Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-672

Potentials of advanced database technology for military information systems

S. Choenni and B. Bruggeman

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-672

Potentials of advanced database technology for military information systems

S. Choenni and B. Bruggeman*

** Royal Netherlands Naval College (KIM)*

This investigation has been carried out under a contract awarded by the Royal Netherlands Navy (RNLN).
The Royal Netherlands Navy has granted NLR permission to publish this report.

This report is based on a presentation held at the 5th IST Symposium on New Information Processing Techniques for Military Systems, Istanbul, Turkey, October 9-11, 2000.

Division:	Information and Communication Technology
Issued:	June 2001
Classification of title:	Unclassified

Summary

Research and development in database technology evolves in several directions, which are not necessarily divergent. **A** number of these directions might be promising for military information systems as well. In this paper, we discuss the potentials of multi-media databases and data mining. Both directions focus on the handling of a vague information need of a user. In general, data mining systems allow a higher degree of vagueness than multi-media systems. Information systems that are able to handle vague information needs adequately will improve the decision making process of armed forces.

Contents

1 Introduction

Both the military and civil communities have come to the realisation that military information systems fundamentally differ from civil ones in some respects [Ref 6]. The difference started with the views that military and civil communities had on information systems. The military community was and is looking forward to information systems that support the *human decision making process* in a time pressured dynamic environment with multiple, often conflicting, goals, especially for command and control tasks. Databases in these information systems should be able to deal with high volume and of uncertain data at lower levels of the decision chain, and with aggregated information at higher decision making levels [Ref 12]. Furthermore, databases should be able to handle vague information needs. For a long time, the civil community had a less ambitious view on information systems. Civil information systems were developed to help employees in performing a number of tasks, often administrative, in an efficient way. This was realised by handing over routine actions to computers. Databases in civil information systems consisted of well-structured and true data. To retrieve data from these systems one should exactly specify what data is to be retrieved. In the sixties it was even worse, relevant data could only be retrieved if the system was exactly told how to navigate towards the data. As soon as the automation of routine jobs was well understood, research and development in civil information systems became more ambitious. Currently, research in databases evolves in several directions, which are not necessarily divergent [Ref 22]. A number of these directions appears to be promising for military information systems as well. In this paper, we discuss the potentials of multi-media databases and data mining for military information systems. Both directions focus on the handling of a vague information need of the user. In general, data mining systems allow a higher degree of vagueness than multi-media systems.

A multi-media database management system aims to reply to the (vague) information need by combining different (advanced) types of data, such as audio, video, images, etc. We present an architecture that facilitates the storage and access to different types of media servers in an integrated manner. Main properties of the architecture are modularity and extensibility. Depending on the imposed requirements and characteristics of an application, this architecture can be elaborated such that a tailored operational multi-media database system can be built, including a military multi-media database system. We report on the issues that play a role in building a military multi-media system from a database perspective.

As noted before, a second direction that is promising for military information systems is data mining. This direction has seen a recent surge in commercial development. Data mining has as

goal to extract implicit, previously unknown, and potentially useful knowledge from large data sets. Extracted knowledge may support or be used in strategic decision making. In the military world, data mining can be applied to search for correlation between doctrines, to predict how an enemy will act based on large volumes of data collected about the enemy (in the past), to improve internal logistics, etc. To realise this task, data mining combines techniques from the fields of machine learning, statistics, artificial intelligence, and database technology. We discuss the requirements to mine operational databases successfully. Furthermore, we report on the results that we have obtained by mining two aircraft incident databases.

The remainder of this paper is organised as follows. In Section 2, we motivate why advanced database technology might be interesting for military applications. In Section 3, we discuss the challenges and potential solutions for multi-media database systems. In Section 4, we give a brief state of the art in the field of data mining. Furthermore, we report on an application concerning aircraft incidents. Finally, the paper is concluded in Section 5.

2 Military benefit

As stated in the introduction, military applications put complex demands on information systems, especially when information systems have to operate in non co-operative environments. Taking the developments in the operational field into account, we expect that the demands that a military information system will have to meet may become even more complex. We note that these demands are partly yielded by technological achievements.

To meet the military demands on information systems, the application of advanced information technology, including database technology, as underpinning is inevitable. For example, the manoeuvrability of aerial targets has increased. An impact of this increase is that the performance of classical methods for trajectory tracking and prediction —which, in general, are based on simple dynamic models that do not exploit knowledge about a "situation"— suffers [Ref 7]. So, the development of alternative trajectory and predicting systems that include data and knowledge bases, which are able to handle fuzziness and uncertainty, is becoming interesting.

Today, a number of trends may be distinguished in the world in which armed forces are operating. To mention some of these trends:

- There is a growing need for joint and combined operations. In order to perform these operations successfully, information from different and heterogeneous sources should be processed and integrated in a consistent manner [Ref 12].
- Due to budget cut, military organizations (in many countries) should work more efficiently [Ref 26]. One way to achieve this is to automate a large number of tasks that is currently performed by human beings.
- Shorter reaction times are demanded. This means that we require high performance data processing tools [Ref 8].
- Soldiers are or will be equipped with advanced technologies and means, such as microphones, videos, digital maps, etc.
- A wide variety of methods and techniques are applied for intelligence gathering and data processing [Ref 8].

Each of these trends has in common that they strongly rely on information processing and integration techniques. For example, applications of a wide variety of methods and techniques for intelligence gathering will result in a (wide) diversity of information, each with their own characteristics, e.g., with regard to uncertainty measures, data types, etc. The challenge is to combine all information in order to select high quality information and to present this information in a suitable way to the user. For the other above-mentioned trends a similar reasoning holds.

