Attribute Induction Generalization Using Concept Hierarchy

By:
Mosa Mohammed Alrotub
Supervisor: Dr. Faraj A. El-Mouadib

A thesis submitted to the Department of Computer Science
In partial fulfillment of the requirements for the degree of
Master of Science

Al-Tahaddi University
Faculty of Science
Department of Computer science
Libya_Sirite, G. S. P. L. A. J.

Academic year 2008/2009

Faculty of Science

Department Of Computer Science

Title of Thesis

Attribute Induction Generalization Using Concept Hierarchy

By

Mosa Mohammed Alrotub

Approved by:

Dr. Faraj A. El-Mouadi
(Supervisor) .....

Dr. Idris S. El-Feghi
(External examiner) .....

Dr. Zakaria Suliman Zubi
(Internal examiner) 2008/11/27

Countersigned by:

Dr. Ahmed Farag Mhgoub

(Dean of faculty of science)

Dedication

To My Parents, Family and Friends

and my country- Libya

Acknowledgment

I would like to express my thanks to Al-tahadi University for giving me the chance to study master in my country, and thanks to my supervisor, ***Dr. Faraj A. El-Mouadib,*** for his help and assistance throughout my work. His steadiness and patient support were essential to completion of this work.

Finally, I would like to thank my family for their continued support and understanding throughout the last few years.

Mosa M. Alrotub

Abstract

In the last thirty years or so, the worlds have seen a rapid growth in the collection of massive data in many different fields. These massive volumes of data have called for a new tools and techniques for data analysis. These new tools and techniques makeup a newly emerging field called Knowledge Discovery in Databases (KDD) or Data Mining (DM). The availability of data from different sorts and research interests has produced great demand for development of new tools and techniques for an automated and intelligent database analysis.
Recently, data mining has been ranked as one of the most promising research topics for the (1990s) by both database and machine learning researchers.

There are several methods for effective data. For instance, the data cube-*based OLAP roll-up operation* can be used to perform user controlled data summarization along a specified dimension, and *an attribute-oriented induction* (AOI) technique can be used to perform data generalization and characterization.
AOI is summarization algorithm that has been effective for KDD, it summarize the information in a relational databases by repeatedly replacing specific attribute values with more general concepts according to user defined Concept Hierarchy (CH) that is called data generalization.

The Data generalization is a process which abstracts a large set of task-relevant data in a database from low conceptual levels to higher ones.
A key in basic AOI is the attribute-oriented concept tree ascension for concept generalization; a concept hierarchy is used to represent background knowledge supplied by domain experts.

A concept hierarchy associated with an attribute in a database is represented as a tree where leaf nodes correspond to actual data values in the database, intermediate nodes correspond to more general representation of the data values. the AOI has absorbed many advanced features of recently developed learning algorithms.
The purpose of this research is to implement a computer system for extracting information to handling complex data types of the attributes and their aggregations by using data generalization with concept hierarchy, this approach will be implemented using visual basic6 language that embeds the SQL and many other advantages, The

system will be tested with Microsoft access database and represent the results by Visualization techniques such as (i.e. rules, Cross tabs, charts, cubes).

June, 2008

Table of contents

Dedication	ii
Acknowledgment	iii
Abstract	iv
Table of contents	vi
List of figures	viii
List of tables	X
Chapter One: Introduction	1
1.1 General information	1
1.2 Background of Knowledge discovery and data mining	3
1.2.1 Knowledge discovery process	3
1.2.2 KDD necessities	5
1.2.3 Types of discovered knowledge	5
1.2.4 Kinds of data can be mined	6
1.2.5 Methods and Techniques of data mining	6
1.3 An Overview of Attribute Oriented Induction (AOI)	7
1.4 Motivation of the work	8
1.5 Thesis Organization	8
Chapter Two: Concept hierarchies and characterization rules	10
2.1 Concept hierarchy	10
2.1.1 Concept hierarchies in data mining	10
2.1.2 Concept hierarchy and generalization operation	11
2.1.3 Types of Concept Hierarchies	13
2.1.3.1 Schema hierarchy	13
2.1.3.2 Set-grouping hierarchy	14
2.1.3.3 Operation-derived hierarchy	15
2.1.3.4 Rule-based hierarchy	15
2.1.4 The role of concept hierarchy in AOI	16
2.2 Characterization rules	17
Chapter Three: System Design and Implementation	19
3.1 The DM-AIG tasks	19

3.2 Principles of attribute-oriented induction 20
3.2.1 Basic strategies of AOI 20
3.2.2 The idea of AOI 21
3.2.3 Basic algorithm 22
3.3 System architecture 23
3.4 System modeling 24
3.4.1 Model building phase 24
3.4.2 Model testing phase 27
3.4.3 Model applying phase 27
3.5 Physical Design 27
3.5.1 Design Considerations 27
3.5.2 Design methodologies 28
Chapter Four: System testing and results 31
4.1 Synthetic data experiments 32
4.1.1 Small synthetic data experiment 32
4.1.2 Large synthetic data experiment 40
4.2 Real data experiment 45
Chapter Five: Conclusion and future work 49
5.1 System advantage 49
5.2 Conclusion 50
5.3 future work 51
References 52
Appendix 55

List of figures

Figure		Page
1.1	The Knowledge Discovery Process	4
1.2	General architecture of DBMiner system	8
2.1	Concept hierarchy operations: data generalization and specification	10
2.2	A concept hierarchy for the attribute *major*	13
2.3	Set-grouping concept hierarchy for the attribute status	14
2.4(a)	Generalization rules for the concept hierarchy GPA	15
2.4(b)	Rule-based concept hierarchy for attribute GPA	16
3.1	The architecture of DM-AIG system	24
3.2	The build process of the system	26
3.3(a)	Process notations	28
3.3(b)	Datastore notations	28
3.3(c)	Dataflow notations	29
3.3(x)	External entity notations	30
3.4	DFD for the DM-AIG system	29
4.1	query when "Status= graduate in small synthetic university dataset	34
4.2	Final generalized relation for threshold=2 in small synthetic university dataset	34
4.3	Final generalized relation for threshold=6 in small synthetic university dataset	35
4.4	Final generalized relation for threshold=12 in small synthetic university dataset	35
4.5	Final generalized relation for threshold=30 in small synthetic university dataset	36
4.6	rules from small synthetic university data experiments when threshold=2	36
4.7	query when "buying = expensive in small synthetic car dataset	39
4.8	Final generalized relation for threshold=3 in small synthetic car dataset	39
4.9	Final generalized relation for threshold=4 in small synthetic car dataset	40
4.10	Final generalized relation for threshold=2 in large synthetic university dataset	41
4.11	Final generalized relation for threshold=6 in large synthetic university dataset	42
4.12	Final generalized relation for threshold=12 in large synthetic university dataset	42

4.13	Final generalized relation for threshold=30 in large synthetic university dataset	43
4.14	Final generalized relation for threshold=3 in large synthetic car dataset	44
4.15	Final generalized relation for threshold=4 in large synthetic car dataset	45
4.16	Query snap shot of "buying = expensive in real car dataset	46
4.17	Final generalized relation for threshold=2 in real car dataset	47
4.18	generalized relation for threshold=4 in real car dataset	47
4.19	Rules from real data experiment when threshold=2	48

List of tables

Table		Page
2.1	Concept hierarchies for a university student data	12
4.1	*Parameters of synthetic data*	31
4.2	Description of the University student database	32
4.3	Concept hierarchies table of the university database	33
4.4	Small synthetic data experiments results	33
4.5	Description car database	37
4.6	Concept hierarchies table of the car database	38
4.7	Small synthetic data experiments results	38
4.8	Small synthetic data experiments results	41
4.9	Small synthetic data experiments results	44
4.10	Small real data experiments results	46
5.1	Summary of the experiments	50

Chapter 1

Introduction

1.1. General Information

In theses days that we are living in where huge amount of information is all around us is referred to as information age. Due to the fact that information leads to power and success and the main source of information is data, and thanks to sophisticated technologies that are available to us such as: computers, satellites, etc., that made it possible to collect tremendous amounts of data. With the advance techniques for mass digital storage and communication technologies, it is possible to collect and store all sorts of data by relying on the power of computers to establish that.

Unfortunately, these huge volumes of data stored on different storage media have become overwhelming for human to understand and digest. This has led to the creation of structured databases and Data Base Management Systems (DBMS). The efficient DBMSs have been very important assets for the management of large amount of data, especially for effective and efficient retrieval of particular information whenever it is needed. The creation of DBMSs has been one of contributors to recent massive gathering of all kinds of information. "Recently, our capabilities of both generating and collection data have been increasing rapidly. The widespread use of bar codes for most commercial products the computerization of many business and government transactions, and the advances in data collection tools have provided us with huge amounts of data"[12]. Because of the availability of powerful and affordable database management systems, millions of databases have been collected and used for all different purposes and applications. "The availability of data from different sorts and research interests has produced great demand for development of new tools and techniques for an automated and intelligent database analysis."[2].

the explosive growth in databases has called for new tools and techniques to transform these data into useful knowledge. The collections of these tools and techniques have been integrated into a new filed known as Knowledge Discovery in Databases (KDD). As it has been defined in [11], *Knowledge discovery* is the *nontrivial* extraction of *implicit*, previously unknown, and *potentially* useful information from data.

In the above definition of knowledge discovery, the nontrivial extraction concept is in the sense that it involves the search for structure or models that governs the data. Being implicit means the discovered knowledge is within the data but it is not explicit (i.e. it is not known before hand). The discovered knowledge has to be useful or beneficial to the user.

The field of KDD was introduced as early as 1989 to refer to the process of finding wide range of knowledge in data and it is one of the most interesting subjects in the field of computer science research, and recently has been an active research area in many business and scientific domains.

According to [10], formally knowledge is defined as: given a set of facts (data) F, a language L, and some measure of certainty C, we define a *pattern* as a statement S in L that describes relationships among a subset F_S of F with a certainty C, such that S is simpler (in some sense) than the numeration of all facts in F_S. A pattern that is interesting (according to a user-imposed interest measure) and certain enough (again according to the user's criteria) is called *knowledge*.

In the literature KDD is also known as Data Mining (DM) while some authors consider DM as a step in the KDD process. Simply stated, DM is a process by which data can be transformed to knowledge. Data mining as defined in [6], is the extraction of "information" or "knowledge" from data, which helps understanding data in databases and automatic construction of knowledge bases from databases. Data mining is also known by many other terms, such as *knowledge mining from databases*, *knowledge extraction*, *data archaeology*, *data analysis*, and so on. "The term DM has

been commonly used by statisticians, data analysts and the MIS (Management Information Systems) community, while KDD has been mostly used by artificial intelligence and machine learning researchers."[3]. basically, data mining deals with the discovery of hidden knowledge in the form of regularities (patterns) from large databases via some analytical software. The field of data mining has attracted researchers from many other fields such as: knowledge base systems, artificial intelligence, machine learning, knowledge acquisition, statistics, and data visualization. Furthermore, various data mining techniques have been used by service providers such as on-line commerce and World Wide Web to better understand user behavior and to increase business opportunities. "Recently, data mining has been ranked as one of the most promising research topics for the 1990s by both database and machine learning researchers"[1].

1.2. Background of Knowledge Discovery and Data Mining

Due to the rapid increase in the size and number of databases has created an urgent need and opportunity for extracting knowledge from these databases. The extraction of knowledge has necessitated the use of some tools and techniques in different fields such as; expert systems, machine learning, intelligent databases, knowledge acquisition, and statistics in order to transform data into useful knowledge.

1.2.1. The Knowledge Discovery Process

Discovering knowledge from data can be seen as a process of several steps which are necessary to ensure the efficient and effective discovery. As it has been mentioned in [17] and depicted in figure1.1, the discovery process consists of the following steps:

1. Identify the problem is the first step *to understand the application domain and to formulate the problem.* This step is clearly a condition for extracting useful knowledge and for choosing appropriate data mining methods according to the application objective and the nature of data.

2. **Preparing the data** is *to collect and preprocess the data* via a number of tasks such as: data collection and integration from different sources, removal of noise or outliers, filling in missing data, and transformation and reduction of the data to an appropriate form. This step is usually time consuming and it could be unnecessary.
3. **Building the model** is the main step in which hidden patterns or regularities are extracted from the data. A model can be viewed as a global representation of the structure that summarizes the systematic component of the data or that describes the data. Some of the models are: classification, clustering and association.
4. **The model evaluation** is to estimate how well a particular pattern (a model and its parameters) meets the criteria of the KDD process. Evaluation of predictive accuracy (validity) is based on cross validation. Evaluation of descriptive quality involves predictive accuracy, novelty, utility, and understandability of the fitted model. Both logical and statistical criteria can be used for model evaluation.
5. **Monitoring the model** is to use and test the model that was built in the previous step to validate its accuracy.

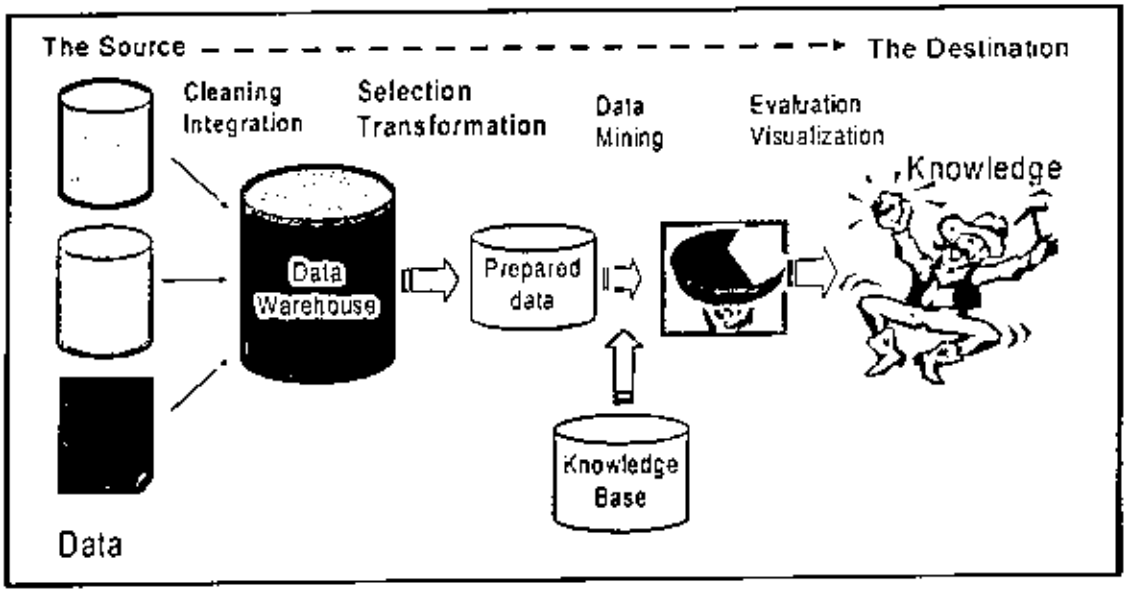

Figure 1.1: The Knowledge Discovery Process.

1.2.2. KDD necessities

There are many reasons that explain the need to use KDD, some of them are:

- Many organizations gather *so much data*, but they do not know what to do with them?
- People store data because they think *some valuable assets are implicitly coded within it*. In scientific endeavors might find some valuable knowledge from such data.
- In business, data captures information about critical markets, competitors, and customers. In manufacturing, data captures performance and optimization opportunities, as well as the keys to improving processes and troubleshooting problems.
- Data that may never be analyzed continues to be collected, at great expense, out of fear that something which may prove important in the future is missed
- Data volumes are large for classical data analysis tools to digest.
- The availability of networking increases the opportunity for access.
- Web navigation on-line product catalogs, travel and services information are very valuable knowledge to business and industry.
- Need to quickly identify and respond to emerging opportunities before the competition.
- As databases grow, ability to support analysis and decision making using traditional (SQL) queries infeasible:
- Many queries of interest are difficult to state in standard query languages.

1.2.3. Types of discovered knowledge

Several typical kinds of knowledge can be discovered usually as rules by data miners, including association rules, characteristics rules, classification rules, discrimination rules, and clustering, evolution, and deviation analysis.

1.2.4. Kinds of data can be mined

Data Mining extracts hidden knowledge, unpredicted patterns and new rules from large databases. Basically, it is concerned with the analysis of data and the use of software for finding patterns and regularities in sets of data. Hence, data mining is actually part of the knowledge discovery process. It can be seen from figure-1, that the data mining is one step in the iterative knowledge discovery process.

