

THE ANALYTICAL MEDIATOR FOR MULTI-DIMENSIONAL DATA

Mazdak HASHEMI, Roger KING, Karen KAFADAR *Member, IEEE*

Abstract—The Analytical Mediator System (AMS) is a fully functional experimental prototype system for importing, cleansing, transforming and integrating, and analyzing heterogeneous data. The AMS provides a straightforward, SQL-like model for managing the entire data management lifecycle. Other critical features of the AMS include: its facility for constructing a domain model suitable for representing the combined semantics of an integrated data set, its easy-to-use facility for reducing the dimensionality of a large data set, its ability to interface with standard data mining tools, its simple GUI, and the easy-to-understand reports it generates.

Index Terms—SQL, Databases, Query, Statistical analysis, Data mining

I. INTRODUCTION

In today's world, a common task facing many organizations requires the use of information from a variety of data sources and repositories that different providers have built over many years. Most companies and organizations collect data over time, whether they are for profit companies or no-profit organizations (e.g. health care, legal, etc). Companies spend considerable financial resources evaluating their data to increase their profit. For example, many companies already store or have gathered large volumes of data in their databases. They use these data sets in an effort to find information leading potentially to new and more successful business tactics. The data can be current or old, and can be stored in various formats; such as relational or non-relational databases, object stores, flat files, knowledge bases, digital libraries, unstructured content stores, electronic mail systems, etc. These sources can reside in local or remote locations, on the intranet or Internet. The user may need not only to read from these sources but to write to

them as well. Accessing accurate information in a timely manner across different organizations is a critical need.

Decision makers such as doctors often need to get the required information as soon as possible. However, inconsistencies in the data, unfamiliarity with the system, and lack of a permanent professional analyst on site at all times, can render such tasks nearly impossible. Examples occur in many domains; for example, a health personnel must decide on a treatment for a patient; a police officer needs to know if a suspect is wanted in another jurisdiction; a social worker must ensure that a welfare applicant is not already receiving benefits elsewhere; a judge needs to see all prior convictions against an offender; an emergency room surgeon requires the medical history of the patient before surgery. Many organizations collect and store information poorly which may not be accessible even within their own systems, much less from other departments outside the organization.

The last few years have seen a spectacular explosion in the quantity of data that are available in electronic formats. Much data are gathered, processed, and stored by many individuals, working for different agencies on varied problems. These massive amounts of collected data are useful only if companies can analyze them. Many data analysis tools have been created, but in reality the most productive way that each organization can benefit from data analysis is by human interpretation of data.

According to an article in *Economist Technology Quarterly* (June 10, 2004), "In the old days, knowing your customers was part and parcel of running a business, a natural consequence of living and working in a community. But for today's big firms, it is much more difficult: a big retailer such as Wal-Mart has no chance of knowing every single one of its customers. So the idea of gathering huge amounts of information and analyzing it to pick out trends indicative of

customers' wants and needs -- data mining -- has long been trumpeted as a way to return to the intimacy of a small-town general store. But for many years, data mining's claims were greatly exaggerated. In recent years, however, improvements in both hardware and software, and the rise of the world wide web, have enabled data mining to start delivering on its promises.". Views on data mining such as this one prompted us to develop methods for facilitating data mining and hence greater value from it.

In general, query languages are used to retrieve information from a database. Often the user knows what types of data he/she needs and uses query languages such as SQL to collect relevant sets of data. This process can easily get very convoluted and complex as the number of tables and relations increases in a database. Considerable expertise may be required to write a SQL query to retrieve data from multiple tables with large record sets. This project was motivated by the need for a more user friendly system. Companies and researchers have tried to improve the speed of this process by doing query optimization and create advanced information retrieval techniques as well as less ad hoc approaches, collectively called knowledge discovery. "The non-trivial extraction of implicit, unknown, and potentially useful information from data" is called knowledge discovery [1]. In some cases they call this approach data mining, the process of discovery of patterns or trends embedded in the data by using extensive statistical analysis. The knowledge discovery process takes the raw results from data mining and carefully and precisely transforms them into logical and useful information. In [2], a clear distinction between data mining and knowledge discovery is drawn.