We feel that both multi-media database management systems and data mining systems are useful to successfully implement these trends. We note that well-defined questions to well-defined databases can be handled by standard database technology. Multi-media database management systems are intended to reply on information needs that are vague and incomplete. An example of such a question is: Give me all information that is known about servo XY, in connection with a defect. Data mining systems are aimed to answer more strategic kind of questions, such as: should we consider an enemy that is able to perform actions X and Y as dangerous or not?

In the two successive sections, we discuss multi-media databases and data mining in more detail.

3 Multi-media databases

Many advanced applications, including military applications, will profit from the integration of information from different media servers. This integration generally leads to a better assessment of a situation; moreover user interactivity will further improve the assessment. Integration and the support of user interactivity are major goals of multi-media databases. Within the scope of soldier modernization programs multi-media databases are promising, since soldiers are equipped with a wide variety of advanced technology and means, such as microphones, videos, maps, etc. Multi-media databases might also be useful in traditional situations as well, e.g., for off-site support. Suppose that in a state of war a ship at sea has problems with its engine and several media servers (including handbooks) contain information about this type of engine, e.g., images of the engine, video fragments how to disconnect/connect several parts of the engine, audio fragments containing the functionality of the several parts of the engine, etc. A multi-media database system may help the crew in solving the problems at the ship by replying adequately to the information need of the crew. Suppose that the first guess of the crew is that something might be wrong with compression and formulates its information need as "give me all information about compression in a combustion chamber". As a reply the system should integrate all information available on different media servers to explain the functions and the use of the different parts of, for example, the combustion chamber in a suitable order. Furthermore, it might also be useful to demonstrate how the compression can be measured and what tools are proper for this purpose. We note that in a state of war it is not always possible to let an expert come from land.

In order to support above-mentioned applications, advanced database architectures and query handling techniques are required. In Section 3.1, we discuss such an architecture and in Section 3.2, we report on the challenges that this architecture entails for query handling.

3.1 Architecture

In order to support multi-media application, we feel that advanced database architectures should provide developers the following components.

- Suitable user interfaces, in which users can express their information need in a human friendly manner.
- A multi-media storage server which provides the capability to store and to access different types of data, e.g., images, video, audio, etc.
- Feature extractors. These extractors might be specialized algorithms that are able to extract relevant features from a piece of data.
- A meta database which contains features about the data stored on the multi-media storage

server, such as color histograms, texture, annotated text, etc.

- A data dictionary which contains relevant information with regard to the whole system.

As depicted in Figure 1, all these components are loosely connected to each other through a network (e.g., Internet), supporting the concept of modularity and extensibility. For example, if a novel feature algorithm has been made available extracting previously unknown features, it can be plugged in into the feature extraction module and the meta database should be extended with a number of attributes or relations to store these features. The remainder of the system can be left untouched. This architecture is suitable for on-site as well as for off-site support for armed forces. A more detailed description can be found in [Ref 25].

To be viable from a database perspective, the architecture has to be supported by advanced query handling techniques that exploit the meta database. The challenge in the field of query handling is to devise techniques that analyse the information need of the user and maps the result to relevant features in the meta database. These features are used to identify the data that meet the information need and to locate the data at the multi-media server. Suppose that a user provides an image and is interested in similar images stored on the server. To answer this information need, a mapping might be on color histograms. The color of the provided image can be computed and compared with the color histograms of the stored images in the meta database. Images with similar color

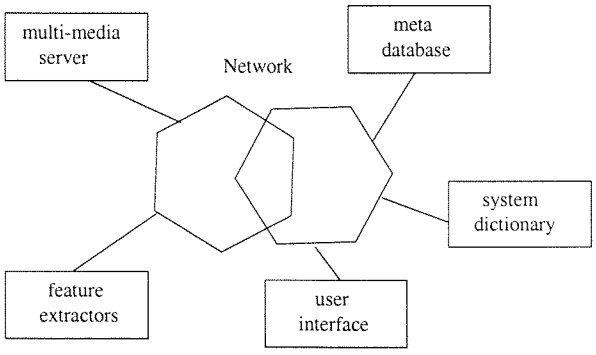

Fig. 1 Architecture of a multi-media database system

histograms are retrieved from the multi-media server and presented to the user.

3.2 Query handling

In order to handle user queries, traditional optimizers[1] produce an evaluation plan for each query. This evaluation plan specifies the actual (basic) operations (i.e., joins, selections, etc.) and the order, in which these operations should be performed. The goal of the optimizer is to choose a

[1] An optimizer is a software module in a database management system that has as goal to speed up the processing of a query. This is mainly achieved by reducing the number of disk accesses.

cheap evaluation plan, primarily in terms of disk accesses. We note that, in general, it is infeasible for an optimizer to search for the cheapest plan, since the problem of query optimization is NP hard. However, query optimizers are performing quite well in finding a cheap evaluation plan for a single query [Ref 14, 24].

For multi-media applications, traditional query optimization techniques are inadequate for the following reasons. First, information needs formulated by a user are not exact as in traditional applications, but rather vague and incomplete. So, an optimizer should be able to handle vagueness and incompleteness.

Second, an information need is often decomposed into a sequence of queries, and current database management systems do not exploit multi query optimization techniques [Ref 5, 10, 20] but are focussed on the optimization of a single query in isolation. Multi-query optimization techniques, which search for a plan of a sequence of queries instead of a single query, may considerably speed up the processing of information needs. For example, a user is searching for pictures of cheerful ladies in uniform (e.g., to promote an airline company). Let us assume that this need is decomposed into two queries: 1) select all cheerful ladies and 2) select all cheerful uniforms. If we have a small set of uniforms, it is better to search for a set of cheerful uniforms first, and then to select pictures of cheerful ladies wearing these uniforms. So, an optimizer should be able to determine the best order in which a query of sequence may be performed.