In principle, data mining is not specific to one type of media or data. Data mining should be applicable to any kind of information repository. However, algorithms and approaches may differ when applied to different types of data. Indeed, the challenges presented by different types of data vary significantly. Data mining is being put into use and studied for databases, including relational databases, object-relational databases and object oriented databases, data warehouses, transactional databases, unstructured and semi structured repositories such as the World Wide Web, advanced databases such as spatial databases, multimedia databases, time-series databases and textual databases, and even flat files. "The goal of data mining is to produce new knowledge that the user can act upon. It does that by building a model of the real world based on data collected from a variety of sources, including corporate transactions, customer histories, customer demographic information, and relevant external databases such as credit bureau information. "[17].

1.2.5. Methods and Techniques of data mining

Consider we have an infinite mount of data, yet little information. Some people can not accept this fact that mean, *more data and less knowledge*. Therefore, there is an urgent need to use data mining techniques to find only the important knowledge from the accumulated huge amount of data. The techniques to extract the knowledge from the database include:

- Decision trees.
- Association rules.

- Clustering.
- Attribute oriented induction.
- Taxonomy formation.

1.3. An Overview of Attribute Oriented Induction (AOI)

One of the early systems for knowledge discovery is DBLearn then named DBMiner was developed at Simon Fraser University in the period between 1989 and 1993. The system integrates data mining techniques and database technologies for the discovery of different types of knowledge such as: characteristic rules, discriminant rules, association rules and others from large relational databases efficiently and effectively. The system has the capabilities to different types of database such as: extended-relational, deductive, object-oriented, active databases, spatial, engineering and multimedia databases.

In the process of knowledge discovery, the system applies the concept of attribute-oriented induction. The process of attribute-oriented induction uses concept hierarchies in order to generalize the basic concepts (basic attribute values) to a more general values (general attribute values) until some certain condition is satisfied. This methodology reduces the number of combinations (attribute values) in the produced rule. Data generalization is the core function of DBMiner system for efficient and flexible generalization of large data sets. The generalization can be categorized into two (data structures) approaches.

- *Multi dimensional data cube approach, and*
- *An attribute oriented induction approach (generalized relation).*

The general architecture of DBMiner is show in Figure-2, which tightly integrates a relation database system such as SQL server, Access with knowledge base (Concept Hierarchy)

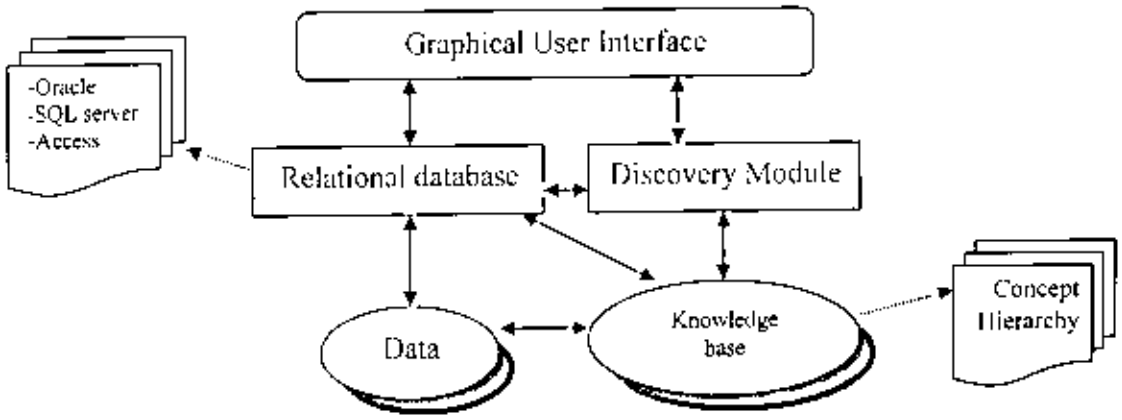

Figure 1.2: General architecture of DBMiner system.

1.4. Motivation of the work

This thesis is setup to explore the concept of the AOI in details and implement a prototype system that simulates the attribute oriented induction in the process of discovering knowledge. In other words the proposed system will be able to produce knowledge at different levels of granularities. The proposed system will be implemented in high-level language called Visual Basic6 with embedded the SQL commands. The proposed system will accept three types of input, which are as follows:

1. *Database*
2. *Learning task*
3. *Number of concept hierarchies*

The system will produce its final results in different types of visualization techniques such as: Cross tabs, charts, cubes, or rules.

1.5 Thesis Organization

The thesis contains five chapters organized as follows:

An overview of the concept hierarchy and characterization rules is discussed in chapter 2. We describe in chapter 3 the algorithm for AOI and design the system that implements it by visual basic6 language. In chapter 4, the system will be tested with Microsoft access database and represent the results by any form of Visualization techniques. Some concluding remarks are

represented in chapter 5 with a summary of the major thesis findings and with suggestions about the directions for future progress.

Chapter2

Concept hierarchies and Characterization rules

2.1 Concept Hierarchy (CH)

2.1.1 Concept hierarchies in data mining

A Concept Hierarchy (CH) defines a sequence of mappings from a set of low-level concepts (attribute values) to higher-level (more general) concepts. "A concept hierarchy is a collection of generalization relation for an attribute in a database; we define a generalization relation to be a relation between an exhaustive set of attribute values and a single higher-level, more general value."[14]. A concept hierarchy associated with an attribute in a database can be represented as a tree where leaf nodes correspond to actual data values in the database; intermediate nodes correspond to more general representation of the data values. Knowledge can also be shown in a hierarchy that shows the high and low levels of knowledge.

In general, CH deals with two main operations namely: generalization and specification as depicted in figure 2.1.

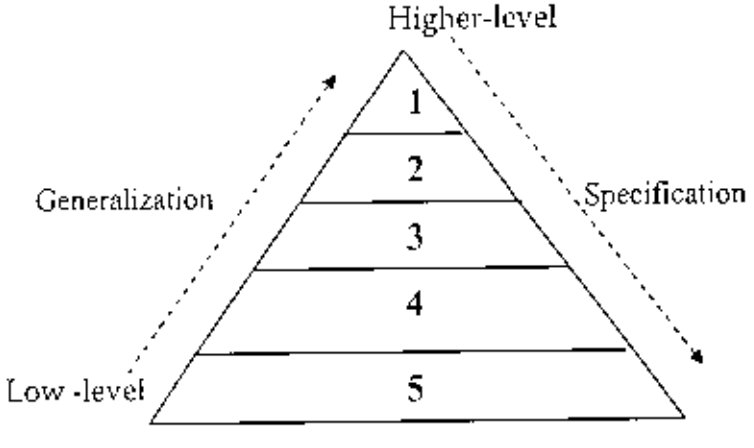

Figure 2.1: Concept hierarchy operations: data generalization and specification.

- In the level 2. Meta knowledge is the knowledge about knowledge.
- In the level 3. Specialized information that is specific to a subject becomes knowledge.
- In the level 4. Data that was processed and put into context becomes information.
- In the level 5. Data is a collection of raw points or facts.

The very basic element of knowledge is shown in a form of noise. It can be described as unclear data with no context. The concept hierarchies can be used to specify the level (granularity) at which the data mining to be performed.

2.1.2 Concept hierarchy and generalization operation

Data generalization is a process, which abstracts a large set of task-relevant data in a database from low-level concepts to higher-level ones. "A generalization relation can be represented by $\{A_1 \ldots A_k\} \subset B$, which indicates that B is a generalization of each A_i, for $1 \leq i \leq k$. All concept hierarchies for a database are obtained from domain experts and stored together in a concept hierarchy table (also called a concept hierarchy file)."[14]. for example table2.1[1], describe concept hierarchies for a university student data for the three attributes: *Major*, *Birthplace* and *GPA* (Grade Point Average). We can imagine a concept hierarchy for an attribute as a tree whose arcs (links) have all been reversed. In a natural tree, all arcs (branches) go from the root node towards the leaf nodes, but in a concept hierarchy tree, all arcs go from the leaf nodes toward the root node. Concept hierarchies represent the *knowledge base* that is necessary to control the generalization process in the attribute oriented induction approach.

Different levels of concepts are organized into a taxonomy of concepts, which is partially ordered from general to specific order. The most general concept is the null description, which is described by the reserved word "ANY", and the most specific concepts correspond to an attribute values in a database. Concept hierarchies can be manually provided by knowledge engineers and domain

[1] This example is taken from 1

experts, or generated automatically by statistical analysis of the data distributions. Moreover, many conceptual hierarchies are actually stored in the database implicitly. For example, the information that "*Vancouver is a city of British Columbia, which, in turn, is a province of Canada*", is usually stored in the database if there are "*city*", "*province*" and "*country*" attributes. Such relationships can be made explicit at the schema level by indicating that "*city* ⊂ *province* ⊂ *country*" then, the hierarchy of all the cities stored in the database can be retrieved and used in the mining process.

Attribute	Concept	Values
Major	Sciences	Biology, Chemistry, Physics, Computing, ...
	art	Literature, Music, Painting, ...
	Any	Sciences, art
Birth Place	British Columbia	Vancouver, Victoria, Richmond, ...
	Alberta	Edmonton, Calgary, Red Deer, ...
	Canada	British Columbia, Alberta,...
	Any	Canada, Foreign
GPA	Excellent	3.5,..., 4.0
	Good	3.0..., 3.49
	Average	2.0,... , 2.99
	Poor	0.0,..., 1.99
	Any	Excellent, Good, Average, Poor

Table 2.1: Concept hierarchies for a university student data.

Attributes with numerical values can be organized into hierarchies with discrete concepts automatically based on database statistics. Such automatic construction can be performed by first obtaining the distribution of attribute values in the database, then setting the range of the values and performing refined classifications in tightly clustered sub ranges." [1]. For example, the attribute GPA may disclose that GPA values falls between 0.00 to 4.00, and most GPAs' for graduate students fall between 3.00 and 4.00. For such attribute, one may classify the range of 0.00 to 1.99 into one class, and 2.00 to 2.99 into another and so forth. In General, hierarchies of numerical attributes can be modified based on database statistics and can be automatically

discovered or refined based on its relationship with other attributes. Different concept hierarchies can be constructed for the same attribute based on different preferences. Another example is the birthplace, which can be organized according to administrative regions such as provinces, countries, etc., geographic regions such as east coast, west coast, etc., or the sizes of the city, such as, metropolis, small-city, town, countryside, etc. Figure2.2 depicts the concept hierarchy of the attribute major within a university.

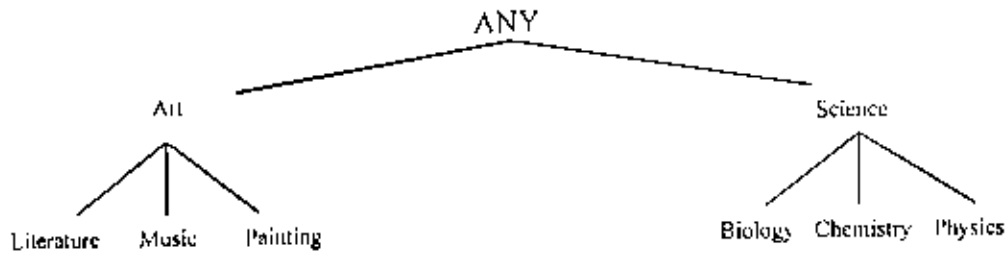

Figure 2.2: A concept hierarchy for the attribute *major*.

2.1.3 Types of Concept Hierarchies

There are four different types of concept hierarchies, namely schema hierarchy, set-grouping hierarchy, operation-derived hierarchy and rule-based hierarchy. The next subsection reviews these different types.

2.1.3.1 Schema hierarchy

This type of hierarchy is formed at the schema level by defining the partial order between the attributes in the database in order to materialize the relationships. For example, an address can be defined to form a partial order at schema level form the attributes house_number, street, city, province, and country,

house_number< street < city < province < country.

The way to define a hierarchy involving more than one relation is to define a view using the relations and WHERE clause, on which the hierarchy is then specified. "Although a hierarchy

defines at the schema level determines its partial order and the generalization and specification directions, for the purpose of executing a data mining task, we need to instantiate this schema hierarchy over the related data in a database to get a concrete or instance hierarchy."[13].

2.1.3.2 Set-grouping hierarchy

This kind of hierarchy is formed by defining set grouping relationships for a set of concepts (or values of attributes) in order to reflect semantic relationships characteristic to the given application domain. A set-grouping hierarchy is also called an instance hierarchy because the partial order of hierarchy is defined on the set of instance or values if an attribute. For example[2], the concepts freshman, sophomore, junior, senior for undergraduate students, and the concepts of M.Sc. and PH.D. for graduate students. These are the values for the status attribute in a university database environment. The concept hierarchy for the status attribute can be represented as follows:

{Freshman, sophomore, junior, senior} ⊂ Undergraduate

{M.Sc., Ph.D.} ⊂ Graduate

{Undergraduate, Graduate} ⊂ status

Such set-grouping hierarchy can be represented in a tree-like diagram as in figure 2.3.

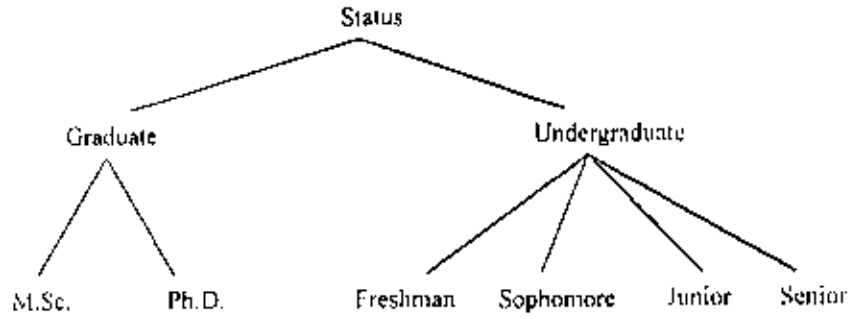

Figure 2.3: Set-grouping concept hierarchy for the attribute status.

[2] This example is taken from [13].

2.1.3.3 Operation-derived hierarchy

According to [13], this type of concept hierarchies is usually defined for numerical attributes and as it has been mentioned before, numerical attributes can be organized as discrete hierarchical concepts, and the hierarchies can be constructed automatically based on database statistics. This type of hierarchy can be created by simple or complex operations on the data. An example of simple operations is the discretization of numerical values into number of bins with equal or varying ranges. An example of complex operations is clustering and distribution analysis. This can be accomplished by first obtaining the distribution of an attribute values, setting the ranges of the bins and then performing refined classifications in tightly clustered sub ranges.

2.1.3.4 Rule–based hierarchy

A rule–based concept hierarchy is defined by a set of rules whose evaluation involves the data in a database. A lattice like structure is used to graphically visualize this type of hierarchy, in which every child-parent path is associated with one of the generalization rules. For example figure2.4(a) represents a set of generalization rules and figure2.4(b) depicts a rule-based concept hierarchy for the attribute GPA.

R1: {0.0 ~ 2.0} → poor;
R2: {2.0 ~ 2.5} ^ {graduate} → poor;
R3: {2.0 ~ 2.5} ^ {undergraduate} → average;
R4: {2.5 ~ 3.0} → average;
R5: {3.0 ~ 3.5} → good;
R6: {3.5 ~ 3.8} ^ {graduate} → good;
R7: {3.5~ 3.8} ^ {undergraduate} → excellent;
R8: {3.8 ~ 4.0} → excellent;
R9: {poor} → weak;
R10: {average} ^ {senior, graduate} → weak;

R11: {average} ^ {freshman, sophomore, junior} → strong;

R12: {good} → strong;

R13: {excellent} → strong.

Figure 2.4(a): Generalization rules for the concept hierarchy GPA.

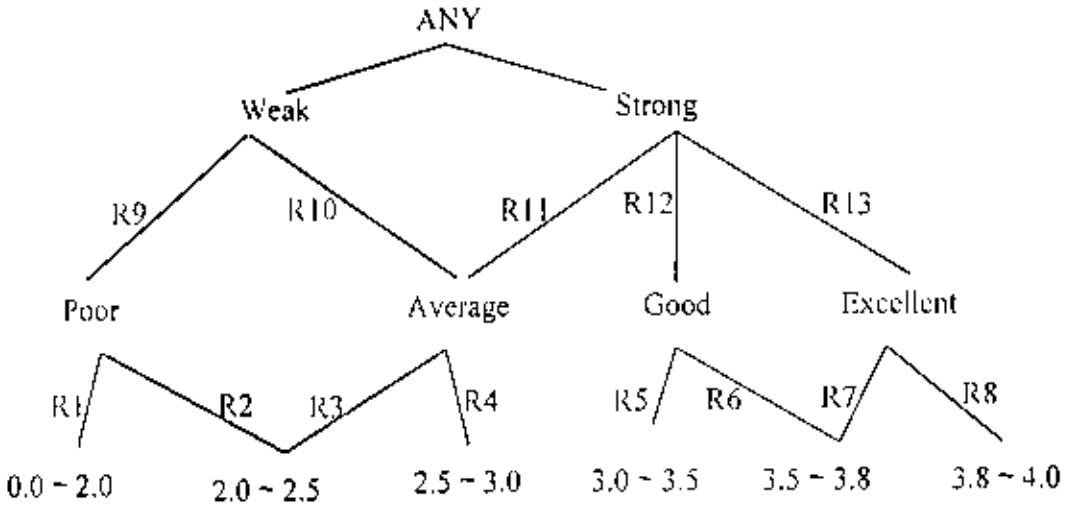

Figure 2.4(b): Rule-based concept hierarchy for attribute GPA.

