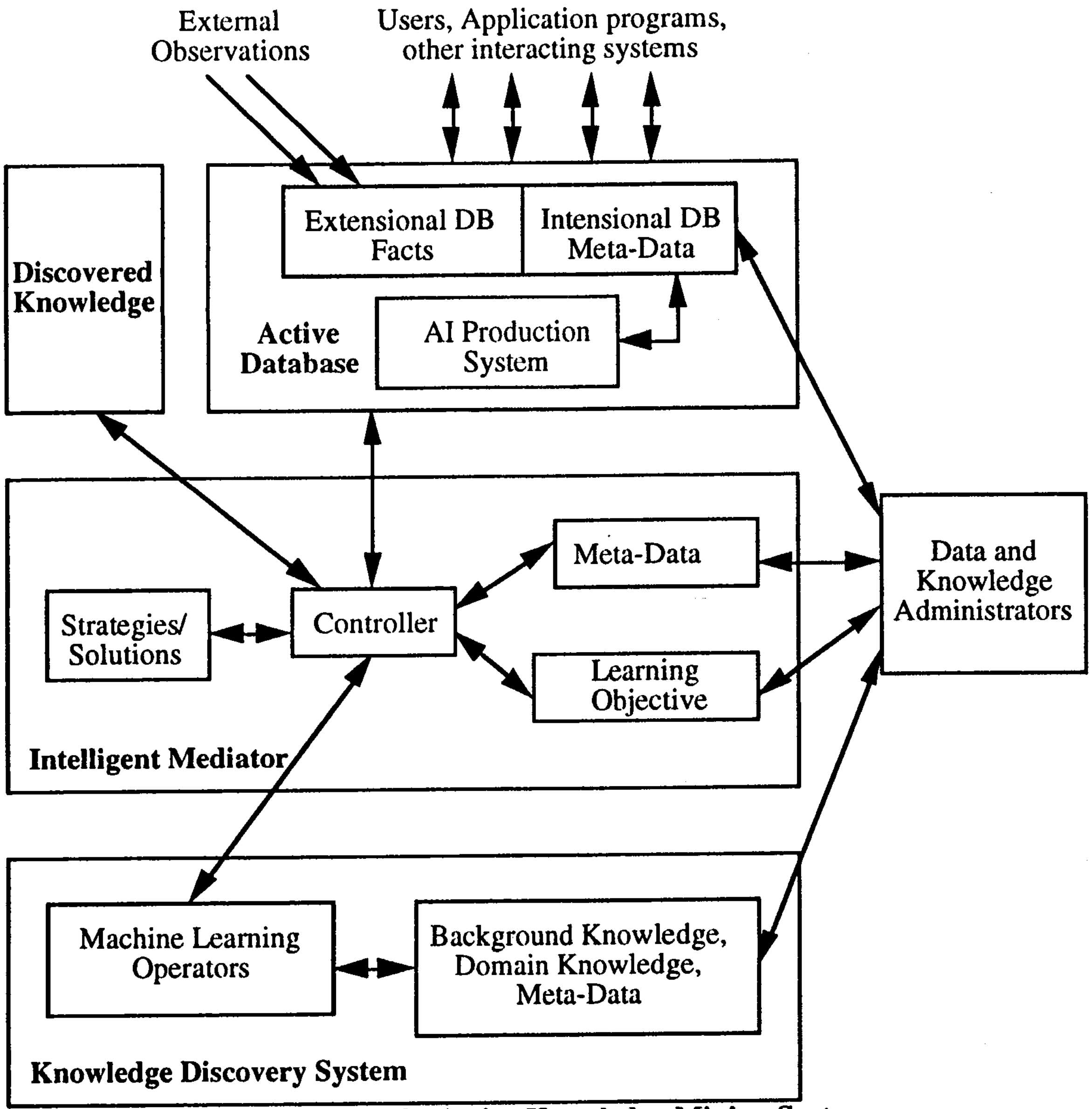

Figure 2 - Active Knowledge Mining System

The Controller interacts with several of the other subsystems, providing the interface between them. It interacts with the Active Database by receiving data from that subsystem and passing validated rules back to the Intensional Database. The Controller interacts with the Knowledge Discovery System by choosing an appropriate operator to be activated and giving it data, then receiving any discovered rules. The Controller also interacts with the Discovered Knowledge subsystem by placing knowledge that is determined to be relevant there. In addition to its interactions with outside subsystems, the Controller is responsible for deciding where to store the discovered knowledge within the Mediator, for determining the relevancy and validity of the discovered knowledge, and for modifying its own metadata and strategy information based on how well the particular learning operators are performing.

We envision that the Intelligent Mediator could be constructed using a blackboard architecture ([EM], [LL]). The Controller has knowledge regarding the capabilities of the agents (or machine learning operators). The operators work on solving a problem or a goal, writing those full or partial solutions to a common area that is shared by all the operators (similar to the Strategies/Solutions component). The Controller can then redirect the learning based on newly received data or on the full or partial knowledge being written to the common area. Multiple operators could work in parallel to achieve the learning goals with the Controller being the unifying agent.

We provide an example in the form of a stock market analysis system. The system would be used, say, by a brokerage firm whose goals for the application are to (1) make money for their investors and (2) make money for the firm. The Active Database Extension that was described in Figure 2 maintains current and historical data on price changes in the stock market and data on the investors and their portfolios. Figure 3a shows an example of tables that could be used in the database extension. Rules in the Intensional Database may include descriptions of when to buy or sell stock for certain types of investors or for the firm in general. For example, if a stock price drops a certain amount, rules