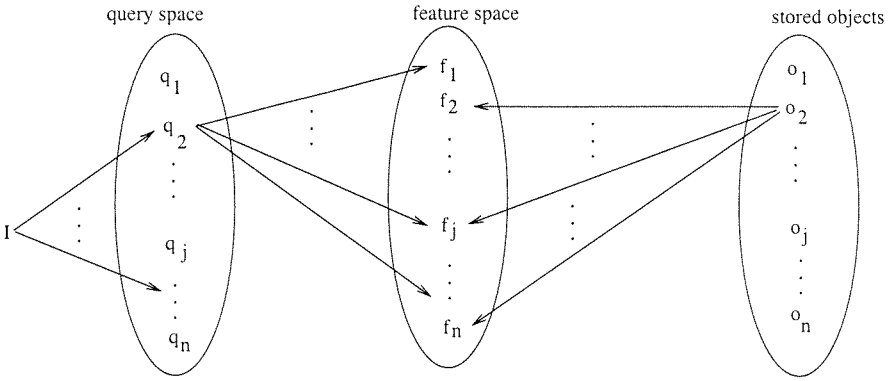

Fig. 2 Mapping of information needs and objects on features

Third, the search for a proper answer to an information need is an interactive process. Suppose that a multi-media database management system comes up with a set of cheerful ladies in uniform as a result on the above-mentioned information need. By means of this output, the user specifies more precisely what in her/his opinion should be understood by cheerful ladies, e.g., by assigning a rating to each picture of the selected set. On the basis of a subset of the pictures (e.g. those with

high ratings), the system searches for pictures that better meets the information need of the user. This does not automatically mean that the originally selected set can be thrown away. It is very well possible that a user later refers on to some pictures that were not interesting at the first glance. Therefore, it might be sensible to store some of the originally selected pictures. The challenge for novel optimizers is to decide what pictures should be stored and for how long. Current database management systems do not take the interactive behaviour of users into account. Next generations database management systems may speed up the processing of information needs by exploiting the interactive behaviour of users.

Although multi-query optimization and the exploitation of interactive behaviour of users are important techniques to speed up the processing of information needs, we feel that an adequate handling of vagueness and incompleteness captured into a specified information need is of crucial importance. If an information need is wrongly interpreted by a database management system — due to inadequate techniques to handle vagueness and incompleteness— compared to what a user has meant, the system will come up with answers that are not relevant. Therefore, we discuss this issue in more detail, and we argue that solutions for this issue are application dependent.

In Section 3.1, we have noted that the meta database (see Figure 1) is exploited in processing an information need. This meta database contains features of the stored objects. In order to process an information need, this is split into a number of queries. Each query is mapped into a set of features, which is used to search for objects that (hopefully) partly meet the information need. The results of all queries are combined to produce the final result to an information need. In Figure 2, we have sketched this process in a simplified form. The main challenge is to map a query into a set of relevant and useful features. Since this is a tough task, we know beforehand that a map on a feature for a query will be partly true and useful. Therefore, proposed solutions assign an uncertainty measure to each map. Different solutions differ in the uncertainty formalism (e.g., Bayesian theory, Dempster-Shafer theory, etc.) they choose [Ref 18, 23]. So, we can distinguish two main problems in the mapping of a query into a set of features.

- How to translate a concept that appears in a query into a set of features (often defined beforehand) that might be quite abstract, such as color histograms, texture, etc. The usefulness of solutions to this problem is application dependent. The better we understand a concept, the better we can translate this concept into a useful set of features. For example, the term data fusion in the military world stresses on the combination of data coming from different sources, while in the database world the same term stresses on the correctness of data to be stored. This also implies that the usefulness of a multi-media database system is determined by how well an application domain is understood by a developer.

- How to measure the "goodness" of a mapping of a concept to a feature, and how to propagate these measurements. We feel that the better we understand the first problem, the better we will be able to handle this problem.

We note that research and development in both above-mentioned fields are still in their childhood.

Another mapping that plays a role in Figure 2 is the mapping of stored objects into features. In the context of information retrieval, a considerable amount of research is devoted to this problem. In the field of information retrieval, one tries to find the relevant documents —given an information need— from a collection of stored documents. A widely accepted model to handle this problem is the following. For each document, a representative set of terms is extracted and stored, e.g. in a meta database. To each term, a term and a document frequency are assigned. The term frequency expresses the number of times that a term t appears in a document d, denoted as $tf(t,d)$. The document frequency expresses the number of documents in which a term t appears, denoted as $df(t)$. The values for $tf(t,d)$ and $df(t)$ are used to retrieve relevant documents.

To determine the probability that a document d is relevant for a given set of terms T we have to compute $Pr(d|T)$. According to Bayes rule $Pr(d|T) \propto Pr(T|d)$ [Ref 16]. To determine $Pr(T = t|d)$, the following formula is widely used

$$Pr(T = t|d) = \alpha_1 \frac{df(t)}{\sum_{t \in T} df(t)} + \alpha_2 \frac{tf(t,d)}{\sum_{t \in T} tf(t,d)}$$

Setting the proper values for α_1 and α_2, which are constants, is often a matter of trial and error.

Extension of the above-mentioned strategy for mapping objects to features is a promising direction in the field of multi-media databases. We note that capturing all mappings of Figure 2 into a consistent framework will be a significant progress in this field.

Since the success of multi-media databases depends on the understanding of the concepts used in an application, it is important that armed forces actively participate in the development of a new generation of military information systems.

4 Data Mining

The large amount of data stored in databases may serve two purposes. First, it may help in understanding a phenomenon, and second it may help to predict the outcome of similar phenomena. In our view, data mining is a powerful tool that contributes to the realization of these purposes.

Data mining has as goal to extract potentially useful information from large databases by using a wide variety of methods and techniques, including statistical ones, that is able to explore large data sets efficiently. In practice it is seldom the case that a proper data set is available that can be directly mined. Therefore, we feel that the following four steps are equally important for effective data mining.