2.1.4 The role of concept hierarchy in AOI

Mining for knowledge can be performed at different conceptual levels. The data in a database is known to be at the primitive level and the knowledge discovered from such raw data is known as primitive knowledge. "Knowledge discovery at the primitive level has been studies extensively. Most of the statistic tools and packages for data analysis are based on the raw data in a data set." [13] Knowledge discovered at higher level of abstraction has superior advantages over knowledge discovered at primitive level of being simpler, more general and covers large amount of data. For example, if the discovered rule at a primitive level is: *80% of people who are titled as professor, senior engineer, doctor and lawyer have salary between $60,000 and $100,000* and after abstracting the data to some higher level, we may have discovered the rule that states: *general speaking, well educated people get are well paid.*

2.2 Characterization rules

In general, the task of data mining can be categorize or classified into descriptive and predictive data mining. The former describes the set of data in concise and summarized manner and presents the general properties of the data whereas the latter constructs a model or a set of models from the data and attempts to predict the behavior of new data sets. the simplest kind of descriptive data mining is the concept description (or class description when the concept to be described refers to a class of objects. A concept usually refers to a collection of data, such as *winners*, *frequent_buyers*, *best_sellers*, and so on. As a data mining task, concept description is not simple enumeration of the data.

Concept description can be accomplished by data characterization and/or data discrimination. Data characterization is a summarization of the general characteristics or features of the class under study (the target class), while data discrimination is the comparison of the target class with one or a set of comparative classes (the contrasting classes). According to [19], characteristic rules express characteristics or properties of one class of instances in a typical attribute-value, or prepositional rule format.

For example, we all know that a swan is a white colored bird. This class can be expressed by the characteristic rule as *if* species = swan *then* type = bird *and* color = white. Such rule satisfies the completeness property (it is true for all or almost all swans) and so that the rule is not necessarily a good to differentiate between classes of instances in the database (because parrots and ducks are birds and can be white colored).

The output of data characterization can be presented in various forms such as tables, charts, cubes, logical rules, etc. A characteristic rule of a class is a conjunction of properties that are shared by all the entities in that class whereas a discriminant rule is a conjunction of properties

that shared by all the entities in the target class and distinguishes its entities from that of contrasting classes.

Chapter 3

System Design and Implementation

Over the whole world, massive amounts of data are already and will continue to be gathered in different types of databases via various kinds of data collecting tools. These scattered databases have created a need and an opportunity for knowledge extraction. KDD or DM, are the tools and techniques for the discovery of knowledge from these databases.

This work is concerned with the development of computer software system for data mining by the use of attribute induction and concept hierarchy concepts. The system name DM-AIG, which is the abbreviation of Data Mining Attribute Induction Generalization. The system is based on concept hierarchy tree ascension for the generalization process. The system uses data mining techniques which permit the discovery of useful knowledge from large amounts of database. The system is design to be used as; data access and summarization tools.

As a data access tool, it allows data analyst to dynamically organize his or her data according to many different high-level concepts without modifying the data itself and the analyst can make query according to high-level concepts rather than according to specific data item values, even though these high-level concepts are not present in the database. As a summarization tool, the system can generalize and summarize the data in large databases to many different levels, so that useful patterns in the data become apparent. Since the database itself is not used for these operations, they can be done quickly and efficiently.

The system can run on a PC with Microsoft Windows XP or higher platform operating system, it requires a processor with speed of 2.00 GHz at least, the required RAM is 512MB or more and hard disk of at least 80 GB in size.

3.1 The DM-AIG tasks

The tasks of the DM-AIG system software are organized into three different types and these are:

- *Data Retrieval task* is simply to retrieve only the data that is relevant to the current discovery task. This data will be displayed to the user.
- *Knowledge base task* is to relate basic data values (concepts) to their higher level concepts with an indication of the generalization levels. This task is represented in a table with three

columns: the first column contains the generalized concept, the second column contains the basic or low level concept and the third column indicated the generalization level.

- *Characteristic discovery task* is the task of finding some interesting relationship between various columns of one or more relational tables of the used database. Performing Characteristic discovery task is the main process of the DM-AIG software.

3.2 Principles of attribute-oriented induction

The concept of Attribute-Oriented Induction (AOI) is a useful data mining method that generalizes a database relation under appropriate set of concept hierarchies to extract meaningful knowledge.

3.2.1 Basic strategies of AOI

According to [1], there are seven basic strategies defined to perform attribute-oriented induction in relational databases:

- Strategy1: ***Generalization on the smallest decomposable component.***
  Generalization should be performed on the smallest attributes to ensure that the smallest possible chance is considered in the generalization process.
- Strategy2: ***Attribute removal.***
  For any attribute that has large number of distinct values and there is no concept hierarchy provided for that attribute, then the attribute should be removed in the generalization process.
- Strategy3: ***Concept tree ascension.***
  For any attribute, if there exits a higher level concept in the concept tree for a given attribute, the substitution of the attribute value by its higher level concept should take place. Thus the tuple covers more cases than its original form "more general". Ascension the concept tree one level at time ensures minimal generalization (reduction of the chance of overgeneralization). should be enforced by ascending the tree one level at time. As a result of the execution of this strategy, identical tuples are produced.
- Strategy4: ***Vote propagation.***
  The vote of generalized tuples registers the number of identical tuples. When merging identical tuples, the votes should be accumulated in the generalized relation.
- Strategy5. ***Threshold control on each attribute.***

Each attribute has threshold (number of distinct values per attribute) associated with it and if the number of distinct values for a given attribute in the generalized relation is greater that the attributes' threshold then generalization on this attribute should be performed.

- Strategy6. ***Threshold control on generalized relations.***
  The generalized relation has threshold associated with it that represents the number of tuples in the final generalized relation. If the number of tuples in the generalized relation is greater than the generalization threshold then further generalization on the relation should be performed.
- Strategy7. ***Rule transformation.***
  When the generalization process has finished, (the attributes thresholds and relation threshold are met), the final generalized relation is transformed into rules of conjunctive normal form, or rules of disjunction of conjunction form.

3.2.2 The idea of AOI

"Many knowledge discovery methods have been developed in recent studies for mining knowledge from data, ranging from database research, generalization, knowledge representation, to mathematical or statistical modeling, etc. based on the kinds of knowledge to be discovered from data, knowledge discovery tools can be classified into major classes : (1) generalization-based discovery, and (2) discovering knowledge without generalization, in each class, further classifications can be performed based on the kinds of rules or the form of knowledge to be discovered, including knowledge rules (characteristic, discriminant, clustering, dependency or association rules), generalized relations and multiple layered databases, etc. these discovery methods and the discovered knowledge will contribute to intelligent query answering." [15]

The basic idea of the AOI follows the generalization-based discovery process where basic data values in the database are substituted basic by higher level concepts in the appropriate concept hierarchy. This called the generalization process by which some meaningful knowledge can be extracted. As it has been mentioned earlier, the generalization process is controlled by two thresholds; ***attribute (column) thresholds*** and ***table threshold***. The ***attribute (column) thresholds*** specify the maximum number of distinct values that may appear in each column of the generalized relation. The ***table threshold*** specifies the maximum number of tuples that may appear in the final generalized relation.

The DM-AIG system is based on the concept of the attribute-oriented generalization algorithm that takes a database relation as input and generalizes it to some certain level. The generalization

process is guided by a set of concept hierarchies, attributes thresholds and table threshold. This generalization constitutes the discovery task of the system. The attributes thresholds and table threshold are user defined values.

The systems' process starts with obtaining what is so called the *prime relation*, which is accomplished by the retrieval of the relevant data from the database in use. The prime relation is tested for single value attributes and to many distinct values attributes and such attributes will be removed. The generalization process starts by generalizing each attribute according to its threshold (column threshold). This process is an iterative one and identical tuples are collapsed together and the vote attributes keeps track of the number of identical tuples and redundant tuples are eliminated by removal. This process continues until the table threshold is met. In other words, the total number of tuples in the final generalized relation must be less than or equal to the table threshold. The final generalized relation is then presented to the user for further adjustment (i.e. generalization levels, sort the relation according to some order, or display numerical summary information, etc ...). The over all process of the system is governed by a simple algorithm detailed in the following subsection.

3.2.3 Basic algorithm

According to [1], the basic idea of attribute-oriented induction is summarized in the following algorithm:

Inputs:

- A relational database.
- The learning task (learning data mining query).
- Set of concept hierarchies.
- Generalization Threshold (GT).

Outputs:

Final Generalized Relation (GR) that is transformed into a characteristic rule learned from the database. The result is displayed by one of the *visualization tools*

The algorithm of AOI consists of the following steps:

Step 1: Collect the task-relevant data by the application of some SQL statements.

Step 2: Perform the following attribute-oriented induction steps:

```
Begin
    For each attribute A_i (1 ≤ i ≤ n), where n = number of attributes in GR Do
        While number of distinct values in A_i > threshold Do
            If
                No hierarchy table for A_i Then Remove A_i
            Else
                Replace A_i's by its corresponding minimal generalized concept;
                Merge identical tuples;
            End if
        Wend
    Next for
        While (number of tuples in GR > threshold) Do
                Selectively generalize attributes;
                Merge identical tuples;
        Wend
End
```

Step 3: Simplify the generalized relation.

Step 4: Transform the final relation into a logical rule (characterization rules).

3.3 System architecture

The general architecture of the DM-AIG system is depicted in figure3.1. For the time being, the system can discover only one of type knowledge in the form of characteristic rules, but it may be improved to discover other type of rules such as; discrimination rules, decision rules, etc.

The system consists of six components, which are:

- User-interface is the part by which the system communicates with the user for the mining request.
- ADODC is Visual Basic-6 component that is used to access the data in a database to find the structure of the used database and its' properties such as; relations names, fields names, sizes. This component is used to find the structures of the concept hierarchies as well.
  A sample VB code can be found in the appendix.
- User request is the query that is submitted by the user to the system.
- Learning request is the translation of the user request into SQL.
- Database is the requested data to be used, which is the source of the knowledge.
- Knowledge base consists of the concept hierarchy relation (table) and the thresholds (table and attributes) for the generalization process.

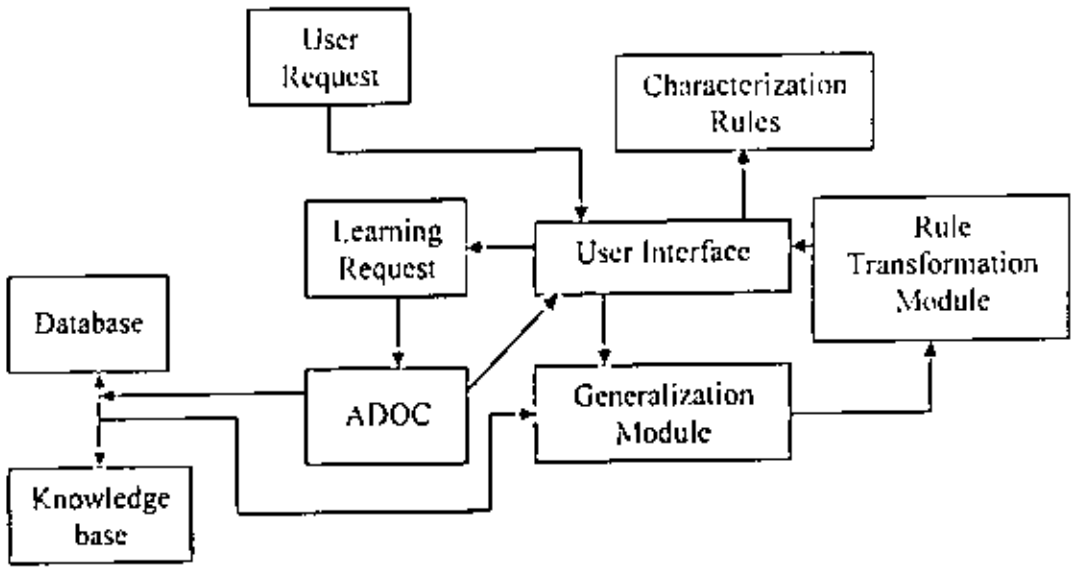

Figure 3.1: The architecture of DM-AIG system

3.4 System modeling

Usually a data mining system tasks consists of three important phases namely; model building phase, testing phase and applying phase. Here we will deal with the model building phase and the other two phases will be briefly mentioned and deal with them in more details in the next chapter.

3.4.1 Model building phase

The model modeling phase can start by building a simulated model of the problem. This model will provide a clear understanding of the problem in question.

From the literature, there are three perspectives that are used in the development of simulated models and these are:

The first perspective is to use some Graphical User Interface (GUI) tools to develop the simulated model on the screen, use ares to connect the system components to create the logical model and the run the simulated model. "In most cases, due to the limitation of the simulation program under use, some simplifications and approximations to the model may be required. Such simplifications or approximations can be very costly." [18].

The second perspective supports the belief that any simulation program would no be able to model all tasks of systems without the need to make some modification. "This suggests that models should be developed from scratch by using a simulation modeling language." [18]. This approach

may increase the time needed to produce the system and may divert the developer to pay more attention to the programming challenges than understanding the system.

The third perspective focuses on the use of a GUI that will automatically generate code with the possibility of the developer to intervene to make some changes to the code to match the system needs. This is very popular because it reduces the need time to produce the system, on the other hand code modification is tedious task.

The building model phase consists of a number of steps and these are:

- The first step in the building model phase is to load the database that will be used to discover knowledge from. This database is called the source data. At this step a connection is established between the database and the system. Our system is currently designed to work only with access database type. Loading the database is one part of the input to the system.
- The second step in the building model phase is to load the knowledge base. The knowledge base consists of:
  1. The first part of the knowledge base consists of an Access database file that consists of a concept hierarchy for an attribute in the database. The number of files that contains concept hierarchies must be equal to or less than the number of attributes (some times this called hierarchy threshold condition) in the used database. A concept hierarchy file name should have the same name as one of the attributes the source database. It is also possible to rename concept hierarchy file name to have the same name as one of the attributes otherwise the concept hierarchy for a given attribute will be considered as unavailable. Only one concept hierarchy can be defined per attribute.
  2. The second part of the knowledge base is to define some thresholds (column threshold and table threshold) to limit generalization process. The purpose of these thresholds is for tuning the results of the discovery task.
- The third step in the building model phase is to receive the query from the use.
  From the source data and the concept hierarchies, the system will display all attribute names and their domains to the user so he/she can formulate the needed query.
- The fourth step in the building model phase is to use the user query to select the task relevant data and to ignore the rest of the data. The relevant data will be stored in what is called task relevant relation. If a query has not been defined, this might result in over generalization in the final generalized relation. When ascension in a concept hierarchy takes place, some

information details are lost. "In order to keep as much information as we can, the proper time to do generalization should be chosen very carefully. If an attribute containing lower leveled concept could be used to form new sub-partitions at the next partitioning stage, generalization should not be performed on it. "[16].

- The fifth step in the building model phase is to remove any attributes that have no concept hierarchies for them. This process will produce what is known as prime relation. Next, is to start the generalization process of the task relevant data in the prime relation by the used of the define concept hierarchies and controlled by the table and attribute thresholds. The attribute threshold specifies the maximum number of distinct values that may appear in the domain of a given attribute in the prime relation. The table threshold specifies the maximum number of tuples that may appear in the final generalized relation. "Such a threshold may also be preset in the data mining system (usually within a range of 10 to 30), or set by an expert or user, and should be adjustable." [21]. the generalization process is an iterative one until the specified thresholds are met. The generalization process is the main task of our system. At the end of the generalization process; the final generalized relation will be produced.
- The sixth and final step is to characterize the data in the final generalized relation.

The over all process is depicted in figure3.2.