may cause a search of the investor data for those investors that might be interested in purchasing that stock. Meta-data in the Active Database may include additional investor characterizations and other domain information. Figure 3b shows some example rules that could be used for the brokerage application. There would also be additional rules, separate from those for the application, that would trigger data to be sent to the Mediator. Figure 3c shows examples of those rules that could be triggered.

Figure 3a - Active Database Extension:

CURRENT_MARKET: (Company_name, price_per_share, PE, Dividend, Low, High, Volume_traded, %_institutional_holding, Industry_grouping)

PAST_MARKET: (Company_name, daily_closing_price, Date, Dividend, Low, High, ...)

INVESTOR_INFO: (Investor_name, Company_name, nr_shares, current_price, current_dividend_yield, %_return, buy_date, buy_price, dividend_yield_at_buy, %_return_at_buy, sell_date, sell_price, commission, portfolio_yield, portfolio_value)

Figure 3b - Active Database Intension - Brokerage Application Rules and Meta-Data:

<u>Brokerage Application Rules</u> (written in pseudo-code):

```
IF CURRENT_MARKET.price_per_share INCREASES_BY > x% THEN
    RECOMMEND_SELL(Aggressive_Investor,
        CURRENT_MARKET.Company_name)

IF CURRENT_MARKET.price_per_share INCREASES_BY > y% THEN
    RECOMMEND_SELL(Conservative_Investor,
        CURRENT_MARKET.Company_name)

IF CURRENT_MARKET.price_per_share DECREASES_BY w% THEN
    RECOMMEND_SELL(Conservative_Investor,
        CURRENT_MARKET.Company_name)

IF CURRENT_MARKET.Dividend INCREASES_BY > z% THEN
    RECOMMEND_SELL(Income_Oriented_Investor,
        CURRENT_MARKET.Company_name)
```

Meta-Data (written in pseudo-code):

Income_Oriented_Investor: INVESTOR_INFO.portfolio_yield - current_inflation_rate > x /* Income-oriented investors are concerned with having a good dividend yield */

Conservative_Investor: For majority of stocks in portfolio, growth of portfolio AND portfolio_yield > current_inflation_rate /* Conservative investors care about long-term growth of their portfolio and a reasonable dividend yield */

Aggressive_Investor: For majority of stocks in portfolio, sell_date - buy_date < 18 months /* Aggressive investors do not hold stocks for long periods, as they are more concerned with short-term buying and selling for profit */