1. A so-called mining question should be formulated. This question should specify the kind of information one is looking for. In general, a mining question is formulated together with domain experts.
2. Then, the data that may be used in order to answer the mining question should be selected, enriched, cleaned, and integrated, i.e., constructing a *data warehouse*. Since intelligence gathering is an important activity in many military applications, data enrichment is a key factor in building data warehouses successfully.
3. A mining algorithm has to be selected/developed that will search the data warehouse for appropriate answers to the mining question.
4. Finally, the answers of the search process should be presented in a way such that domain experts are able to understand and evaluate these answers.

We note that these four steps are defined as *Knowledge Discovery in Databases (KDD)* [Ref 15] and should be iteratively applied. In general, data mining is a highly interactive process. In practice, users start with a rough idea of the information that might be interesting, and during the mining session the user more explicitly specifies, based on, among others, the mining results obtained so far, which information should be searched for.

In Section 4.1, we describe a comprehensive and extensible architecture that supports the above-mentioned steps. Since the mining step provides us arguments to assess the performance of data mining in military applications, we focus on a number of mining techniques in Section 4.2. Then, in Section 4.3, we briefly describe a data mining application that we have performed at our laboratory.

4.1 KDD Architecture

To be viable, a data mining tool should support all the steps distinguished for knowledge discovery. In Figure 3, the architecture of a data mining tool is presented that supports all these steps. This ar-

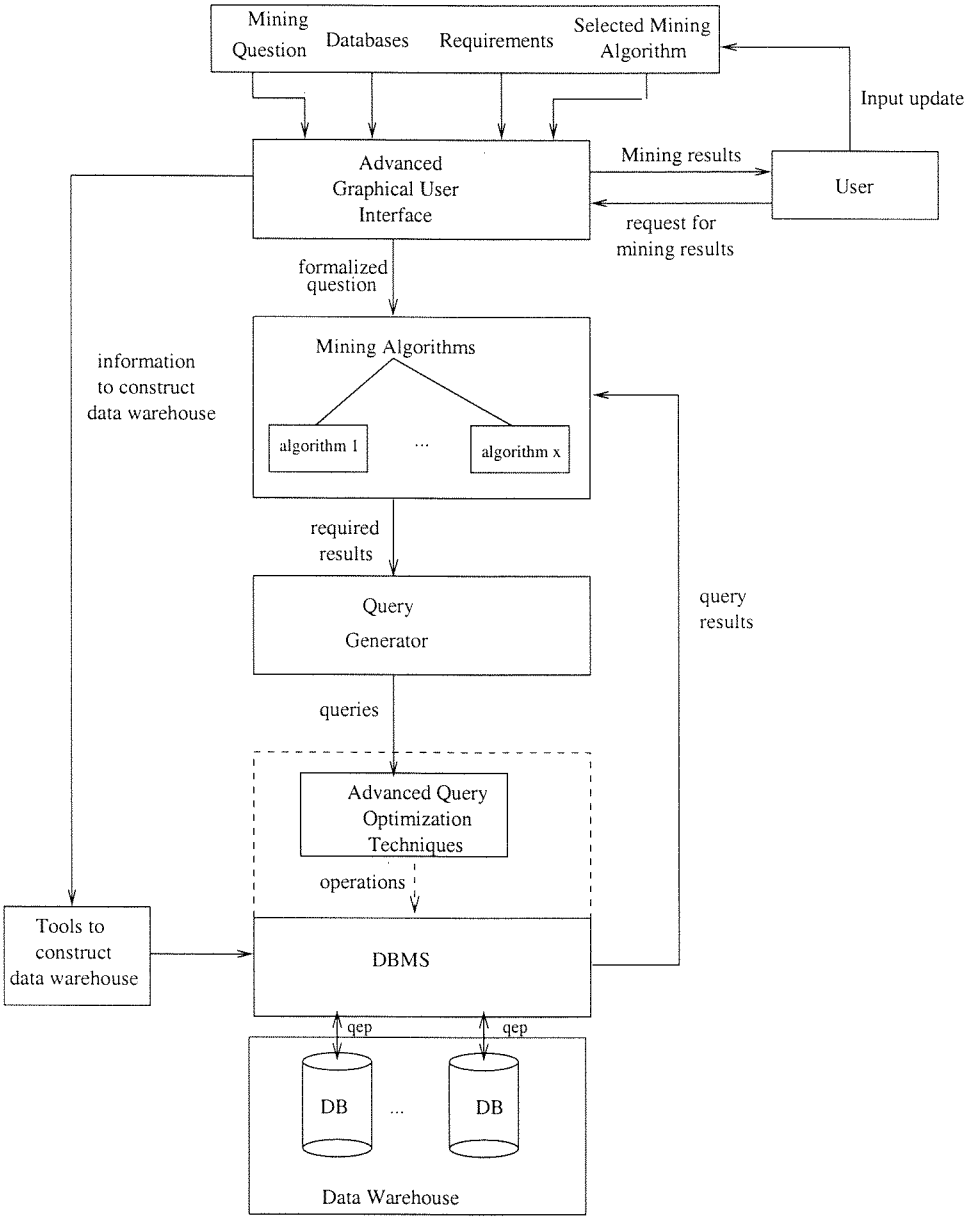

Fig. 3 Architecture of a data mining tool

chitecture consists of a number of modules. The input consists of a mining question, the databases that should be mined, the algorithm that is selected for mining, and additional requirements that may be posed by the user. For example, a user may demand that an attribute in a database should not be involved in the mining process for reasons of privacy (e.g., income of pilots). The user has

the possibility to request intermediate mining results, and if desired, the user is able to modify the input. An advanced graphical user interface is required to facilitate the presentation of the input and the interaction between tool and user. Mining results should be presented in a way that can be easily interpreted by domain experts.