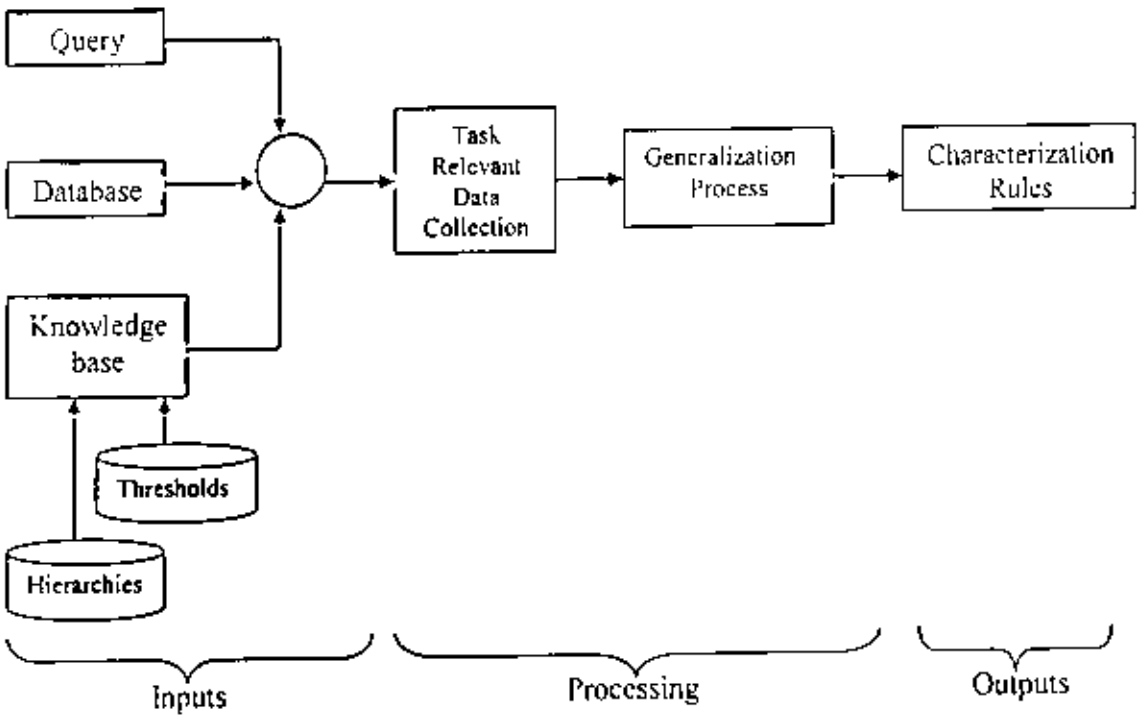

Figure 3.2: The build process of the system.

3.4.2 Model testing phase

After the model is built, the model must be tested via sample of databases in the form of experiments. Such experiments will deal with different database varying in number of records and number of attributes. The model testing phase will be detailed in the next chapter.

3.4.3 Model applying phase

After the testing phase has resulted in satisfactory results, then the model can be put in use in the real world.

3.5 Physical Design

3.5.1 Design Considerations

There are number of aspects that must be taken in consideration during the process of software design. Each of these aspects should contribute toward achieving the goals of the software under development. Some of these aspects are:

- Extensibility: There should be no major changes to the original architecture of the software in case of new capabilities are need to be added.
- Robustness: The software should tolerate unpredictable or invalid input.
- Reliability: The software should perform all of its required tasks for a specified period of time.
- Fault-tolerance: The software should be able to recover from some of its component failure.
- Security: The software should be able to block any unauthorized access.
- Maintainability: The software can be restored to a specified condition within a specified period of time. For example, antivirus software may include the ability to periodically receive virus definition updates in order to maintain the software's effectiveness.
- Compatibility - The software should be able to be incorporated with other products even when some new version has been issued.
- Modularity: The software should be composed of modules each of which can be implemented and tested in isolation before integrated in the over all system. The modularity leads to an ease in maintenance and development of the system.

- Reusability: Each module of the system should capture the essence of its functionality and nothing and nothing more or less. This single-task approach renders the components reusable wherever similar needs arise.

3.5.2 Design methodologies

The purpose of design methodologies is to provide a framework for the actual design of a system. They aim is to simplify the of design process and to enforce some standard design principles that improve the quality of the design. One of the earlier design methodologies is the Data Flow Diagram (DFD) and still considered to be one of the best modeling techniques to represent the processing requirements of a system. "The DFD is an excellent communication tool for analysts to model processes and functional requirements. One of the primary tools of the structured analysis efforts of the 1970's." [22]. DFD is a significant modeling technique that explains the course or movement of information in a process. The flow of the information in a process is based on the input and the output. A DFD can represent technical or business processes to illustrate the data flow from a process to the next and finally to the final results. DFD is common tool for a designer to show the interaction between the system and outside entities. According to [23]; the DFD notation is composed of four basic symbols as shown below:

- Process notations

  The process component of the DFD is represented graphically by a circle as depicted in figure 3.3(a). The process is also called a bubble, a function or a transformation. The task of this component is to give and indication of transforming the input to output via the execution of some process on the data.

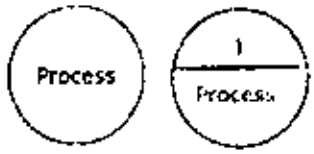

Figure 3.3(a): Process notations.

- Datastore notations

  Datastores component are repositories of data in a system and represented by a rectangle shape opened from one of its sides as depicted in figure 3.3(b). Datastores are refereed to as files.

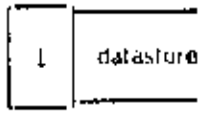

Figure 3.3(b): Datastore notations.

- **Dataflow notations**

  Dataflow component represents the pipelines by which packets of information flow from one process to another. Dataflow components are represented by arrows of different shapes labeled with the name of the data to be moved as depicted in figure 3.3(c).

Figure 3.3(c): Dataflow notations.

- **External entity notations**

  External entities component represent the means by which the system communicates with the outside world. In other words they represent the systems sources and destinations of the input and the output. External entities are indicated by rectangular or square shapes with the name of the source or destination as shown in Figure 3.3(d).

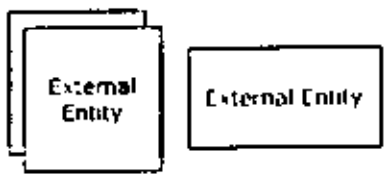

Figure 3.3(d): External entity notations.

From the above details of the DFD components, the process of the DM-AIG system using DFD terminology is depicted in figure3.4. According to [20], the efficiency of AOI algorithm can be analyzed as follows:

- The efficient retrieval of the work relation W' depends on the query processing methods used. Given a successful implementation of the systems, this step is expected to have good performance.
- Scanning the relation at most once. The cost for computing the minimum desired level and determining the mapping pairs (v, v') for each attribute is dependent on the number of distinct values for each attribute and is smaller than n (the number of tuples in the initial relation).
- The derivation of prime relation, P is performed by inserting generalized tuples into P. If there are a total of n tuples in W' and p tuple in P, for each tuple t in W', we substitute its attribute values based on the derived mapping pairs. This results in a generalized tuple t', each t' takes

$O(\log p)$ steps to find the location for count incrimination or tuple insertion. Thus the total time complexity is $O(n \times \log p)$ for all of the generalized tuples.

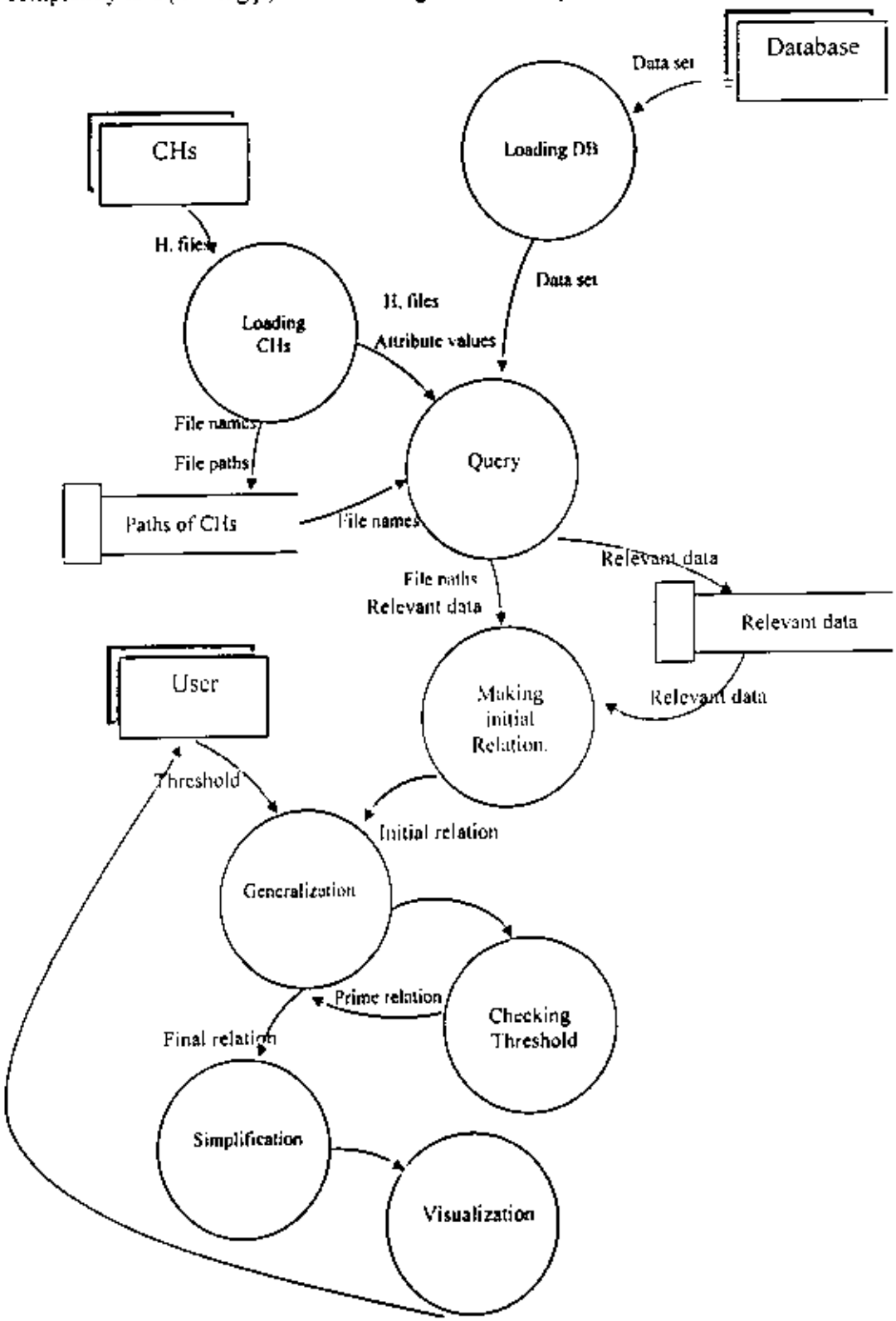

Figure 3.4: DFD for the DM-AIG system

Chapter 4

System testing and results

System testing is an empirical technical investigation phase that is conducted to convince beneficial users with the quality of the system functions and capabilities in the context in which it intended to provide. This process is mainly concern with applying the system with the intent to find error and pitfalls. Testing software can never completely guaranty the correctness of the software and so testing can provide a criticism or comparison of the state of the software to some given specifications. Software testing should not be confused with Software Quality Assurance (SQA) where the latter is concerned with a lot of aspects one of which is testing.

The purpose of this study is to produce knowledge at different levels of granularities by the implement the AOI algorithm. In order to accomplish this task, it was necessary to test our system with different types and sizes of data, and to develop a method to evaluation the results.

To test the effectiveness of our system, we have conducted some experiments using two synthetic datasets (databases) and one of real dataset. The real data used in our experiments is available on public domain on the Internet [21].

One of the most important dataset that has been used in testing softwares concerning AOI is from Natural Science and Engineering Research Council of Canada (NSERC) but this data is not available from the public domain. So, synthetic data was generated randomly to be used in testing our system. The parameters used for the data and system are as depicted it table 4.1.

Parameters	Description
N	The number of attributes in initial relation
T	The number of transactions (tuples) in initial relation
AT	The attribute threshold
GT	The generalization threshold
CH	The number of hierarchy files can be loaded

Table 4.1: Parameters for the data and system.

All of our the experiments are performed on a PC with Microsoft Windows XP professional operating system with a processor speed of 2.00 GHz, RAM size of 512MB and hard disk of size 80 GB.

The PC computer is also equipped with Microsoft® Universal Data Access Component 2.5 or 2.6 (MDAC 2.5 or MDAC 2.6) also a reference to Microsoft® Service Component OLEDB Service Component 1.0 stored as: "C:\Program Files\Common Files\System\OLE DB\oledb32.dll.

4.1 Synthetic data experiments

For synthetic data experiments, we have generated four datasets with different sizes and number of attributes. Two of them are of small size and the other two of large size.

4.1.1 Small synthetic data experiments

The first synthetic data experiment is concerned with information about students within a University environment. This database has the same format as the one used by the NSERC. The description of this database is as follows:

- Database file name is S_synthetic_University.mdb.
- Database size is 10,000 records or instances.
- Number of attributes is 5.
- Attribute names and values are the same as the ones used in the NSERC database and are depicted in table 4.2.
- Hierarchies used for this database are the same as the ones used in the NSERC database and are depicted in table 4.3.
- Hierarchy files names are: Status.mdb, Major.mdb, Birthplace.mdb, GPA.mdb.

No.	Attribute names	Description of values
1	Name	Student names
2	Status	M.A., M.S. Ph.D., Junior, Sophomore, Senior, Freshman
3	Major	History, math, literature, physics, Chemistry, Computing, Biology, Music, Statistics
4	Birthplace	Vancouver, Calgary, Edmonton, Ottawa, Bombay, Richmond, Victoria, Shanghai, Burnaby, Nanjing, Toronto
5	GPA	From 1.0 To 4.0

Table 4.2: Description of the University student database.

Hierarchy	descriptions
{Computing, math, biology, statistics, physics}⊂ science	
{Music, history, literature}⊂ art	
{science, art }⊂ any (major)	
{Freshman, sophomore, junior, senior}⊂ undergraduate	
{MS., MA., PHD.} ⊂ Graduate	
{undergraduate, Graduate}⊂ any (status)	
{Burnaby, Vancouver, Victoria, Richmond}⊂ British Columbia	
{Calgary, Edmonton}⊂ Alberta	
{Ottawa, Toronto}⊂ Ontario	
{Bombay}⊂ India	
{Shanghai, Nanjing}⊂ China	
{China, India}⊂ foreign	
{British Columbia, Alberta, Ontario}⊂ Canada	
{foreign, Canada }⊂ any (Birthplace)	
{0.0 – 1.9} ⊂ poor	
{2.0 – 2.9} ⊂ average	
{3.0 – 3.4} ⊂ good	
{3.5 – 4.0} ⊂ excellent	
{poor, average, good, excellent} ⊂ any (GPA)	

Table 4.3: Concept hierarchies table of the university database.

This experiment has been conducted on postgraduate student where the query was "Status= graduate", this query is depicted in figure 4.1, and it has resulted in a match of 4323 tuples out of the 10,000 tuples that has been stored in the task relevant data relation. This experiment has been carried out into four different versions with different attribute threshold (2, 6, 12, and 30) for the other remaining attributes (Major, Birthplace and GPA). The resulted number of tuples in the final generalized relation is depicted in table 4.4.

Experiment version	Attribute threshold	Number of tuples in the final generalized relation
Version-1	2	16 tuples
Version-2	6	40 tuples
Version-3	12	180 tuples
Version-4	30	540 tuples

Table 4.4: Small synthetic data experiments results.