Figure 3c - Active Database Intension - Mediator Triggering Rules:

Mediator Triggering Rules (written in pseudo-code):

```
IF New_High(CURRENT_MARKET.price_per_share) THEN
    Send_to_Mediator(
        CURRENT_MARKET where CURRENT_MARKET.Industry_grouping =
            Current(industry_grouping);
        PAST_MARKET where PAST_MARKET.Industry_grouping =
            Current(industry_grouping))
IF CURRENT_MARKET.Volume_traded - PAST_MARKET.Volume_traded > z THEN
    Send_to_Mediator(
        CURRENT_MARKET where CURRENT_MARKET.Industry_grouping =
            Current(industry_grouping);
        PAST_MARKET where PAST_MARKET.Industry_grouping =
            Current(industry_grouping))
```

Figure 3d - Learning Objectives:

Does a new high indicate that others in the industry grouping will follow?

Are there relationships or trends that occur between industry groupings?

Does a new high indicate that a pattern is occurring?

Figure 4 provides a diagram of the information flow for the brokerage application. Based on the rules in the Intensional portion of the Active Database and the incoming data on market changes, one or more rules may be triggered, both for the brokerage application or for data to be sent to the Mediator. These rules may be based on the data changing over a certain threshold, an exception being noted, or a time constraint being met, as shown in Figure 3b and 3c. The rules in Figure 3c cause a subset of data to be sent to the Controller in the Intelligent Mediator. The Controller determines which learning operator should be initiated using the data sent from the Active Database, the learning objectives, and knowledge regarding the machine learning operators and their capabilities. The Learning Objective in the Intelligent Mediator holds the learning goals, which are to find stock market purchases that will make money for both the investors and the company. These goals should have smaller subgoals that are defined by the Data and Knowledge Administrators. For example, Figure 3d shows examples of goals that could be used to indicate what type of learning should occur. The Controller will attempt to find a goal by matching attributes in the data received to attributes of the learning goals. In the process of choosing a strategy, the Controller may request additional data from the Active Database to augment that which has already been sent. For example, if the data received pertains to stocks within an industry grouping that have reached new highs and the chosen Learning Objective is to look for relationships or trends between industry groupings, the Controller will request stocks from a different (but related group) to determine if a relationship exists.