Once the tool has received its input, it should extract and pass proper information to a module that contains a suite of tools to set up a data warehouse on the one hand, and on the other hand it should formalize the mining question such that it is understood by the selected mining algorithm. In general, the generation of a data warehouse may be a complex task. Therefore, tools are required to support this task. For example, there is a practical need for tools that detect conflicting data, filter noisy data, etc.

Once a data warehouse has been established, a mining algorithm will heavily interrogate the database to expose useful knowledge hidden in the database. Each interrogation should be translated into a query that is understood by the underlying database management system (dbms). Therefore, a query generator should be part of a data mining tool. If a dbms receives a query, it generates an efficient query execution plan (qep), which describes step by step the operations to be performed in order to retrieve the result of a query from the data warehouse. Then, the query execution plan is performed, and the result is passed to the dbms, which passes it, in turn, to the mining algorithms module.

Since the retrieval of data from databases is still a bottleneck [Ref 9, 14], the performance of a data mining tool depends on the query processing capabilities of a database management system (dbms). We have observed that mining algorithms may pose simultaneously sequences of interdependent queries to a dbms and that exploitation of dependencies speed up query processing [Ref 5, 10, 20]. Since database management systems on the market place do not take advantage of this fact, we suggest to implement advanced query optimization techniques on top of commercial database systems in order to gain performance.

Our motivation to equip the mining algorithm module with a wide variety of algorithms is that some algorithms may very well be suited for some applications, while less suited for others.

A wide variety of techniques are available for mining purposes (step 3) varying from classical techniques, such as regression, to more novel ones, such as evolutionary computing techniques. In the following, we discuss how to set up a mining algorithm module. We briefly discuss a number of mining techniques which might be included in such a module.

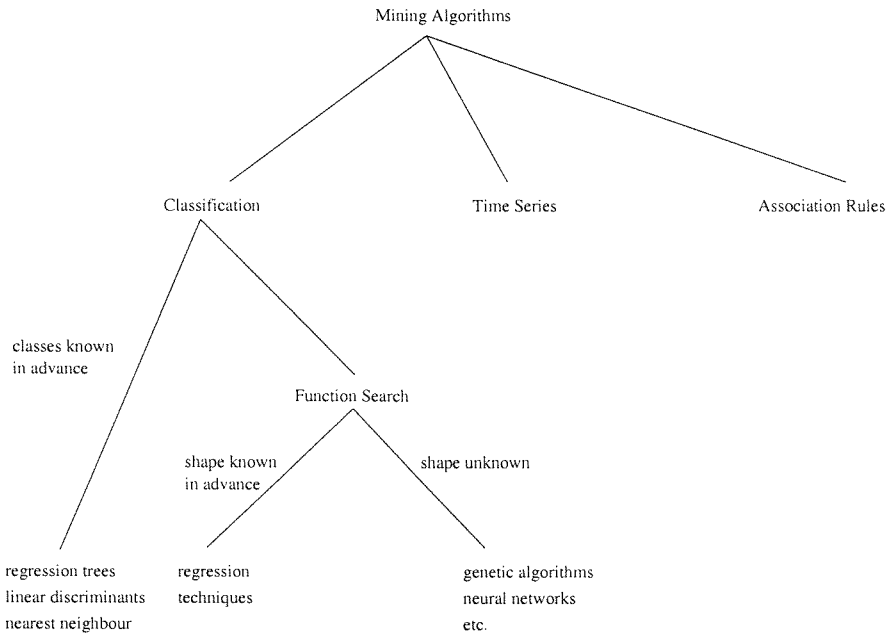

Fig. 4 Module of mining algorithms

4.2 Mining algorithm module

During the past years many algorithms have been developed and implemented for data mining tasks by the research as well as the commercial community. A proper question that raises up: "What mining algorithm do I need?". Although a rough categorization is possible for the mining algorithms, the question remains tough to answer. Currently, we may distinguish three categories of algorithms: classification algorithms, algorithms for association rules, and algorithms to mine time series databases.

Classification has as goal to distribute objects/tuples on the basis of common properties into a number of classes [Ref 1]. Algorithms for association rules are focussed towards the search of frequently occurring patterns in a database [Ref 2]. Mining algorithms for time series databases search for common patterns embedded in a database of sequences of events [Ref 4].

As motivated before, a mining algorithm module should be equipped with a wide variety of algorithms. In order to select the proper algorithm, a characterization of each type of algorithms is necessary. Then, a system or an expert may choose an algorithm for the problem at hand.

In Figure 4, we illustrate for classification problems how such a module may be set up. On the basis of a number of characteristics that may be relevant in choosing a mining technique, a tree is constructed. Suppose we have a mining problem and we know that attributes $x_1, x_2, x_3, ..., x_n$ are

linear dependent on an attribute y. If we are searching for a function to describe this dependency, then a linear regression technique might be a good choice for this task. For a more detailed characterization of a number of mining techniques, we refer to Section 4.2.2

Today, the majority of the mining algorithms are focussed towards association rules and classification. In the following we give a brief overview of the algorithms in these fields.

4.2.1 Association rules

The field of association rules has been inspired by databases that store items purchased by a customer as a transaction [Ref 2, 3, 15]. A planning department may be interested in finding associations between sets of items. An example of such an association may be that 90% of the transactions that purchase diapers also purchase beer. This might be a good reason to place these two items close to each other to provide the customer a better service.

For reasons of convenience, we model a supermarket database as a relation *basket*$(i_1, i_2, i_3, ..., i_k)$, in which $i_j, 1 \leq j \leq k$, is a binary attribute that records whether an item has been sold or not. Note that a tuple in the database corresponds to a shopping basket at the counter.