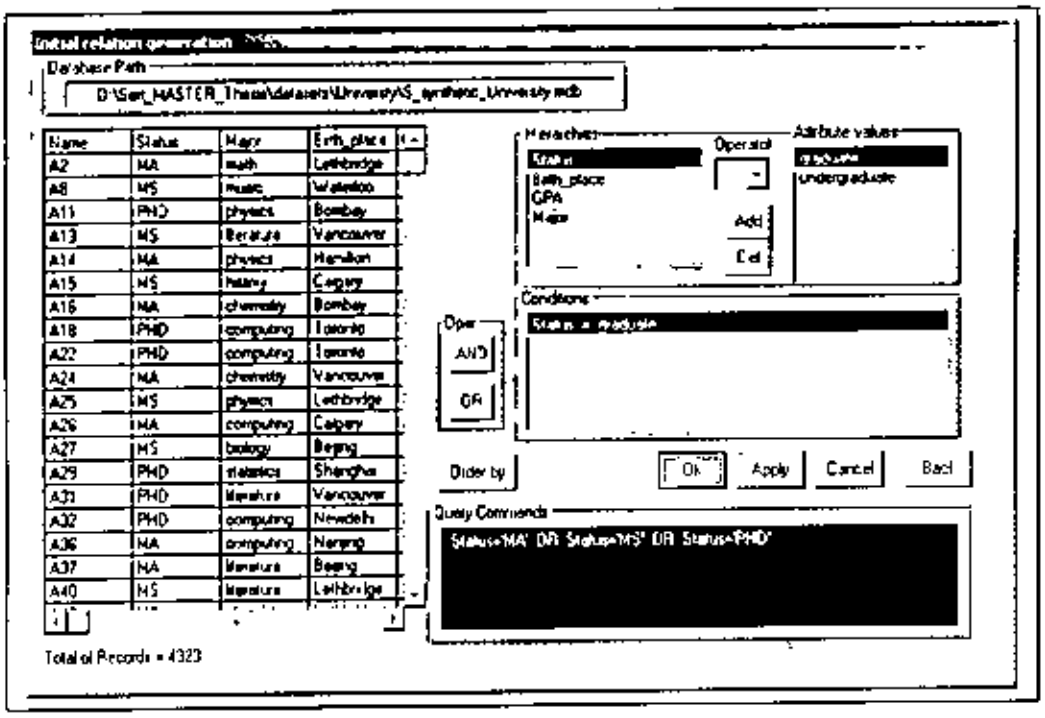

Figure 4.1 query when "Status= graduate.

The final obtained results from our system for the small synthetic data experiments results are depicted in figure 4.2, figure 4.3, figure 4.4 and figure 4.5 according to the above mentioned threshold respectively.

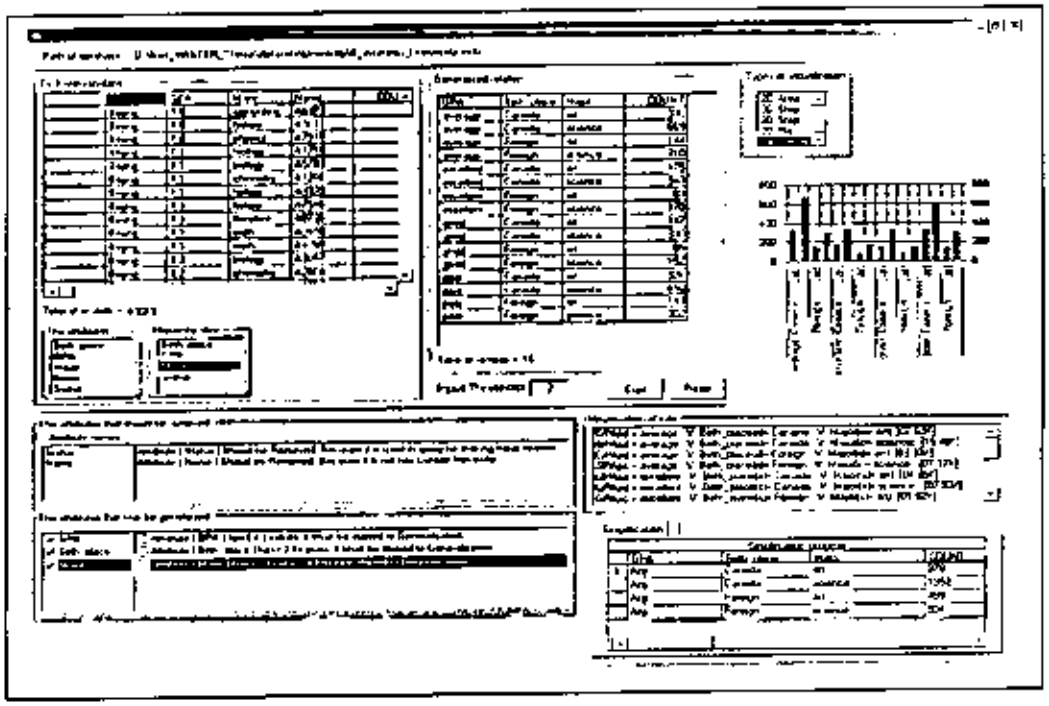

Figure 4.2: Final generalized relation for threshold=2.

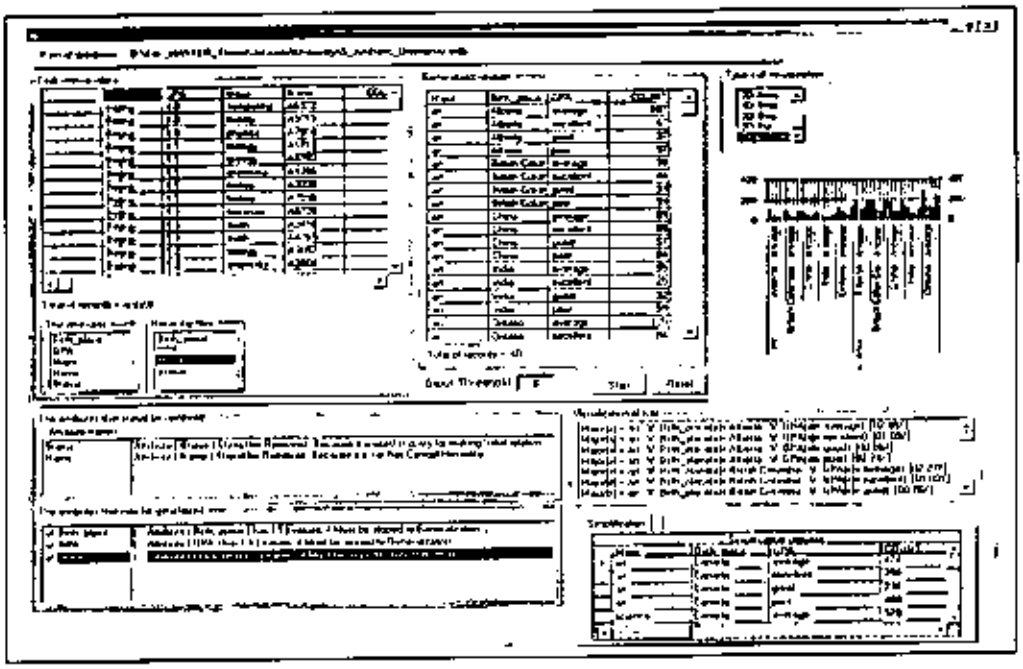

Figure 4.3: Final generalized relation for threshold=6.

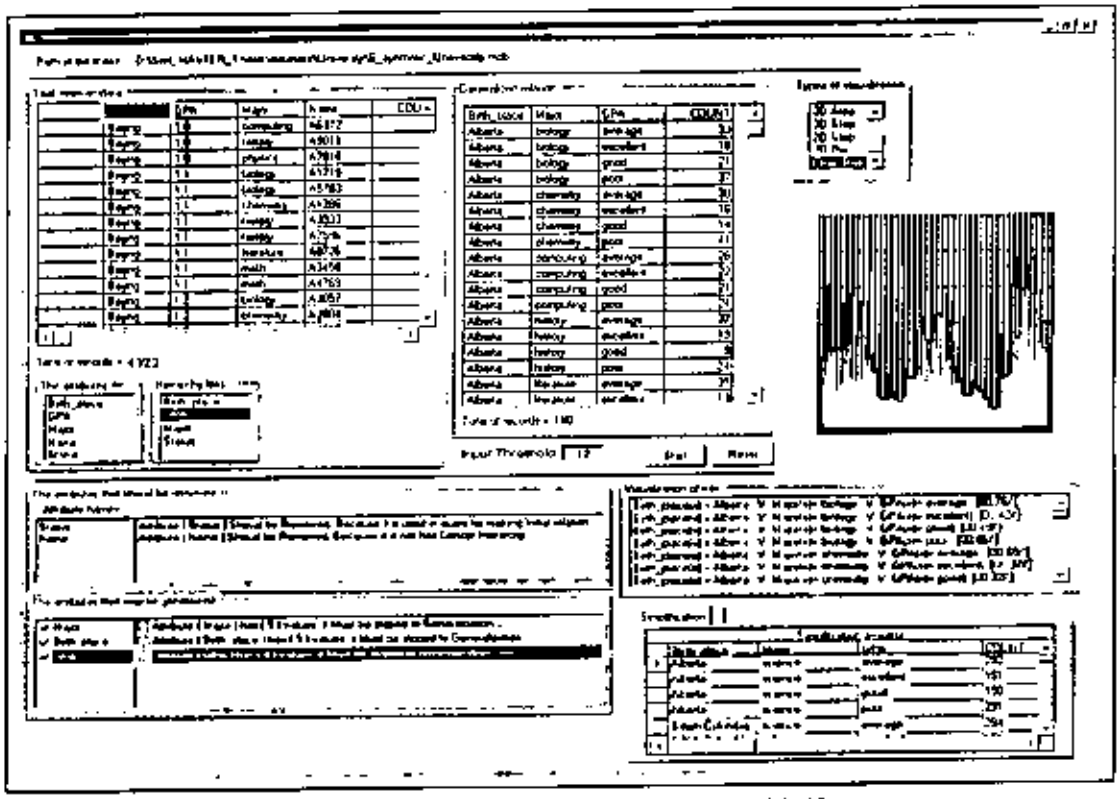

Figure 4.4: Final generalized relation for threshold=12.

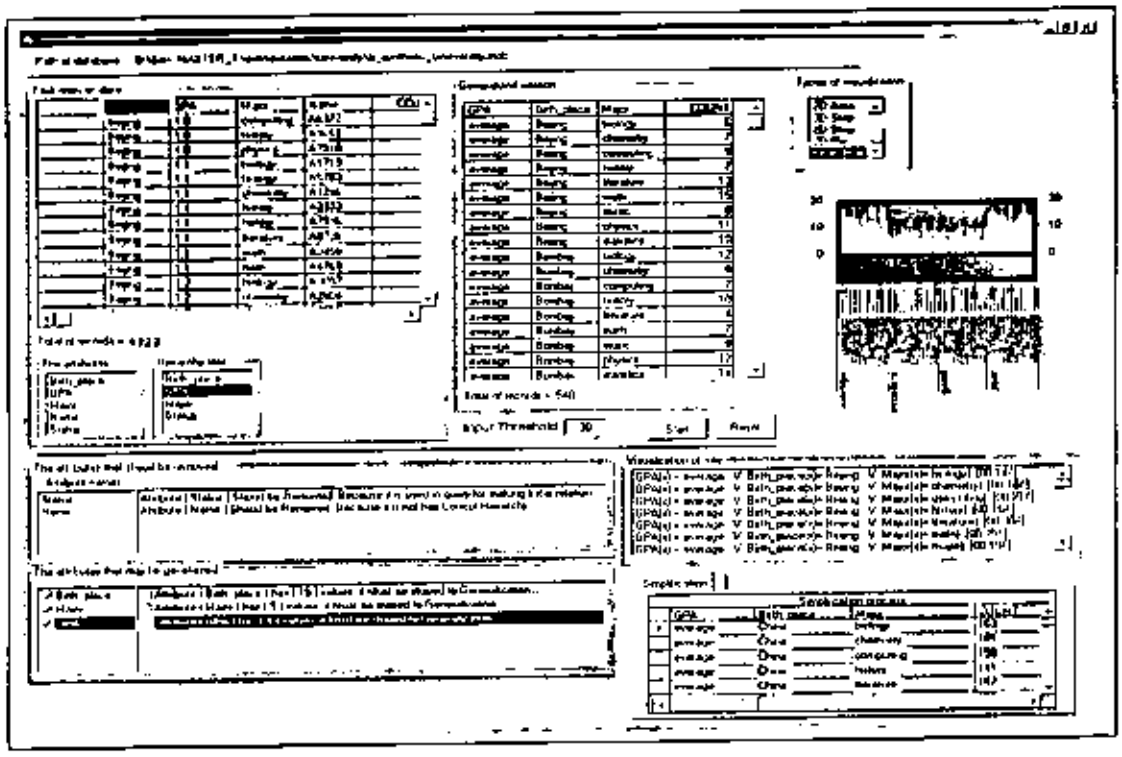

Figure 4.5: Final generalized relation for threshold=30.

The rules from our system for the small synthetic data experiments results, when the threshold =2, are depicted in figure 4.6.

path of database
D:\Art_MASTER_Thesis\datasets\University5_synthetic_University.mdb

Virtualization rules

GPA(x) = average V Birth_place(x)= Canada V Major(x)= art) [07.63%]
GPA(x) = average V Birth_place(x)= Canada V Major(x)= science) [15.48%]
GPA(x) = average V Birth_place(x)= Foreign V Major(x)= art) [03.33%]
GPA(x) = average V Birth_place(x)= Foreign V Major(x)= science) [07.17%]
GPA(x) = excellent V Birth_place(x)= Canada V Major(x)= art) [04.05%]
GPA(x) = excellent V Birth_place(x)= Canada V Major(x)= science) [07.93%]
GPA(x) = excellent V Birth_place(x)= Foreign V Major(x)= art) [01.92%]
GPA(x) = excellent V Birth_place(x)= Foreign V Major(x)= science) [03.93%]
GPA(x) = good V Birth_place(x)= Canada V Major(x)= art) [03.40%]
GPA(x) = good V Birth_place(x)= Canada V Major(x)= science) [07.63%]
GPA(x) = good V Birth_place(x)= Foreign V Major(x)= art) [02.06%]
GPA(x) = good V Birth_place(x)= Foreign V Major(x)= science) [03.52%]
GPA(x) = poor V Birth_place(x)= Canada V Major(x)= art) [07.54%]
GPA(x) = poor V Birth_place(x)= Canada V Major(x)= science) [14.11%]
GPA(x) = poor V Birth_place(x)= Foreign V Major(x)= art) [03.31%]
GPA(x) = poor V Birth_place(x)= Foreign V Major(x)= science) [06.99%]

Figure 4.6: rules from small synthetic data experiments when threshold=2.

The second synthetic data experiment is concerned with information about car evaluation; this database has the same format as the one used for the evaluation of Hierarchy INduction Tool (HINT).

The description of this database is as follows:

- Database file name is S_synthetic_Car.mdb.
- Database size is 8,000 records or instances.
- Number of attributes is 6.
- Attribute names and values are the same as the ones used in the Car database and are depicted in table 4.5.
- Hierarchies used for this database are created according to the values of attributes that are used in car database[1] and are depicted in table 4.6.
- Hierarchy files names are: buying.mdb, maint.mdb, doors.mdb, persons.mdb.

No.	Attribute names	Description of values	Remarks
1	buying	vhigh, high, med, low	vhigh= very high
2	maint	vhigh, high, med, low	maint= maintenance
3	doors	2, 3, 4, 5 or more	
4	persons	2, 4, more	
5	lug boot	small, med, big	
6	safety	low, med, high	

Table 4.5: Description car database.

[1] The data is taken from [21].

Hierarchy	descriptions
{moderation, expensive } ⊂ buying	buying price
{med, low } ⊂ moderation	
{v_high, high }⊂ expensive	v_high= very high
{cheap, expensive }⊂ maint	price of the maintenance
{med, low }⊂ cheap	
{v_high, high }⊂ expensive	
{soprt, normal }⊂ doors	number of doors
{2, 3 }⊂ sport	
{4, 5more }⊂ normal	
{normal, abnormal }⊂ persons	capacity in terms of persons to carry
{more}⊂ abnormal	
{2, 4 }⊂ normal	

Table 4.6: Concept hierarchies table of the car database.

This experiment has been conducted on buying price of cars where the query was "buying = expensive". Executing this query has resulted in a match of 4004 tuples out of the 8,000 tuples that has been stored in the task relevant data relation. This experiment has been carried out into two versions with different attribute threshold (3 and 4) for the other remaining attributes (maint. doors and persons). The resulted number of tuples in the final generalized relation is depicted in table 4.7.

Experiment version	Attribute threshold	Number of tuples in the final generalized relation
Version-1	3	8 tuples
Version-2	4	12 tuples

Table 4.7: Small synthetic data experiments results.

The final obtained results from our system for these two experiments are depicted in figure 4.8, and figure 4.9 according to the above mentioned threshold respectively.