Companies and organizations use query languages and spend much money and resources on data mining tools to analyze their data. However, without knowledge about the data nor a data expert involved, the results and analyses likely will not mean much to them.

In addition to the problem of data access, many companies also encounter the problem of data quality. Just as one cannot discover the structural damage to a building without examining the foundation, one cannot discover interesting or unusual events in poorly collected (or stored) data. For example; in recent years the assurance of quality care is a core function of public health. Methods to achieve this goal are sorely lacking. Many health care organizations have been using different systems to store their data in a non-centralized architecture, resulting often in many mistakes during medical transcribing and data entry, and possibly inaccurate diagnoses of patient care and incorrect information to perform their analyses and conduct their evaluations over time. While each institution (e.g., clinic, health center, hospital) may have procedures to monitor the quality of care, the institution needs a unifying structure to explore the quality of care embedded in the emerging electronic health record. Efforts such as screening for communicable disease (e.g., HIV[3, 4], tuberculosis[5], and chlamydia[6]); provision of immunization services[7, 8], adherence to cancer screening (e.g., mammography[9]) and guidelines-based measures of quality of care for chronic disease (e.g., diabetes[10, 11]) could be formulated as measurable parameters where continual monitoring could ultimately improve the overall health of the population.

In general, human beings play a vital role in the retrieval of useful information. Without prior stated goals and no plan for use of the final results, the whole process of data mining may be useless.

II. OVERVIEW

Data always involve different levels of uncertainty, resulting in uncertainty in knowledge and information retrieved from databases. SQL and other tools permit data extraction from a database, but cannot quantify users' insufficient knowledge of the tool. Consequently,

they might not extract the desired information, or the tool might be too complicated for them to use. The levels of ability and familiarity with the data and the system, and the skill levels with the domain of application, may be highly variable for the user. These users with Weak knowledge of the system which can lead to syntactic or semantic errors, or even the wrong information altogether to lack of knowledge about the data. Motro [12] categorizes three different levels of uncertainty in database systems (1) Uncertainty about current information in the system (e.g., raw data) (2) Uncertainty about returned information from the database (e.g., analysis) (3) Uncertainty in knowledge of available tools for retrieving and accessing the current information. (e.g., use of tool).

The Analytical Mediator System (AMS) provides a powerful and flexible systematic tool that will reduce confusion and uncertainty for average users of data analysis and reporting. AMS creates a search space by finding and building legal and possible combination of data variables and also allowing users to search, filter and calculate statistics on their data set. Figure 1 provides an overview of AMS architecture (details will be described in S3). AMS acts as a semantic search engine for databases. Web search engines are great and powerful, but they suffer from the problem of not having any knowledge about the domains of interest. This is being investigated in the context of the greater Semantic Web effort [13]. One goal is to incorporate knowledge into the search engine about the specific domain of application beyond simple parsing and key word searches [14].

As mentioned earlier, we can analyze data on a database using a query model approach and data mining tools. Many database applications require more accountability and flexibility in the data analysis process. In many cases, the average user requires assistance from an expert data analyst to retrieve valid and practical information from the database. Often, database users focus

on querying, discovering new patterns, finding statistical trends, etc. However, none of the tools can understand the thought process for each person or user of the system. AMS attempts to identify patterns of potential interest to users and also allows them to search and run analyses on any data sets.