Let I be the sets of all items, $X, Y \subseteq I$, and $s(X)$ the percentage of baskets containing all items in X. Then, $X \rightarrow Y$ is a association rule with regard to t_1 and t_2 iff $s(XY) \geq t_1$ and $\frac{s(XY)}{s(X)} \geq t_2$, in which $Y \not\subseteq X$. The latter equation expresses the confidence in a rule, while the former equation guarantees that itemsets should have a minimum size, and therefore a minimum support.

The problem is to find all rules that have minimum support and confidence greater than the defined threshold values t_1 and t_2. In general, two steps are distinguished in solving this problem. In step 1, all itemsets that have minimum support are selected. And in step 2, for each itemset X found in step 1 and any $Z \subset X$, all rules that have minimum confidence are generated.

Suppose that we have the following four transactions: $T_1 = \{a, b, c\}$, $T_2 = \{a, b, d\}$, $T_3 = \{a, d, e\}$ and $T_4 = \{a, b, d\}$. Let the minimum support and minimum confidence be 0.7 and 0.9 respectively. Then, the following itemsets meet the minimum support. $s(\{a\}) = 1$, $s(\{b\}) = 0.75$, $s(\{d\}) = 0.75$, $s(\{ab\}) = 0.75$, and $s(\{ad\}) = 0.75$. The valid association rules are[1]: $b \rightarrow a$ and $d \rightarrow a$, both with confidence 1.0.

Having obtained the itemsets that meet the minimum support, step 2 is straightforward. The solution for step 1 is harder. A simple solution for step 1 is to form all itemsets and obtain their

[1] Note that $a \rightarrow b$ is not a valid rule, since $\frac{s(ab)}{s(a)} = \frac{0.75}{1} < 0.9$.

support in one pass over the data. However, this solution is computationally infeasible, since the number of itemsets grows exponentially with the number of items. The challenge is to minimize the complexity and the number of passes over the database. A large number of papers in the literature report on various type of solutions for this problem, exploiting mathematical properties as well as domain knowledge.

4.2.2 Classification

As noted before, classification has as goal to distribute the tuples of a database into a number of, pre-defined or not, classes. In general, if the classes are not pre-defined, the term clustering is used. In the remainder of this section, we restrict ourselves to the case that the classes are pre-defined.

Today, the most popular classification techniques are based on decision trees, while evolutionary techniques are gaining considerable attention. Classical techniques, such as regression, kernel density methods, etc. are still appropriate for many data mining tasks. Due to space limits, we briefly characterize decision trees and two evolutionary techniques namely, genetic algorithms and neural networks[2]. Our characterization is based on the following terms: the assumptions on which a technique is based, the quality of the solution produced by a technique, and the "complexity" of a technique. Such a characterization can be made for classical techniques as well.

Decision trees[Ref 21] This type of algorithms picks an attribute as root, and splits it on all possible values, resulting in a tree with depth 2. If all corresponding tuples to a leaf are in a same class C, label that leaf with C. As long as leafs are unlabeled: choose a new attribute and expand the tree by splitting it on all possible values again. This process terminates until all leafs have been labelled. Algorithms based on decision trees differ in the way a choice is made for an attribute that will be split.

Assumption: None.

Quality of solution: The results of decision trees are easily interpreted and are useful as long as the trees are not too large. In general, large trees have a higher misclassification rate than small ones. There are techniques to determine the right size of a tree [Ref 13].

Complexity: The most expensive operation is the splitting of an attribute on attribute values and the classification of all tuples according to these values. Furthermore, a tree grows exponentially with the number of attributes.

Genetic algorithms[Ref 19] Genetic algorithms start with an initial population. Traditionally,

[2]For an overview of classical techniques in the context of data mining, we refer to [Ref 13], which is available upon request.

an individual/object in the population is represented as a string of bits. The quality of each individual is computed, called the fitness of an individual. On the basis of these qualities, a selection of individuals is made. Some of the selected individuals undergo a minor modification, called mutation. For some pairs of selected individuals a random point is selected, and the substrings behind this random point are exchanged, called cross-over. The selected individuals, whether or not modified, form a new generation and the same procedure is repeated with the new generation until some defined criterion is met.

Assumption: A representation of a population and a quality measure should be defined.

Quality of solution: Genetic algorithms have been successfully applied in a wide variety of applications. In some cases, it is proven that it converges to a local optimum. In other cases, experiments show that convergence occurs.

Complexity: The most expensive operation is the computation of the fitness function. A genetic algorithm searches different (small) parts of a search space. The complexity is linear with the number of individuals in a population and the number of generations that should be investigated.

Neural networks [Ref 17]: A neural network is a function that maps input patterns to output patterns. It consists of nodes and connections between nodes. Nodes are organised in layers, one input layer, a number of hidden layers, and an output layer. A node in layer i is connected with all nodes in layer $i+1$. The connections are labelled with a weight. The input nodes receive binary values from their environment. The other nodes compute a function from their weighted input and propagate the result. The function looks as follows:

$$V_j = g(\sum_k w_{j,k} * V_k)$$

in which V_j denotes the value of node j, $w_{j,k}$ the weight of the connection between node j and node k, and g is a function. The output of the network strongly depends on the function g. Often g is defined as follows:

$$g = \begin{cases} 1 & \text{if } x - \theta_x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

in which θ_x is a threshold value.

To make the network learn the correct function, we let it adjust the weights using a set of input-output pairs. A simple idea for an adjustment scheme is: if a network gives a wrong answer, the weights are adjusted proportionally to their contribution to the wrong answers.

Assumption: At least one hidden layer is necessary to approximate continuous functions.

Quality of solution: Depends on an appropriate choice of the parameters, such as number of nodes, hidden layers, etc. In general, this is a tough task.