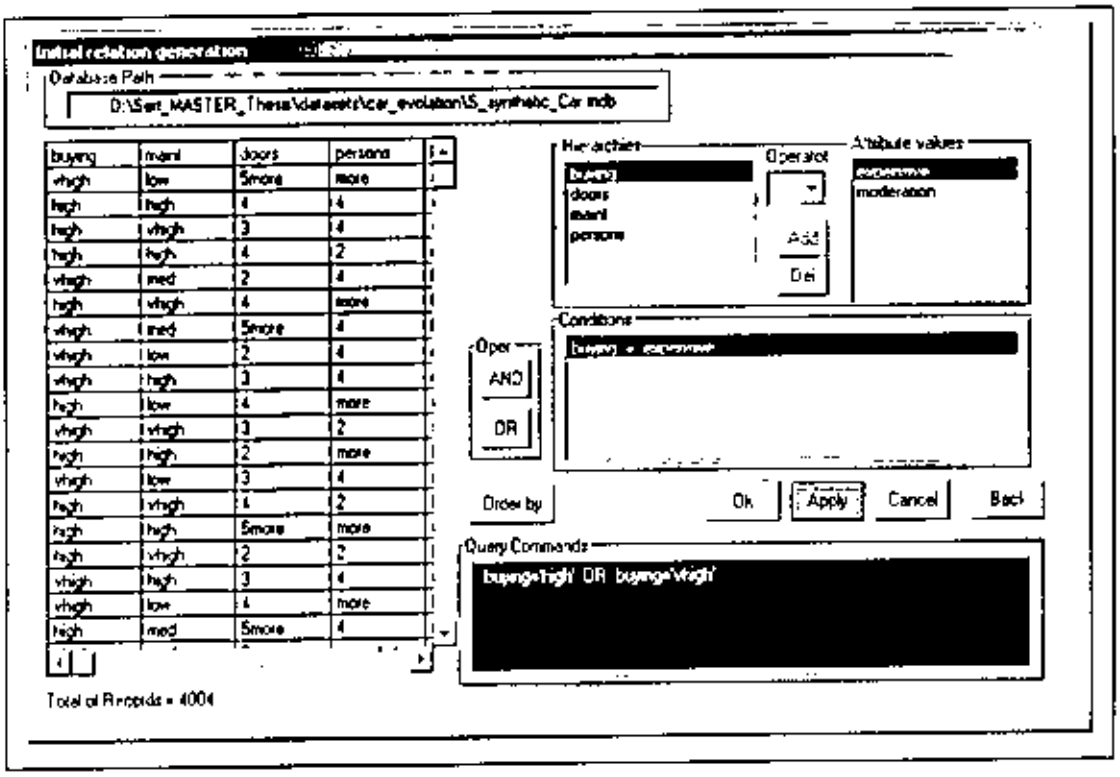

Figure 4.7 query when "buying = expensive.

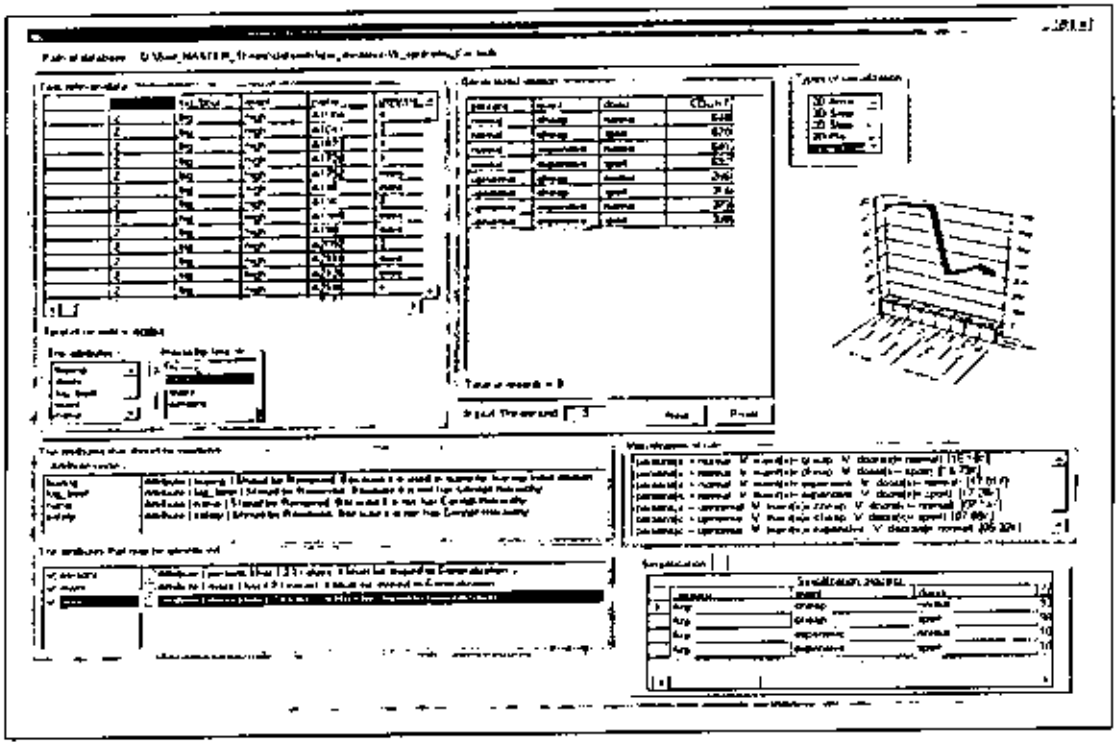

Figure 4.8: Final generalized relation for threshold=3

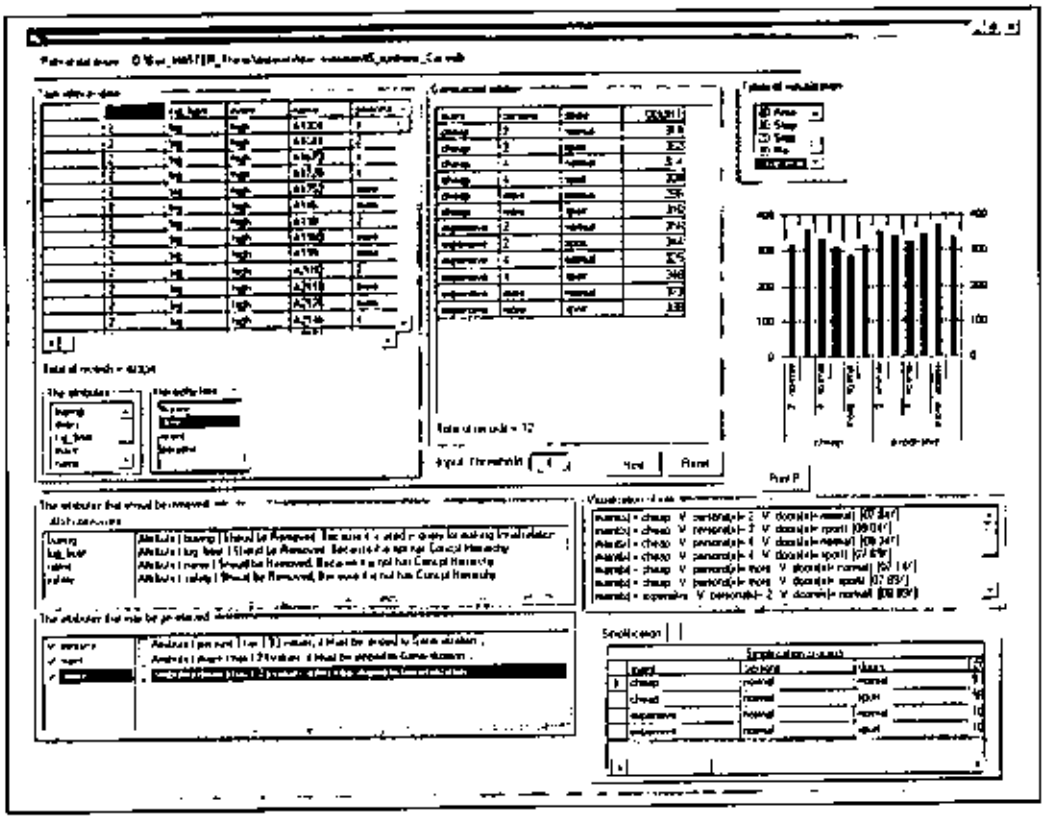

Figure 4.9: Final generalized relation for threshold=4.

4.1.2 Large synthetic data experiment

The first synthetic data experiment is concerned with information about students within a University environment. This database has the same format as the one used by the NSERC. The description of this database is as follows:

- Database file name is l._synthetic_University.mdb.
- Database size is 60000 records or instances.
- Number of attributes is 5.
- Attribute names and values are the same as the ones used in the NSERC database and are depicted in table 4.2.
- Hierarchies used for this database are the same as the ones used in the NSERC database and are depicted in table 4.3.
- Hierarchy files names are: Status.mdb, Major.mdb, Birthplace.mdb, GPA.mdb.

This experiment has been conducted on postgraduate student where the query was "Status= graduate". This query has resulted in a match of 45067 tuples out of the 60,000

tuples that has been stored in the task relevant data relation. This experiment has been carried out into four different versions with different attribute threshold (2, 6, 12, and 30) for the other remaining attributes (Major, Birthplace and GPA). The resulted number of tuples in the final generalized relation is depicted in table 4.8.

Experiment version	Attribute threshold	Number of tuples in the final generalized relation
Version-1	2	32 tuples
Version-2	6	80 tuples
Version-3	12	80 tuples
Version-4	30	240 tuples

Table 4.8: Small synthetic data experiments results

The final obtained results from our system for the large synthetic data experiments results are depicted in figure 4.10, figure 4.11, figure 4.12, and figure 4.13 according to the above mentioned threshold respectively.