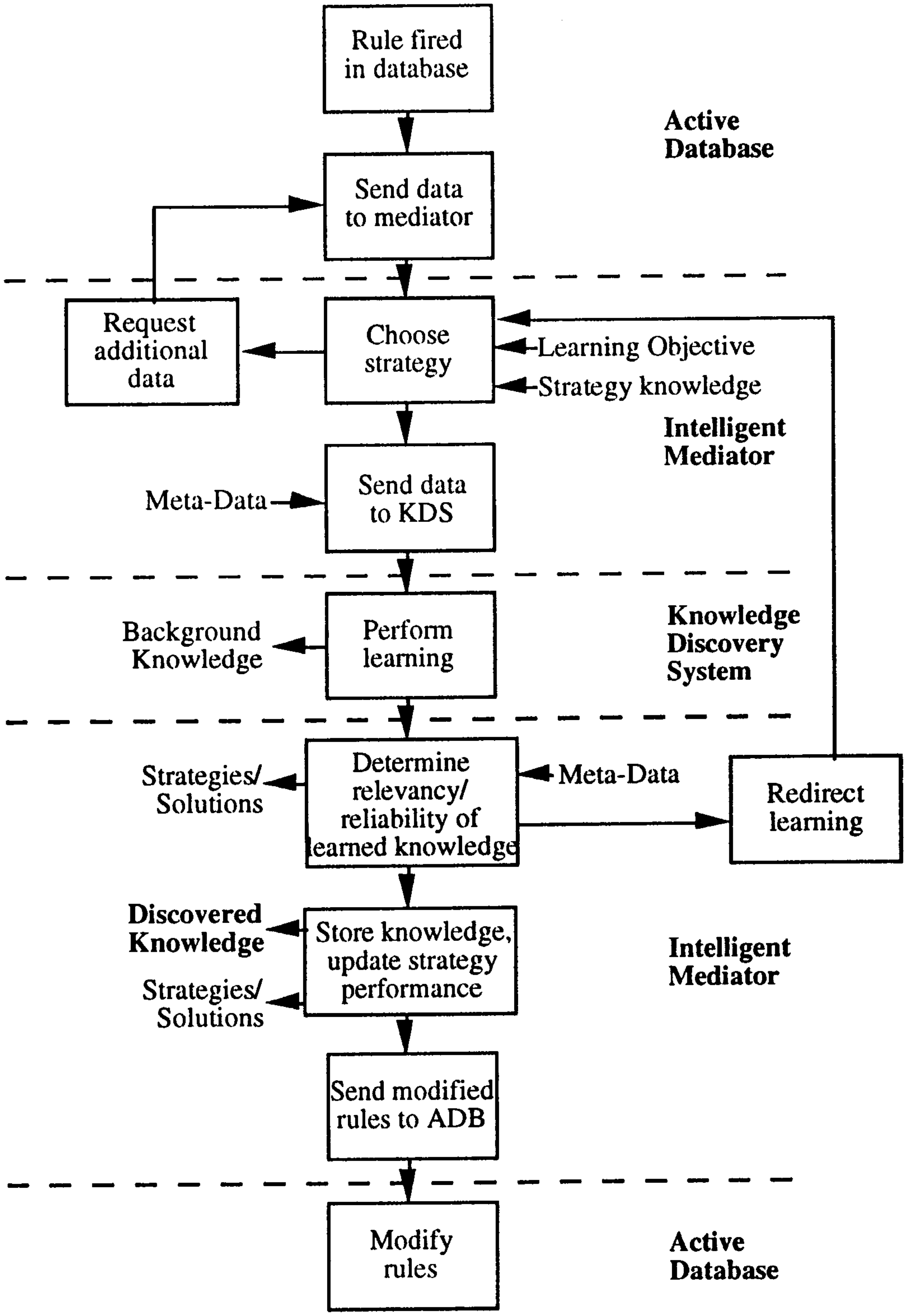

Figure 3 - Information flow of Active Knowledge Mining

When a strategy is chosen, the Controller sends the data to the Knowledge Discovery System, along with any additional data or meta-data that may be required by the particular strategy. The Meta-Data in the Intelligent Mediator may include rules to resolve conflicting goals and, particularly in this application, data about related database information. For example, other information could include financial information about the companies (e.g. earnings per share, long-term debt, net income, shares outstanding, and industry subgroupings), government approval (e.g. rate increases), known relationships to comparable stocks, other analyst recommendations, and other actions (e.g. interest rates, utility approval, OPEC prices, and price of gold). This meta-data would serve the other function of the mediator as described by [Wie], in that the mediator would serve as an interface between a number of other heterogeneous databases. The related databases could contain information regarding the corporations' financial data, new product announcements, government interest rate changes, OPEC prices, and commodities information. Through the use of the Meta-Data, the Mediator could query the potentially heterogeneous databases and augment the learning process by pulling in pertinent data and sending that data or information from the Meta-data to the Knowledge Discovery System.

Having received all of the data from the Controller, the particular machine learning operator in the Knowledge Discovery System is initiated. Background knowledge in the Knowledge Discovery System used by the operator could include general rules regarding stock performance, other stock market analysis rules, and any other domain knowledge. When some knowledge is discovered, either fully or partially, it is returned to the Controller in the Mediator. The Controller must then determine whether or not the knowledge is relevant and also perform some reliability checks, again using stored Meta-Data. The reliability checks may actually need to be performed over a longer period of time, to see whether other database changes substantiate the discovered knowledge. For example, if one stock in an industry grouping reaches a new high, there may not yet be an indication that the other stocks will follow suit. However, over time the other stocks in the industry group could follow, thus creating a trend. For this reason, the discovered knowledge may be stored in either the Strategies/Solutions portion of the Mediator, particularly if the discovered knowledge is incomplete, or in the Discovered Knowledge subsystem. In addition, if it is desired to maintain performance records on the machine learning operators, that information is stored in the Strategies/Solutions portion of the Mediator.