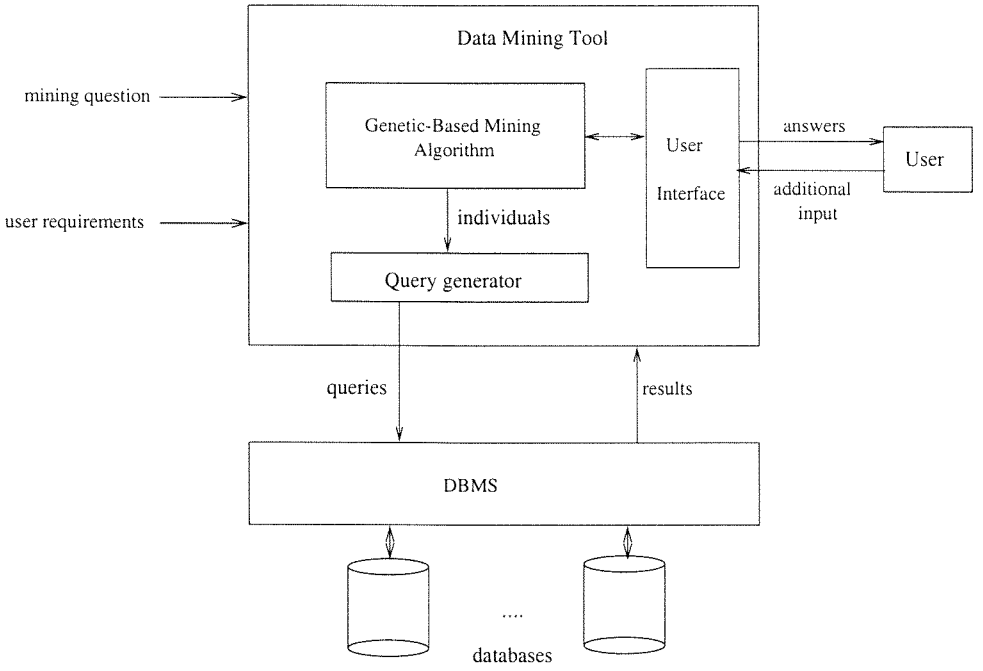

Fig. 5 Architecture of SHARVIND

Complexity: The computation of the values that should be propagated is the most expensive operation. Furthermore, once a network is established, the complexity is linear with the input. The number of connections grows also linear with the addition of a node in a layer.

4.3 An Application

At NLR, we have developed and implemented a re-targetable data mining tool for classification tasks that is running in a Microsoft environment [Ref 11]. Re-targetable means that the tool can be integrated with other database management systems, such as ORACLE, without much effort. Our mining tool is currently equipped with a mining algorithm module, a query generator module, and a simple user interface. The mining algorithm module consists of a genetic-based mining algorithm. The architecture of the tool, called SHARVIND, is depicted in Figure 5. The tool takes as input a mining question, which actually selects the part of the database where interesting knowledge should be searched, and possibly requirements (e.g., not to use certain attributes in the mining process) posed by a user. Then, a random number of expressions, called initial population, is selected. We note that an expression is a conjunction of predicates defined over a number of database attributes. The initial population is manipulated by applying the cross-over and mutation operators. Since we require the number of tuples that satisfy an individual/expression to compute the fitness of an individual, individuals are translated into corresponding SQL queries which are passed to the dbms. The fittest individuals are selected to form the next generation and the manip-

ulation process is repeated until no significant improvement of the population can be observed. As output, the tool delivers expressions whose corresponding number of tuples falls in a user-defined interval.

We have mined two real-life databases with the tool. Both databases contain aircraft incident data, one is set up by the Federal Aviation Administration (FAA) in the USA, referred as FAA database, and the other is set up by the Joint Research Centre (JRC) in Italy, referred as ECCAIRS database. We note that the FAA database is obtained from the Internet and is mined at our laboratory, while the JRC database is mined on location. In the FAA database aircraft incidents are recorded from 1978 to 1995, while ECCAIRS is recently taken in production.

The ECCAIRS database consists of 36 relations and about 300 attributes. Two major relations of the database are the ACS and the $OCCS$ relations. The ACS relation contains information with regard to aircraft, such as manufacturer, motor, speed of the aircraft, etc., and information about the environment in which the aircraft is involved, such as weather conditions. The $OCCS$ relation describes in general terms an occurrence (incident or accident) and contains general information with regard to an occurrence, for example, time and location of an occurrence, etc. The relation $ACCS$ contains 5202 tuples and 186 attributes and $OCCS$ contains 5202 tuples and 27 attributes. Although the number of tuples is not large, mining might be interesting due to the large number of attributes.

However, currently 17 relations do not contain any tuples, while other relations consist of tuples having many NULL values. So, in order to make the database suitable for mining, we have cleaned the database. We have removed attributes that have less than 2000 entries filled in, attributes whose values consist of natural language, attributes that are fully functional dependent on another attribute, and attributes with high and low selectivity factors.

After performing the removals, we have joined the relations ACS and OCCS and 64 attributes were left for mining.

At NLR, the FAA database is implemented as a single table that is sorted on an attribute, called report number, which served as primary key. In the following, we mean by the FAA database, the database as it is implemented and filled at our laboratory.

The FAA database consists of more than 70 attributes and about 60.000 tuples. As in the case of ECCAIRS, this database contains also NULL values, redundant data, and attributes with very high and low selectivity factors. Therefore, we have cleaned this database in the same way as

ECCAIRS, in order to make it suitable for mining. After cleaning 30 attributes were left and 60.000 tuples.