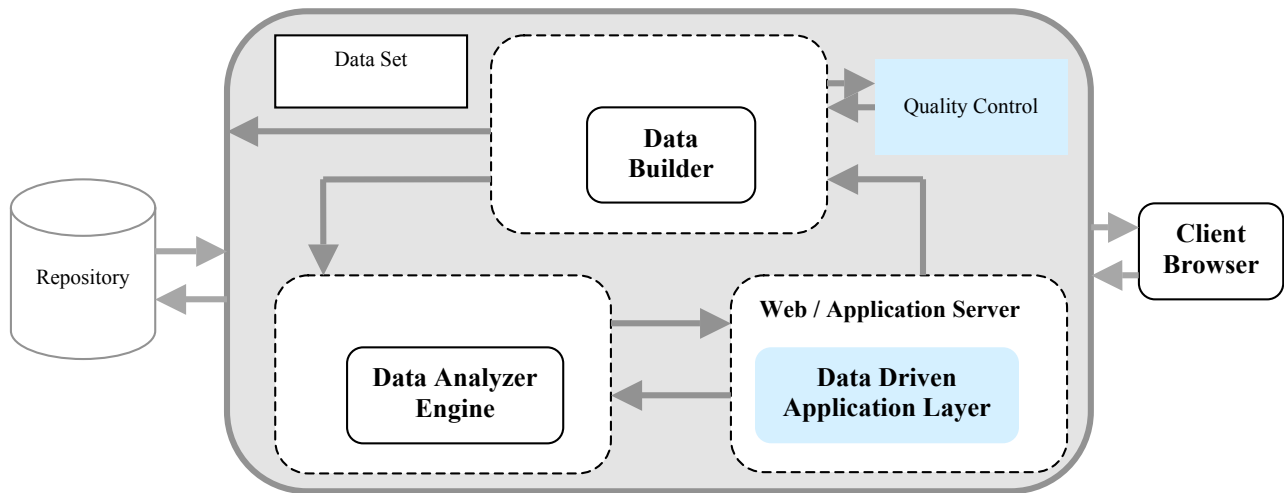


Fig. 1. Overview of AMS architecture

Many companies, organizations and researchers can improve their data analyses by improving the quality of their data. It is also important to have a manageable, flexible and easy to understandable tool for data analysis. Therefore, AMS allows researchers, companies and organizations to verify their data quality and provide the average user with the flexibility to construct reports on top of their databases without having explicit knowledge in a query language or software analysis tool.

Frequently, a user must wait several hours or days for a data expert (analyst) to retrieve needed information; even then, the user and analyst may not understand each other and the results may not be communicated clearly.

In summary, AMS was developed to allow researchers, users and analysts to:

- 1) Create a domain for their data,
- 2) Import data sets,
- 3) Cleanse and check data quality from data sets,
- 4) Utilize semantic data processing,
- 5) Perform data analyses.

The following definitions will be used throughout this article.

AMS: The Analytical mediator system is the core of this tool, which builds the structure, imports, validates and analysis the data sets.

Multi-dimensional data: Data sets that contain multiple attributes on each record.

Options: Each attribute can have multiple values, where each value can also contain multiple values. For example: Attribute Education can have four values: [high-school, undergrad, grad] where undergrad can also have four values: [freshman, sophomore, junior, senior]. (In some fields, “options” are called “levels” of “factors” on “variables”)

III. RELATED WORK

Other approaches similar to AMS have appeared in the literature, but address the problem differently. This section reviews the relevant publications and indicates differences between them and the proposed AMS approach. The initial objective of this review was to simply understand and characterize the fundamental differences among these closely related lines of research and then formulate new ideas which can then be incorporated together to yield a better performing system. Some of this literature falls under the subject headings of “approximate”, “fuzzy”, “incomplete”, “inaccurate”, “probabilistic” and “uncertain data management”. A partial list of articles include references [12, 15-33].