Based on the knowledge that is returned, the Controller may decide to redirect the learning, either because the discovered knowledge is incomplete or because additional characteristics should be analyzed. In this case, the Controller begins to choose another strategy, thus creating a cycle. For example, the chosen learning algorithm may not discover a trend if one particular stock has reach a new high and the others have not followed. However, if the Controller requests data on stocks in a related industry (e.g. chip manufacturers related to computer hardware manufacturers related to software vendors), a trend may be discovered there. It is this loop that could potentially be performed in parallel, with multiple machine learning operators working simultaneously on either the same or different subsets of data and goals. In this case, the Controller must collate the discovered knowledge into a usable form. For example, discovered rules might include those such as "A successful company is indicated by its price reaching a new high" or "Successful stocks are those that are in demand by institutional buyers" or "The best performing companies are those whose annual compounded growth rate of earnings is greater than x%." Finally when the Controller determines that the discovered knowledge may affect the rules stored in the Active Database, those rule modifications are sent back to the Active Database where they are then stored. Therefore, these three rules may be added to the brokerage application rules to trigger the investors to potentially buy the stock when there is a large volume buy by an institution, when the stock hits a new high, or when a company's growth rate exceeds the threshold rate.

4. Knowledge Representation

As one can see from the architecture and from the example, there is a great deal of knowledge that must be stored and maintained in the active knowledge mining system. Figure 3b and 3d showed the type of knowledge that might need to be represented in the brokerage application. The different types of information are presented below.

Active Database - Intensional DB Meta-Data. The Active Database requires information about the specific application domains that interact with it so that the rules in the Intensional portion of the database may trigger the correct actions. The active rules control actions and reactions of the database to specified events by either causing other changes within the database, alerting an operator, or sending data to an outside application. Therefore, if more than one application is using the database, one can imagine that there will be subsets of rules that pertain mainly to one specific application that uses the Active Database. In addition, the database will need to have the standard schema information and database dictionary necessary for maintenance.

Intelligent Mediator - Meta-Data. Since the Meta-Data will aid the Controller, it must contain information about the active database, both in terms of its structure and contents as well as its interactions with the outside applications. In this way, the Mediator may effectively modify the Active Database rules when necessary without violating assumptions made by the particular applications. The Meta-Data must contain knowledge that will aid the Controller in determining the relevancy/reliability issue. Criteria for these checks may initially be entered by a data or knowledge administrator, but the system should be able to update those criteria if the need arises. Finally, the Meta-Data should contain knowledge about other heterogeneous knowledge-bases and databases accessed by the Mediator, such as the Discovered Knowledge subsystem. Using this information, the Controller should be able to pull in information from other databases for use by the Machine Learning Operators or by the Controller itself.

Intelligent Mediator - Learning Objective. The Learning Objective component will maintain the goals of the learning process. As with the Meta-Data, the data or knowledge administrator may initially define or load those goals and subgoals for use by the Controller. However, we envision that as time goes on, the Mediator should be able to reason about those goals and at least suggest changes to the knowledge administrator, if not actually implement the changes directly.

Intelligent Mediator - Strategies/Solutions. This component will maintain knowledge regarding the capabilities of the various machine learning operators. At the present time, we envision that the capabilities will be represented as strengths and weaknesses of the operators with regard to the learning task at hand. In addition, if we choose to follow the ideas of [Ho] which would provide adaptive control of the learning session, we need to store information regarding the performance of the learning operators. Using this information, the Controller could then determine at what point in the learning process a particular operator should be used and for what duration. As with the Learning Objective, the information about operator capabilities is expected to change over time.