We have mined both databases by posing several questions concerning safety aspects to our tool. We start with the question "What are the profiles of risky flights?" We have posed to the FAA databases some additional mining questions (concerning safety aspects), such as "Given the fact an incident was due to operational defects not inflicted by the pilot, what is the profile of this type of incident", etc. We have presented the mining results to safety experts at our laboratory and the overall conclusion was that the answers to the mining questions were correct and promising. The mining results helped safety experts to gain insight in the databases and hopefully also knowledge in future. For example, an unexpected result from the FAA database was the following association: *aircraft_damage* **is** ('minor') $\wedge$ *primary_flight_type* **is** ('personal') $\wedge$ *type_of_operation* **is** ('general operating rules') $\wedge$ *flight_plan* **is** ('none') $\wedge$ *pilot_rating* **is** ('no rating') $\rightarrow$ pilot_induced.
This association means that pilots without flight certificates and flight plans who are flying in private aircraft are causing more incidents than other groups. This result was on the first glance a bit strange for our safety expert, since pilots without flight certificates are not allowed to fly. After a while it appeared that the association was correct and our safety expert was able to explain the association. The pilots without certificates appeared to be students whose incidents were recorded in the FAA database as well.

5 Conclusions

We have discussed a number of trends that may be distinguished in the military community. To implement these trends successfully, an adequate processing of various types of information with acceptable performance is required. In this paper, we have briefly introduced the field of multimedia databases and data mining. We have touched on the potentials of these fields for the next generation of military information systems as well as the challenges they entail.

Acknowledgement The authors thank Wim Pelt from the Royal Netherlands Navy, who made this research possible.

References

1. Agrawal, R., Ghosh, S., Imielinski, T., Iyer, B., Swami, A., An Interval Classiefier for Database Mining Applications, in Proc. of the 18th Very Large Data Base, 1992, pp. 560-573.
2. Agrawal, R., Imielinski, T., Swami, A., Mining Association Rules between Sets of Items in Large Databases, in Proc. ACM SIGMOD '93 Int. Conf. on Management of Data, 1993, pp. 207-216.
3. Agrawal, R., Srikant, R., Fast Algorithms for Mining Association Rules, in Proc. Int. Conf. on Very Large Databases, 1994, pp 487-499.
4. Agrawal, R., Srikant, R., Mining Sequential Patterns, in Proc. 11th Int. Conf. on Data Engineering, 1995, pp. 3-14.
5. Alsabbagh, J.R., Raghavan, V.V., *Analysis of Common Subexpression Exploitation Models in Multiple Query Processing*, in Proc. 10th Int. Conf. on Data Engineering, IEEE Press, pp. 488-497, 1994.
6. Boyes, J., Andriole, S., (Eds.) Principles of Command and Control, AFCEA International Press, Washington, D.C., USA, 1987.
7. Bruggeman, B., N de Reus, Tracking and Prediction: A View to the Future, Proc. AFDRs/WG15 Conf on Maritime Combat Systems Engineering, Portsdown, 1998.
8. Bruggeman, B., Command & Control- Advanced Reasoning Methods, Proc. C4I Symposium, Den Helder, the Netherlands, 1999.
9. Choenni, R., *On the Automation of Physical Database Design*, Ph.D. thesis, University of Twente, 1995.
10. Choenni, R., Kersten, M., Saad, A., van den Akker, J., A Framework for Multi-Query Optimization, in Proc. COMAD '97 8th Int. Conference on Management of Data, 1997, pp. 165-182.
11. Choenni, R., On the Suitability of Genetic-Based Algorithms for Data Mining, Advances in Database Technologies, ER '98 Workshops on Data Warehousing and Data Mining, Mobile Data Access, and Collaborative Work Support and Spatio-Temporal Data Management, Y. Kambayashi, D.L. Lee, E-P. Lim, M. Mohania, Y. Masunaga (Eds), LNCS 1552, Springer Verlag, Germany, 55-67, 1998.
12. Choenni, R., Leijnse, K., A Framework for the Automation of Air Defence Systems, To appear in: RTA SCI Panel Symposium on Warfare Automation: Procedures and Techniques for Unmanned Vehicles, NATO RTA, Neuilly-Sur-Seine Cedex, France, 1999.
13. Choenni, R., de Laat, R., Data Mining: A Brief Introduction, NLR memorandum ID-97-004, 1997, Amsterdam.

14. Elmasri, R., Navathe, S.B., *Fundamentals of Database systems*, The Benjamin/Cummings Publishing Company, California, USA, 1988.
15. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., (Eds), Advances in Knowledge Discovery and Data Mining, AAAI/The MIT Press, 1996.
16. G. Grimmett, D. Stirzaker, Probability and Random Processes, Oxford Science Publications, Oxford University Press, USA, 1989.
17. B. Kosko, Neural networks and Fuzzy Systems, Prentice Hall Inc., 1992.
18. M. Lalmas and I. Ruthven. Representing and Retrieving Structured Documents using the Dempster-Shafer Theory of Evidence: Modelling and Evaluation. Journal of Documentation, 54(5):529-565, December 1998.
19. Michalewicz, Z., Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York, USA.
20. Sellis, T.K., *Multiple-Query Optimization*, in ACM Trans. on Database systems 13(1), ACM Press, pp. 23-52, 1988.
21. R. Quinlann, C4.5: Programs for Machine Learning, Morgan Kaufmann, USA.
22. Thuraisingham, B. (Ed.), Handbook of Data Management 1998, Auerbach Publications, Fl, USA, 1998.
23. H.Turtle, W. Croft, Inference Networks for Document Retrieval, Computer and Information Science, Univ. of Massachusetts, USA.
24. J. Ullman, Principles of Database and Knowledge-Base Systems, Vol 2.: The New Technologies, Computer Science Press, USA, 1989.
25. A. de Vries, Content and Multi-media Database Management Systems, Ph.D. thesis, Univ. of Twente, Enschede, the Netherlands, 1999.
26. Klomp, J., Zetten, H. van, Army Organic Air Defense: Effective and Affordable after 2000?, Militaire Spectator 167(3), 1998 (in Dutch).