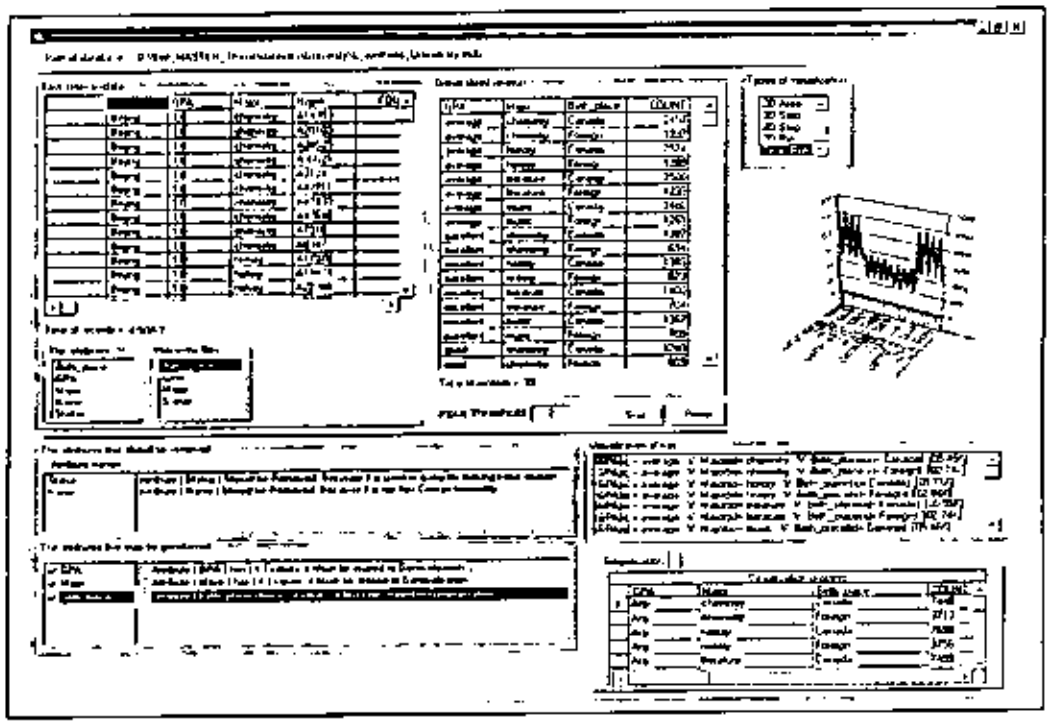

Figure 4.10: Final generalized relation for threshold=2

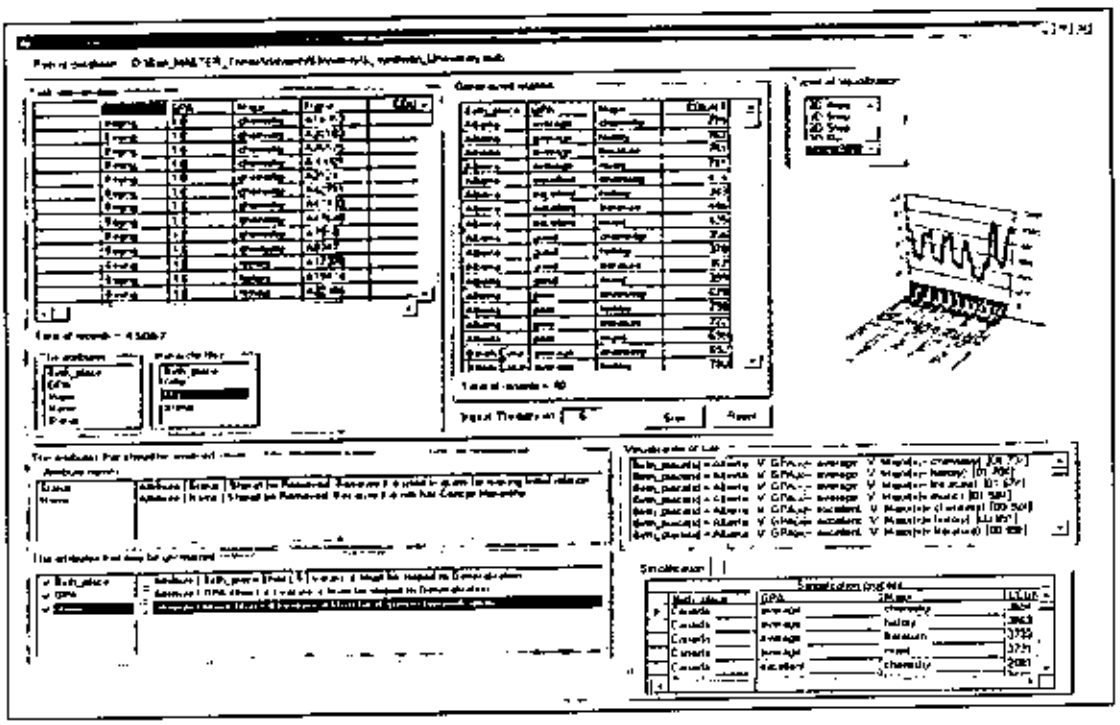

Figure 4.11: Final generalized relation for threshold=6

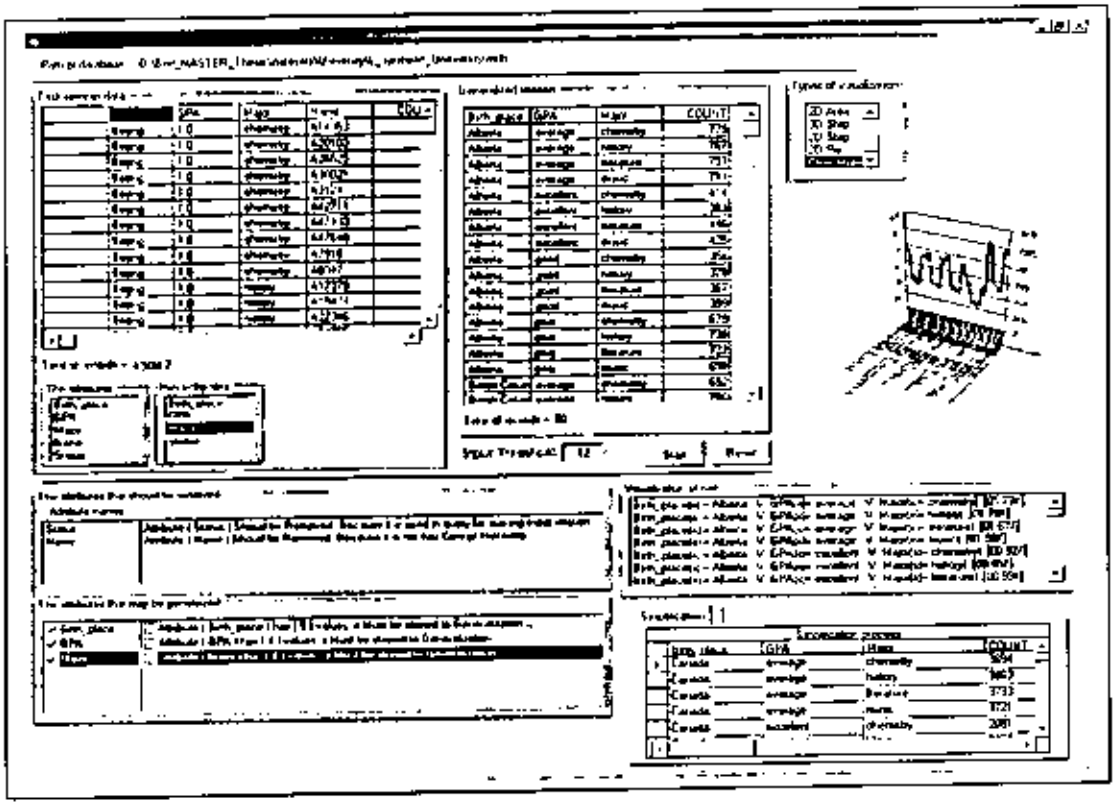

Figure 4.12: Final generalized relation for threshold=12

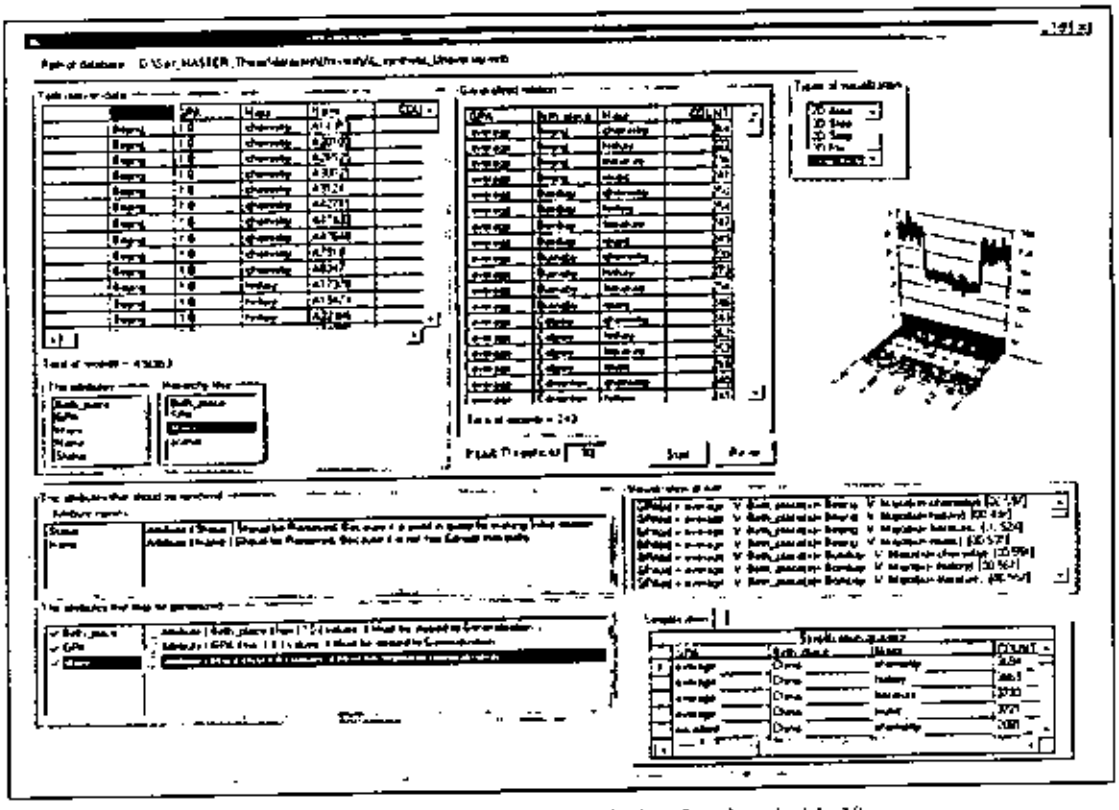

Figure 4.13: Final generalized relation for threshold=30

The second synthetic data experiment is concerned with information about car evaluation; this database has the same format as the one used for the evaluation of Hierarchy INduction Tool (HINT).

The description of this database is as follows:

- Database file name is L_synthetic_Car.mdb.
- Database size is 30,000 records or instances.
- Number of attributes is 6.
- Attribute names and values are the same as the ones used in the car database and are depicted in table 4.5.
- Hierarchies used for this database are the same ones used for the second synthetic data experiment as in table 4.6.
- Hierarchy files names are: buying.mdb, maint.mdb, doors.mdb, persons.mdb.

This experiment has been conducted on buying price of the car where the query was "buying = expensive". This query has resulted in a match of 14943 tuples out of the 30.000 tuples that has been stored in the task relevant data relation. This experiment has

been carried out into four different versions with different attribute threshold (3, and 4) for the other remaining attributes (maint, doors and persons). The resulted number of tuples in the final generalized relation is depicted in table 4.9.

Experiment version	Attribute threshold	Number of tuples in the final generalized relation
Version-1	3	8 tuples
Version-2	4	12 tuples

Table 4.9: Small synthetic data experiments results.

The final obtained results from our system for the small synthetic data experiments results are depicted in figure 4.14, and figure 4.15 according to the above mentioned threshold respectively.

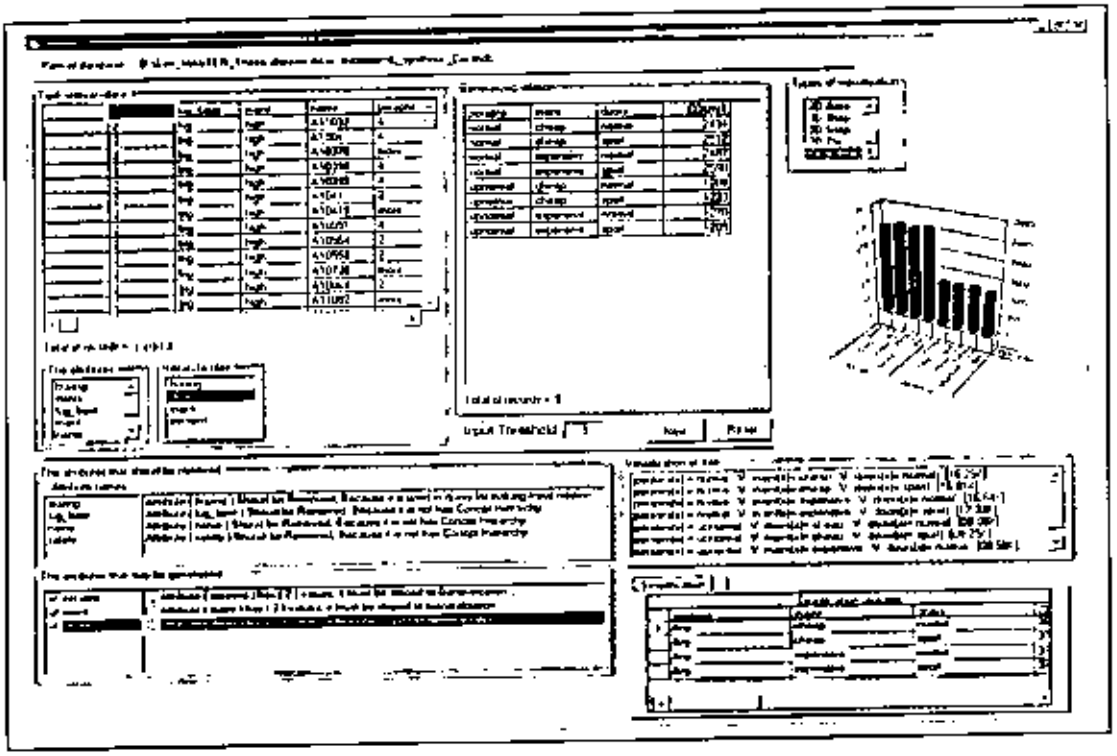

Figure 4.14: Final generalized relation for threshold=3

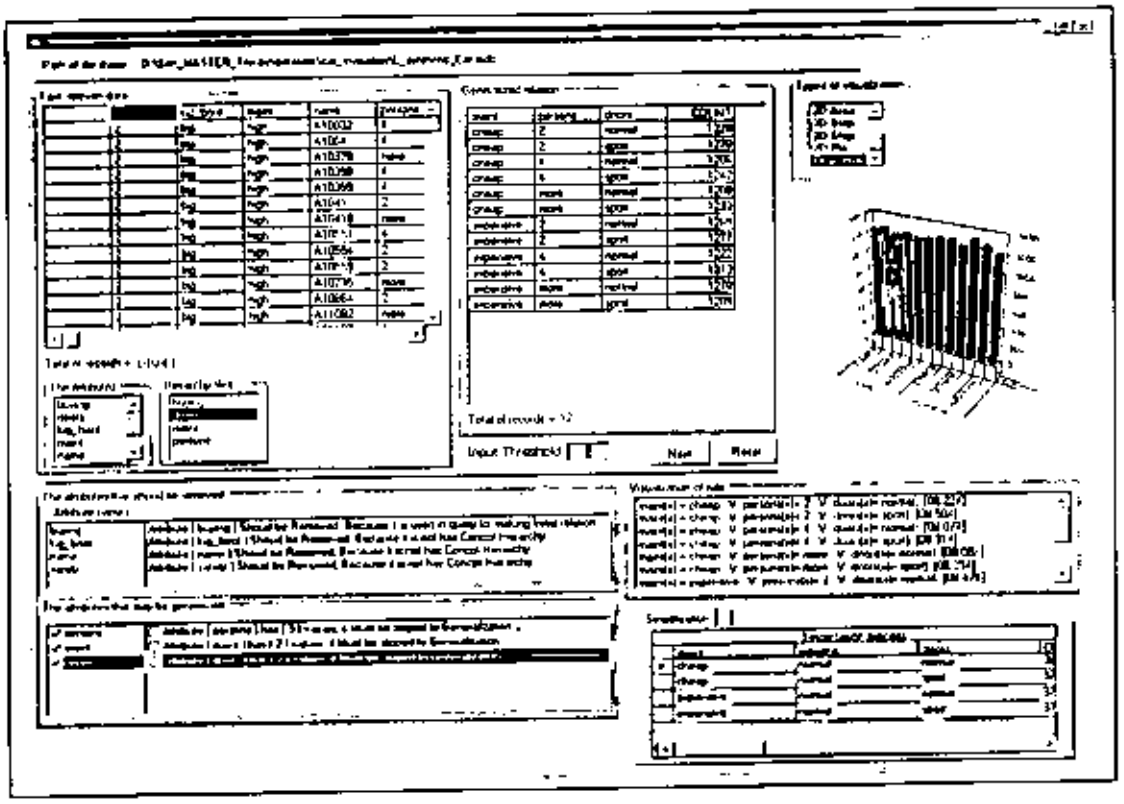

Figure 4.15: Final generalized relation for threshold=4

4.1 Real data experiments

The Real data experiment is concerned with information about car evaluation; this database has the same format as the one used for the evaluation of Hierarchy INduction Tool (HINT). This data consists of one relation "Cars". The description of this database is as follows:

- Database file name is RealData_Car.mdb.
- Database size is 1728 records or instances.
- Number of attributes is 6.
- Attribute names and values are the same as the ones used in the Car database and are depicted in table 4.5.
- Hierarchies used for this database are created according to the values of attributes that is used in car database and are depicted in table 4.6.
- Hierarchy files names are: buying.mdb, maint.mdb, doors.mdb, persons.mdb.

This experiment has been conducted on buying price of a car where the query was "buying = expensive". Snap shot of this query is depicted in figure 4.16 and it has resulted in a match of 864 tuples out of the 1728 tuples that has been stored in the task relevant data relation. This experiment has been carried out into two different versions with different attribute threshold (2 and 4) for the other remaining attributes (maint, doors and persons). The resulted number of tuples in the final generalized relation is depicted in table 4.10.

Experiment version	Attribute threshold	Number of tuples in the final generalized relation
Version-1	2	8 tuples
Version-2	4	12 tuples

Table 4.10: Small real data experiments results.

The final obtained results from our system for the small real data experiments results are depicted in figure 4.17, and figure 4.18.

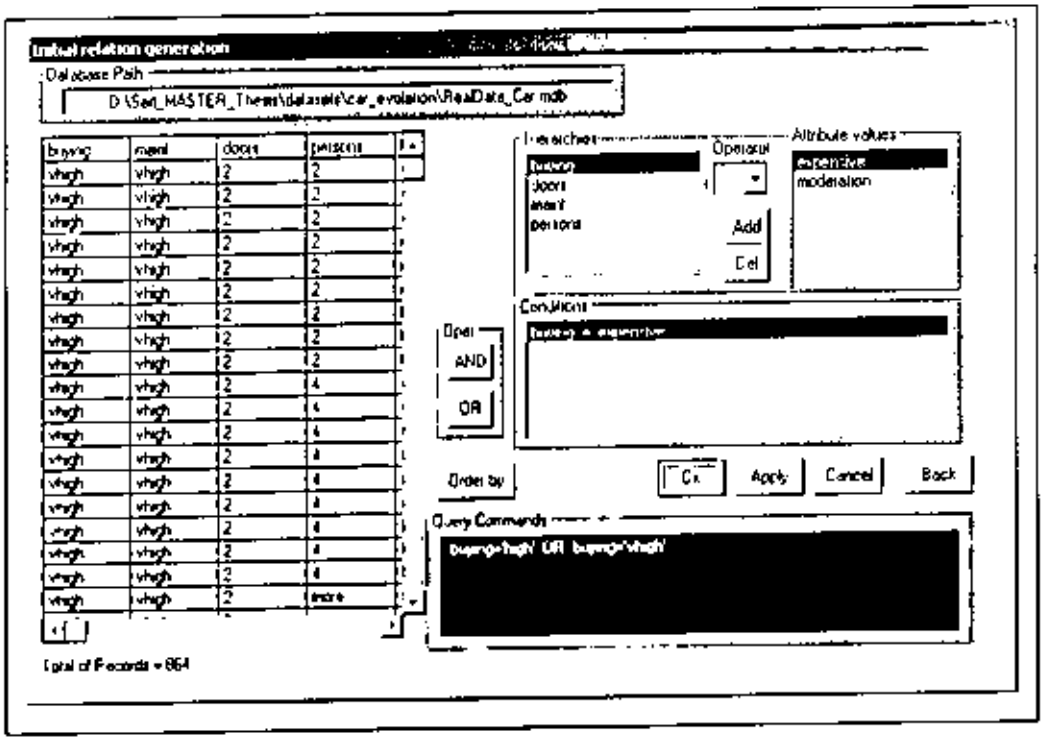

Figure 4.16: Query snap shot of "buying = expensive".