3.1) *Advanced Knowledge Systems*

Several ongoing knowledge based systems projects are underway; some of them target specifically biomedical data; e.g.,

- I. Protégé [34]: an open-source, Java tool that provides an extensible architecture for the creation of customized knowledge-based applications, including modeling and graphical display of data.
- II. Cougaar, or Cognitive Agent Architecture [35]: a DARPA (Defense Advanced Research Projects Agency) project aimed at highly survivable agent systems.
- III. Sesame [36, 37]: an open-source, a web-based architecture to store and subsequently query RDF data and schema information.
- IV. Metalog (REF): a reasoning system built for the Semantic Web by adding a query layer on top of RDF with a simple user friendly graphical interface.
- V. The Data Knowledge Management System (KMS) [38]: a system framework (not a complete solution) designed to support biopharmaceutical research, specifically to organize knowledge about drug targets, lead compounds, and laboratory processes.
- VI. fMRI Data Management Tool [39]: an extension of Protégé, and is aimed at supporting the manipulation of data from experiments.
- VII. The visual multidimensional queries project: [40].

3.2) *Foundations of Mediation - Schema Integration and Evolution, and Multimedia DBs*

Early references for schema integration and evolution research date as early as 1980; the research spiked in the mid to late 1980's, and it continues today [41-43]. This work served in large part to enable later mediation technology. Most of these approaches were based on fairly

low-level, but somewhat formally defined, operations that mapped schemas to common models (often object-like) and resolved terminology conflicts. Other operations are specifically dedicated to evolution, i.e., adding attributes, creating subtypes, etc. Traditional mediation technology focuses largely on relational or table-based data. But the web is a naturally multimedia environment, and there are also well-developed results in the area of multimedia databases [44-47].

3.3) *Database Integration Tools*

The database community has developed other languages, besides Common Object Interconnection Language (COIL), for the specification of database “mediators” [48, 49]. The most well known of these projects, TSIMMIS, was built at Stanford [50, 51] and uses its own object exchange model (OEM) to perform integration. Wrappers map data from underlying sources into the OEM and to map user-queries into source-specific queries. TSIMMIS provides a wrapper specification language to automatically generate its wrappers. The system provides a mediator specification language to generate simple mediators automatically.

Other commercial mediation systems are similar to, but far less powerful than, COIL. One of the most well known is the Cerebellum Portal Integrator [52]. These products provide relational-like operators, along with an iconic drag-and-drop interface that allows users to specify the translation of individual database schemas into a common model, and users specify integration into unified schemas.

3.4) *Ontology and Semantic Web*

As mentioned in Section 1 search engines would be more efficient if they incorporated knowledge about the domain being searched. Solution to this problem has been introduced with

using one of the components of Semantic Web called Ontologies [13]. Ontology uses a formal language to define concepts and relationships for a specific domain or field. The Lowell Database Research Self Assessment proposed that Ontologies could be used to support information integration work [53]. Basically the ontology links a number of concepts (e.g., symptoms to diseases), with the goal of making risk-assessment decisions.

The Semantic Web team has focused on, and conducted substantial research in, the creating languages to support Inference engines, annotation methods, ontologies and several other supporting technologies[54-57]. Many tools have been developed to browse the Semantic Web: for example: SHOE, a basic search tool developed at the University of Maryland [58] which achieves a higher level of accuracy to the search process by using underlying semantic data. Basically, the user chooses an ontology and then a class against which a query should be issued. The system presents a list of properties for that class and then the user selects one or more values, which gets a list of hits that may include URL bindings.

Another approach comes from the University of Karlsruhe called SEAL [59, 60]. This portal basically sits above the inference engine and associated knowledge base and consists of different components, such as a navigation unit which automatically generates links for all related instances. SEALS performs semantic ranking of query results similar to what Automated ranking of database query results [61] attempt to do for SQL databases. They have built a generic automated ranking infrastructure for SQL databases and have tried to incorporate IR methods to rank the query results for databases.

Other researchers [55],[62] created other methods to combine an ontology-based semantic search with a free-text search. In this kind of system, the user will type free text to be searched, and also traverse relevant taxonomies to find instances of certain types directly.