Knowledge Discovery System - Background knowledge. Many of the machine learning operators require some kind of background information, particularly if an incremental algorithm is being used. This background or domain knowledge may include information about past knowledge that has been discovered, or possibly partial solutions that are simultaneously being worked on by other operators. Relevant knowledge may initially be input by the knowledge administrators. However, it is envisioned that the Intelligent Mediator Controller may augment this knowledge with additional information and that the operators themselves may update the information.

5. Ongoing Research and Future Work

There are a number of research issues that we are addressing in the development of our system. The first is to determine the method for representing the knowledge that is needed by the various components of the system. This is particularly important, since we want to use the knowledge about the learning goals and the capabilities of the machine learning operators to aid in the task of choosing a strategy.

The next issue is to determine how and when the Controller should change learning strategies. We intend to work from the current research in this area that was described earlier, along with our ideas of using the operator capabilities.

It is important that the Controller provide an operator with a complete subset of data for analysis. Therefore, we must allow the Controller to request possibly additional subsets of data in addition to that provided initially by the Active Database. Obviously, this issue involves determining what attributes would be considered "similar," both to the existing subset of data, and to the learning goals.

The issue of determining the relevancy and the reliability of the discovered knowledge is also very important to this system. Many of the other knowledge mining systems circumvent these problems by requiring that the user solve these issues. Relevancy may also be determined by performing the learning on only those attributes that are used in a query. At this point, we envision that the data or knowledge administrator may have to input a ranking of attribute relevance and a list of reliability checks. Another method of ascertaining reliability of discovered knowledge is through the incremental learning and through substantiation by future discovered knowledge. The issue of incorporating discovered knowledge back into the knowledge base has been addressed in [YK], and these techniques may prove useful in this active database environment.

We are currently working on the knowledge representation issues and on developing a more detailed architecture for the system. We have two different active databases available, so we intend to experiment with each to determine which is more suitable for our application. We also have a number of machine learning algorithms available, so we are looking at which might be most suitable for an initial system.

We feel that knowledge mining is an important step forward in the information management area. Since active databases are becoming available commercially, it seems a natural extension to use their assets in knowledge mining research. Research into mediation also seems promising and especially suitable for handling some multistrategy learning problems. Using these three components, we have constructed an architecture for an active knowledge mining system. The creation of a system such as this is a viable and necessary step in the evolution of knowledge mining systems.

Acknowledgments

This research was conducted in the Center for Artificial Intelligence at George Mason University. The Center's research is supported in part by the National Science Foundation under grant No. IRI-9020266, in part by the Defense Advanced Research Projects Agency under the grant No. N00014-91-J-1854, administered by the Office of Naval Research, and the grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, and in part by the Office of Naval Research under grant No. N00014-91-J-1351. This research is also supported by a DARPA grant, N0014-92-J-4038 administered by the Office of Naval Research .

6. References

[Br] Brachman, R.J., Selfridge, P.G., Terveen, L.G., Altman, B., Borgida, A., Halper, F., Kirk, T., Lazar, A., McGuinness, D.L., and Resnick, L.A., "Knowledge Representation Support for Data Archaeology," *Proceedings of the ISMM International Conference on Information and Knowledge Management*, Baltimore, MD, November 1992.

[CC] Cai, Y., Cercone, N., Han, J., "Attribute-Oriented Induction in Relational Databases," in [PF].

[CR] Cox, M., and Ram, A., "Using Introspective Reasoning to Select Learning Strategies," in [MT].

[EM] Englemore, R.S., and Morgan, A.J., Blackboard Systems, Addison-Wesley Publishing Company, Inc., 1988.

[HC] Han, J., Cai, Y., Cercone, N., and Huang, Y., "DBLEARN: A Knowledge Discovery System for Large Databases," *Proceedings of the ISMM International Conference on Information and Knowledge Management*, Baltimore, MD, November 1992.

[HW] Hanson, E.N., and Widom, J., "An Overview of Production Rules in Database Systems," IBM Research Report RJ 9023 (80483), IBM Research Division, Almaden Research Center, San Jose, CA, October 1992.