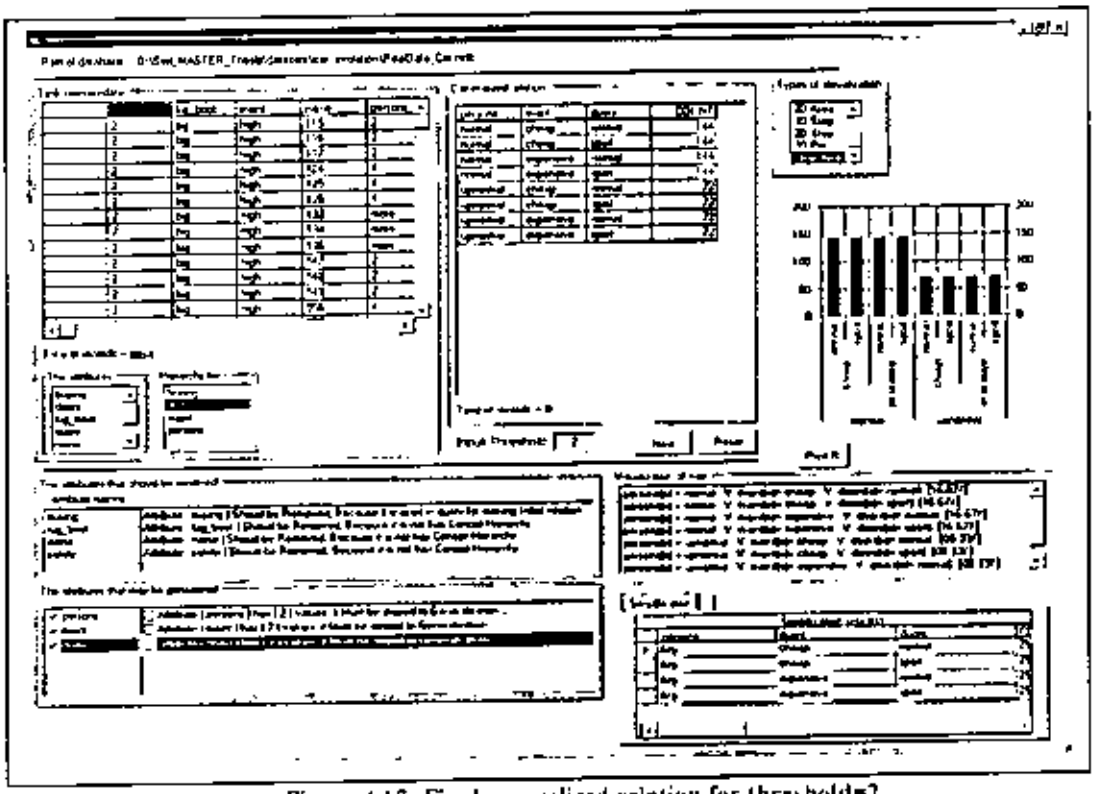

Figure 4.17: Final generalized relation for threshold=2

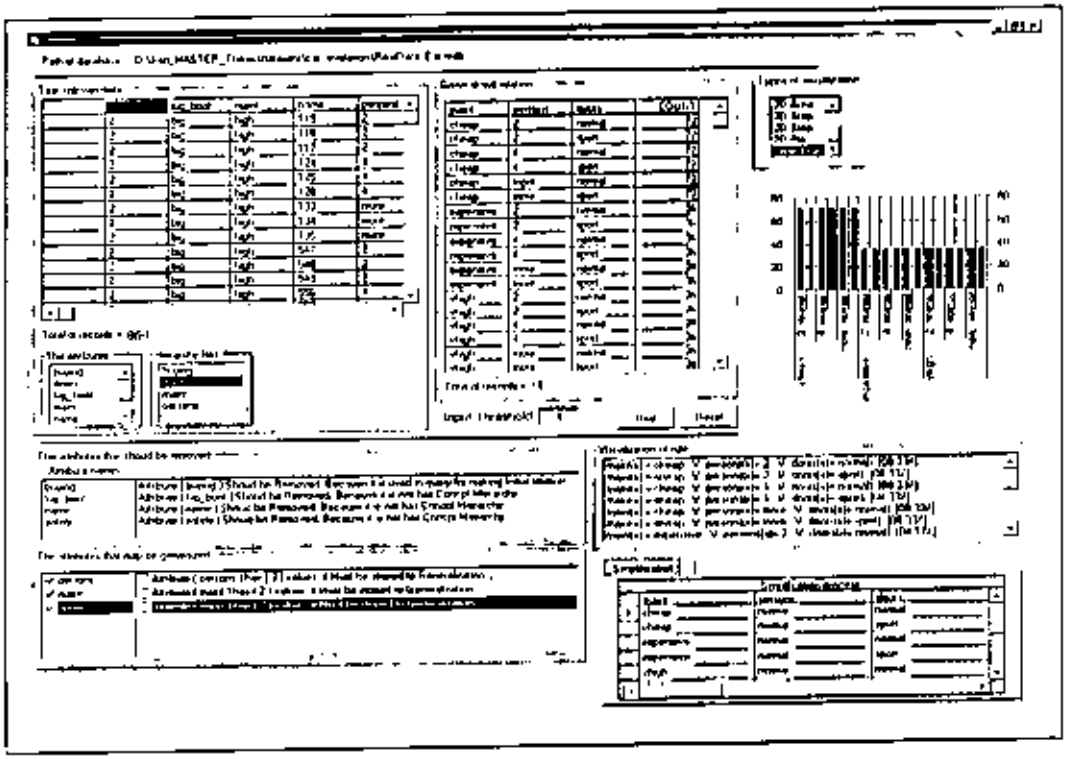

Figure 4.18: Final generalized relation for threshold=4

The rules from our system for the real data experiment results when the threshold =2 is depicted in figure 4.19.

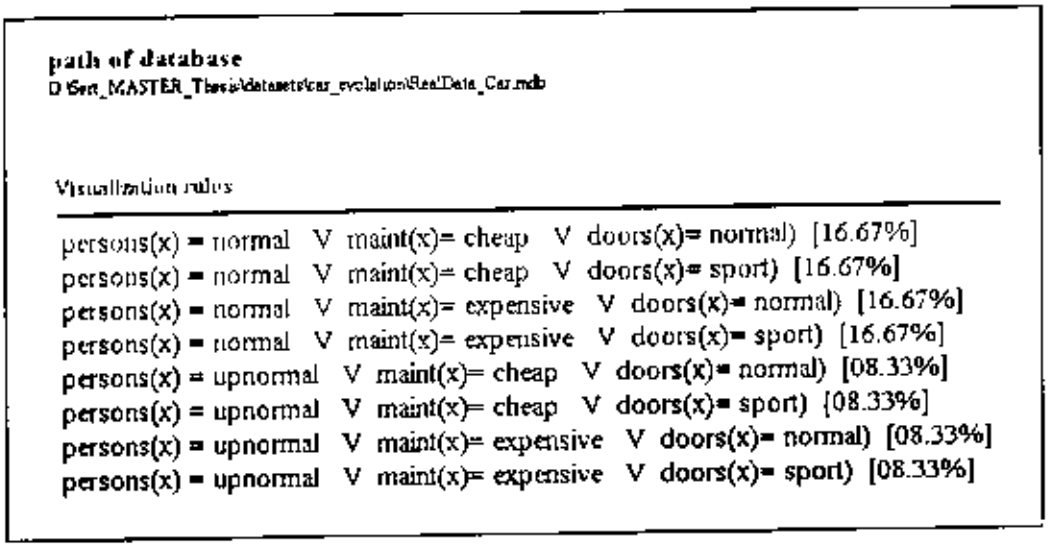

Figure 4.19: Rules from real data experiment when threshold=2.

Chapter 5

Conclusion and future work

5.1 System advantage

- The DM-AIG system performed interactive data mining at multiple concept levels on any user-specified set of data in access database using as SQL Data Mining Query Language, DMQL, or graphical user interface. Users may interactively set and adjust various thresholds, control a data mining process, perform roll-up or drill-down at multiple concept levels, and generate different forms of outputs, including generalized relations, generalized feature tables, multiple forms of generalized rules, visual presentation of rules, charts, curves, etc.
- Most existing learning algorithms which do not take full advantages of these database facilities, our approach primarily adopts relational operations, such as selection, join, projection (extracting relevant data and removing attributes), tuple substitution (ascending concept trees), and intersection (discovering common tuples among classes). Since relational operations are set-oriented and have been implemented efficiently in many existing systems, our approach is not only efficient but easily exported to many relational systems.
- The system deals with any number of attributes in the database (data set).
- The concept hierarchy can be renamed by the system to have the same name as one of the attributes in the database otherwise the concept hierarchy will be considered unavailable
- The system views the SQL commands to the user.
- The system is able to load the hierarchy files from any paths without using the same paths.
- The system communicates with various databases systems for data mining using the ODBC technology.
- The system is developed and adding procedure to applying the simplification operation in the final relation.

5.2 Conclusion

As it has been mentioned in chapter 4, we have conducted a total of 14 experiments with 5 different data sets. The differences in the data sets are in data types, sizes, threshold and other data related characteristics. The results of these experiments are summarized in table 5.1.

Experiments No.	Dataset name	Type of Data	No. of records	No. of attributes	No. of hierarchies used	Experiment versions	Attribute threshold	No. of tuples in final generalized relation
1	University students	Small synthetic data	10000	5	4	1	2	16
						2	6	40
						3	12	180
						4	30	544
2	Car buying	Small synthetic data	8000	6	4	1	3	8
						2	4	12
3	University students	Large synthetic data	60000	5	4	1	2	32
						2	6	80
						3	12	80
						4	30	240
4	Car buying	Large synthetic data	30000	6	4	1	3	8
						2	4	12
5	Car buying	Real data	1728	6	4	1	2	8
						2	4	12

Table 5.1: Summary of the experiments.

At this stage the author would like to make the following conclusions:

1. The use of concept hierarchies has proven to be a valuable tool in the process of performing data generalization.
2. The concept hierarchies are valuable tools in limiting the search space, which in fact one of the problems in AI domain.
3. The thresholds should be chosen very carefully so to protect the mining process from over or under generalization. The over generalization results in non-informative rules and under generalization will result in too long rules (too many conjunctions of disjunctions).

4. In every stage of the generalization process, the attribute generalization control should be left to the user because the user should have some idea which attribute should be generalized to what level in a subjective manner.

5. In comparison to tuple-oriented generalization, if there are p nodes in each concept tree and k attributes in the relation, then the total size of the search is p^k, on the other hand the attribute-oriented generalization has much smaller search space than tuple-oriented generalization.

5.3 Future work

After the development our system and the obtained results, the author would like to make a number of observations for future research work:

- The system may be developed to deal other types of database such as Oracal database, SQL Server database, etc...
- The system may be developed to discover other types of data mining tasks such as classification rules, discrimination rules, etc...
- The system may be developed to handle complex rule-based concept hierarchies.
- The system may be developed to generate the hierarchy files automatically.

References

1. J. Han, Y. Cai. N. Cercone. Knowledge discovery in database: An attribute-oriented approach. In proc. 18th Int . Conf. Very Large Databases, pages 547-559, Vancouver. Canada, August 1992.
2. El-Mouadib, F., PhD. Thesis in title of Taxonomy Formation By Approximate Equivalence Relations, Polish Academy of Science, Institute of Computer Science, Warsaw, Poland, 2000.
3. Fayyad U., Piatetsky-Shapiro G., and Smyth P., 1995. From Data Mining to Knowledge Discovery: An Overview", In Fayyad U., Piatetsky-Shapiro G., Smyth p. And Uthurusamy R. (eds.) Advances in Knowledge Discovery and Data Mining , AAAI Press, pp.1-34.AAAI Press/ the MIT Press,1996
4. Cai, Y., Cercone,N., and Han, J. 1991. Attribute-Oriented Induction in relation databases. In knowledge Discovery in databases, pages 213-228. AAAI/MIT Press, Cambirde. MA.
5. J. Han and Y. Fu, Exploration of the power of attribute-oriented induction in data mining. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge discovery and data mining. AAAI/MIT Press, 1995.
6. J. Han, y. Fu, S. tang. Advanced of the DBLearn system for knowledge discovering in large databases. In proc. 1995 Int'l joint conf. On Artificial Intelligence, Montréal, Canada, Aug.1995.
7. Wei Lu, Jiawei Han, Beng Chin Ooi. Discovery of General Knowledge in Large Spatial Database". GrantA-3723 and a research grant from Centre for System Science of Simon Fraser University.
8. Carter, C., and Hamilton, H. (1994) Performance improvement in the implementation of DBLEARN, TR CS-94- 05, University of Regina, Canada.
9. D.W. Cheung, A. W. Fu, and J. Han. Knowledge discovery in databases: A rule-based attribute-oriented approach. In Proc. 1994 Int'l Symp. On methodologies for Intelligent Systems, pp. 164-173, Charlotte, North Caroline, October 1994.
10. Xiaohua Hu, Knowledge discovery in databases: An attribute- oriented rougt set approach. Thesis for the degree of Doctor of Philosophy in computer science of Regina University, Suskatchewan, Canada, June1995.

11. W, J., Frawley,G. . Paitetsky-Shapiro and C. J. Matheus, knowledge Discovery in databases: An Overview, in G. Paitetsky-Shapiro and W. J.Frawley (eds.), knowledge Discovery in databases, AAAI/MIT Press 1991,1-27.
12. M. S. Chen, J. Han, and P. S. Yu. Data mining: An overview from a database perspective. IEEE Trans. Knowledge and Data Engineering, 8:866-883, 1996.
13. Yijun Lu. Concept hierarchy in data mining: specification, generalization and implementation. M.S. Thesis, department of Computer Science, of the University of Simon Fraser,1997.
14. David fudger, Howard j. Hamilton. A Heuristic for Evaluating database for knowledge discovery with DBLearn, In RSKD'93: International Workshop on rough set and knowledge discovery.
15. J. Han, Yue Huang, Nick Cercone, Yongjian Fu. Intelligent Query Answering by Knowledge discovery Techniques. In IEEE Trans. Knowledge and Data Engineering(accepted),1995.
16. Yaoyao Zhu , Unsupervised database discovery based on artificial intelligence techniques, M.S. Thesis, department of Electrical & Computer Engineering and Computer Science of the College of Engineering , of the University of Cincinnati, 2002.
17. [101] Introduction to Data Mining and Knowledge Discovery 2nd edition by Two Crows Corporation, *Third Edition* ISBN: 1-892095-02-5.
18. Hamad I. Odhabi, Ray j. Paul, Rebert D. Macredie: The four phase method for Modelling complex systems.Centre for Applied Simulation Modelling (CASM) Department of Information Systems and Computing Brunel University.proceeding of the 1997.
19. Tom Brijs, Koen Vanhoof, Geert Wets. Reducing redundancy in characteristic rule discovery by using integer programming techniques. 1999 . intelligent data analysis 4(2000) 229-240, 1088-467X/00/$8.00 © 2000 – IOS Press. All rights reserved.
20. J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.
21. Marko Bohanec, Blaz Zupan: UCI Repository of Machine Learning databases [http://ics.uci.edu/ml/datasets].Car Evaluation Data Set Center for Machine Learning and Intelligent Systems about Citation Policy Donate a Data Set Contact, 04 February, 2008, 06:24PM.

22. http://www.edrawsoft.com/Data-Flow-Diagrams.php, 05 February, 2008, 11:47AM
23. http://www.smartdraw.com/examples/view/index.aspx?catID=.Examples.SmartDraw.Software_Design.05 February, 2008, 01:05PM

Appendix

This code for showing the relation names and field names in the database:

```
Call ado_Connection(PATHMDB)
Set adoRsfields = adoconnection.OpenSchema(adSchemaColumns)
Do Until adoRsfields.EOF
    If Trim (adoRsfields! TABLE_NAME) = Trim (Tabeldata) Then
       List1.AddItem Trim(adoRsfields!COLUMN_Name)
    End If
    adoRsfields.MoveNext
Loop
```

الملخص

نتيجة للنمو السريع الذي يطرأ على البيانات في مجالات مختلفة ، دعت الحاجة إلى ضرورة وجود أدوات وتقنيات جديدة لتحليل البيانات. هذه الأدوات والتقنيات الجديدة تدعى اكتشاف المعرفة في قواعد البيانات (KDD) اوالتنقيب عن البيانات (Data Mining).

هناك عدة طرق للبيانات الفعالة منها:

- Data Cube-*Based OLAP Rroll-Up Operation*
- *An attribute-oriented induction* (AOI) technique وهي مجال البحث.

وتستخدم هذه الخوارزمية (خوارزمية AOI) لتجميع المعلومات في قاعدة بيانات علائقية عن طريق تكرار عملية استبدال قيم الحقول في قاعدة البيانات بالقيم الأكثر عمومية مستخدمة الشكل الهرمي لذلك الحقل و المعرف من قبل المستخدم وهو ما يسمى بتعميم البيانات (Data generalization) والتي تعنى استخلاص البيانات المهمة ذات العلاقة من قاعدة البيانات وذلك بالانتقال في الشكل الهرمي من المستوى الأقل إلى المستوى الأعلى ، ولهذا فإن الشكل الهرمي للحقل يمثل الركيزة الأساسية لأداء عملية التعميم للبيانات.

إنّ غرضَ هذا البحثِ هو تنفيذ نظام حاسوب لاستخلاص المعلومات بمعالجة أنواع البيانات المعقدة وتجميعها باستعمال تعميم البياناتِ بواسطة المفاهيم الهرمية، هذا النظام تم تنفيذه بواسطة لغة فيجوال بيسك6 مستخدمة أوامر SQL والعديد من المزايا الأخرى، والنظام تم اختباره مع قاعدة بيانات *اكسس* .. وتم أظهار النتائج بأشكال مختلفة.

يوليو 2008\2009

إن الدراسة ليست غاية في حد ذاتها
وانما غاية من خلق الإنسان المتوازن الجميع

التاريخ : .....
الموافق : 2008/12/21 ميلادي
الرقم الإشاري : ع ث س/ك ع/813/1/2009 م

كلية العلوم
قسم الحاسوب
فرع البحث

إستقراء تعميم قيم خصائص البيانات باستخدام المفاهيم الهرمية التدرجية

مقدمة من الطالب

موسى محمد سالم

* * لجنة المناقشة :

1 – د. فرج عبد القادر المؤدب
(مشرفاً) .....

2 – د. زكريا سليمان زوبي
(ممتحناً داخلياً) .....

3 – د. ادريس ساسي الفقيه
(ممتحناً خارجياً) .....

يعتمد

أمين اللجنة الشعبية لكلية العلوم

info@alitahadi.edu.ly 674 054/60361 - 62151 054/60362-65704

استقراء تعميم قيم خصائص البيانات باستخدام المفاهيم الهرمية التدرجية

إعداد الطالب

موسى محمد الرطب

إشراف

د. فرج عبدالقادر المؤدب

قدمت هذه الرسالة لاستكمال متطلبات الحصول على شهادة الدرجة العليا الماجستير

جامعة التحدى - كلية العلوم

قسم الحاسوب

ليبيـــا – سرت 2008\2009