All of the efforts mentioned above, and many others, allow the use of semantics in the browsing and search process. Although these methods bring more powerful searches and browsing tools, they are still very simple and have been used mostly by the Semantic Web Team. They also have several limitations; e.g., they don't allow the user the flexibility to group searches and create smaller domains outside what the authors intended in the first place, nor do they allow analytical components.

Overall, we can see that most of these systems are the start of the Semantic Web and it should establish a good base for a solid research and development in this field.

3.5) *Syndromic surveillance*

One of the more recent developments in the analyses of potential biochemical threats is the conduct of syndromic surveillance[63-65], searching for spatial and temporal patterns of disease within populations. While there are specialized programs for such analyses, often they require sophisticated technical understanding of spatial statistics to use them properly (S+ Spatial Stats, SATSCAN, CLUSTERSEER) [66-68] . Moreover the real challenge is to harness a variety of heterogeneous and novel[69, 70] data sources (e.g., across health care institutions in a jurisdiction[71]). With AMS, public health informaticians [72] can identify possible patterns of disease or the early manifestations of illness (i.e., syndromes).

The main objective for creating AMS is to use, compare, and add to available tools in data mining, machine learning [73-75] and online analytical application tools, and to build a flexible and reliable data analyzer tool for an average user. AMS allows users to convert their multi-dimensional data sets (e.g., data with multiple options for each attribute) to a data set of lower dimension, by considering the possible legal pattern based on the domain set of attributes

and options, and evaluating them for inclusion in a lower dimensional data set. The detail of AMS approach will be covered in section 4.

IV. AMS APPROACH

In this section, we briefly review the AMS approach. The AMS model consists of three building blocks: data quality, data processing and data analysis. Each block is described briefly below. AMS tried to discover the best methods for dealing with each of these components.

4.1) *Data cleaning*

To retrieve more efficient results from any data set, data quality must be checked. Many organizations collect data over the years and combine their results from different sources; in many cases much duplication arises in their data which can lead to inaccurate results. To address this issue, AMS includes a data cleaning process. Data cleansing has been addressed in several articles (e.g., [76-81]) and has many components, such as “deduplication”, a key operation in integrating data from multiple sources [82]. Deduplication is handled using merge/purge algorithms to prevent creating duplicated records especially in the data integration process. Deduplication algorithms find and merge multiple data items that are likely to represent the same real-world entity. Different deduplication algorithms have been published, (as in, for example, the machine learning literature [83-85]). Rather than creating a new algorithm to prevent deduplication in data cleaning process, our goal is through a quantitative and comparative research to combine the best applicable approach that is available for implementation in AMS.

4.2) *Data Quality Parser*

Scientists and researchers in many disciplines such as social sciences pool data from multiple resources for their research purposes. In many cases the origin of the data and the creation of the data are unknown. In many current database applications, engineers and database

designers have focused more on schema design; most of the traditional database systems are very weak when it comes to dealing with checking data quality. The data quality component of AMS is a user-interface that allows the user to run a quality check and create a tag for each record in the database. The tag will show when the record was created and who uploaded/updated/modified the data. The data quality agent is divided into two different categories: 1) New data 2) Existing data.

The user of a local database might not always know the quality of the data that is being accessed. If data are imported from another source or exported to another source it will have a different quality based on the user needs and application domains. For instance, the manager of a wholesale branch might be more concerned about the quality of the cost measurements, whereas the payroll department may be more interested in the quality of the demographic data.

The Data Quality Component of AMS also addresses the deduplication mentioned above and produces “cleaner” data for the analysis part of this application.

4.3) *Query Analyzer*

The goal of the query analyzer is to manipulate heterogeneous data, and then make use of those data through queries and computations, and doing it all in a way that is both flexible and manageable which basically is the goal of AMS.