[Ho] Holder, L.B., "Selection of Learning Methods Using an Adaptive Model of Knowledge Utility," in [MT].

[Ka1] Kaufman, K.A., Michalski, R.S., Kerschberg, L., "Mining for Knowledge in Databases: Goals and General Description of the INLEN System," in [PF].

[Ka2] Kaufman, K.A., Michalski, R.S., Kerschberg, L., "An Architecture for Integrating Machine Learning and Discovery Programs into a Data Analysis System," in [PS].

[Ke] Kerschberg, L., "Methods for Knowledge Discovery, Integration, and Evolution in Knowledge- and Data Bases," presentation to DARPA, February 6, 1992.

[LL] Lander, S.E., and Lesser, V.R., "Customizing Distributed Search Among Agents with Heterogeneous Knowledge," *Proceedings of the ISMM International Conference on Information and Knowledge Management*, Baltimore, MD, November 1992.

[MK] Michalski, R.S., Kerschberg, L., Kaufman, K.A., and Ribeiro, J.S., "Mining for Knowledge in Databases: The INLEN Architecture, Initial Implementation and First Results," *Journal of Intelligent Information Systems*, Vol. 1, No. 1, August 1992, pp. 85-113.

[MT] Michalski, R.S., and Tecuci, G., Proceedings of the First International Workshop on Multistrategy Learning, Harpers Ferry, WV, November 1991.

[PF] Piatetsky-Shapiro, G., and Frawley, W. (Eds.), Knowledge Discovery in Databases, AAAI/MIT Press, 1991.

[PS] Piatetsky-Shapiro, G. (Ed.), Proceedings of the AAAI-91 Workshop on Knowledge Discovery in Databases, Anaheim, CA, July 1991.

[SK1] Seligman, L., and Kerschberg, L., "Approximate Knowledge-base/Database Consistency: An Active Database Approach," *Proceedings of the ISMM International Conference on Information and Knowledge Management*, Baltimore, MD, November 1992.

[SK2] Seligman, L., and Kerschberg, L., "Knowledge-base/Database Consistency in a Federated Multidatabase Environment," to be presented at the *IEEE RIDE-IMS '93 Workshop*, Vienna, Austria, April 1993.

[SG] Spears, W.M., and Gordon, D.F., "Adaptive Strategy Selection for Concept Learning," in [MT].

[SR] Stonebraker, M., Rowe, L.A., Hirohama, M., "The Implementation of POSTGRES," IEEE Transactions on Knowledge and Data Engineering, Vol. 2, No. 1, March 1990.

[SJ] Stonebraker, M., Jhingran, A., Goh, J., Potamianos, S., "On Rules, Procedures, Caching and Views in Data Base Systems," Proceedings of the 1990 ACM-SIGMOD Conference on the Management of Data, Atlantic City, NJ, June 1990.

[Ts] Tsur, S., "Data Dredging," IEEE Data Engineering *Bulletin*, Vol. 13, No. 4, December 1990, pp. 58-63.

[Wid] Widom, J., "The Starburst Rule System: Language Design, Implementation, and Applications," IBM Research Report RJ 9070 (80883), IBM Research Division, Almaden Research Center, San Jose, CA, November 1992.

[Wie] Wiederhold, G., "Mediators in the Architecture of Future Information Systems," *IEEE Computer*, Vol. 25, No. 3, March 1992, pp. 38-49.

[WK1] Weitzel, J.R. and Kerschberg, L., "A System Development Methodology for Knowledge-Based Systems," *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. 19, No.3, May-June 1989.

[WK2] Weitzel, J.R. and Kerschberg, L., "Developing Knowledge-Based Systems: Re-organizing the System Development Life Cycle," *Communications of the ACM,* Vol 32, No. 4, April 1989.

[YK] Yoon, J.P. and Kerschberg, L., "A Framework for Knowledge Discovery and Evolution in Databases," accepted for publication in *IEEE Transactions on Knowledge and Data Engineering*, Special Issue on Knowledge Discovery in Databases, to appear.