In general purpose data mining applications; in particular, integrating arbitrarily diverse data and creating general-purpose agents that can perform precisely specified tasks, is not currently feasible. At present, most data warehousing environments provide only a small handful of online analytical processing (OLAP) operations. OLAP provides approximate answers to the user while more exact answers are still being computed [86]. These operations can compute only

very simple aggregate values, those typically useful in traditional data processing environments, such as banking and insurance – where data are naturally flat and contain only simple implied semantics.

4.4) *AMS approach to Data Analysis*

Researchers and organizations use different data mining tools and techniques to analyze data that have been stored in databases, possibly over many years. The tools and features can be customized based on the domain and usage of each application. In many cases the users need extensive training to understand the tool before using it. This can easily become a challenging task for both users and administrators of such tools. Section 1 of this paper briefly reviewed some of the problems and challenges involved in the current data mining tools and solutions.

The goal of this project was not to create another data mining tool, but use current research and technology to create a novel instrument that can interface with any organized data set and reduce the dimensionality of the data to enrich and simplify the data analysis. The preliminary plan was to build a tool that will interface with a relational database or data set with similar structure. The user can customize his/her domain via a graphical interface. The domain is basically a combination of one or multiple data sets that the user constructs for his/her analysis. Each domain can contain multiple attributes where each attribute might have several options as shown in Figure 2.

$$\{ \exists \text{ attribute } A, \forall \text{ option } O \mid [O_1, O_2, O_3, \dots, O_n] \subset A \}$$

Fig. 2. Definition of attributes and options

The number of options is used to find the range for each attribute. Based on Figure 2 each attribute can have multiple options. In many database applications the dimension of an attribute could change during the life of the project (e.g., adding, deleting and modifying options in the

database). Therefore, regular updates are necessary to ensure accuracy (e.g. the range should display the current number of options).

The AMS interface allows the user to define attributes, dimensions and construct a domain (section 5 briefly described the architecture of this component). This interface labels parameters and preferences for each user. After the parameters have been set, the user can import the data set to AMS data builder.

For our experiment, the data sets contain only numerical data that arrived from organizational and medical data sets. The AMS data builder creates a legal hierarchy structure (relationships and dependencies between attributes and options) and generates valid paths based on conditional probabilities for each data set. It also computes simple computational statistics, such as mean, median, standard-deviation, range, for each variable in data set that the user will upload. In addition, since many analysts like to use their own formulas for statistical measurement, this tool allows users to define those formulas and build them into the AMS library.

The AMS converts multilevel-dimensional data sets to one dimension “behind the scenes” and generates combinational sets based on schema, rules and hierarchy structure of the data. For future development, our plan is to implement and experiment with several classification algorithms, such as Naïve Bayes classifiers, decision trees, neural networks, and support vector machines (SVMs), to determine which algorithm performs best on building and discovering different legal combinations for AMS data sets [73, 74]. Based on performance results, advanced users can select different algorithms for their data processing.

In summary, we developed a mediator that communicates with multiple data sets or selected databases. The mediator allows users to construct their own domain of interest by

selecting different attributes and options. It gives users the flexibility to insert and run pre-built statistical formulas and functions and search for undiscovered paths by generating all possible number of combinations for each attribute and its options.

V. DETAILED EXAMPLE

To better understand the principles described in section 4.4, we show a detailed example for the domain of organizational development. Consider an organization hierarchy that contains six (6) levels:

Levels [CEO, VP, DIRECTOR, MANAGER, DEPARTMENT, EMPLOYEE]

The data set contains multiple annual and semi-annual surveys that measure employees' job satisfaction and other measurements criteria. An organization's strategy is best seen in the pattern of moves and approaches devised by leadership to produce successful organization performance. A key factor affecting organizational performance is the ability of the leadership team to chart the organization's long-term direction, and execute the strategy that leads to the intended results. Therefore, many organizations try to gather data on how well people, systems and culture perform strategic tasks. AMS was used to measure strategic results in large departments within an organization and was able to find new patterns to achieve better organization success rate.

The concept hierarchy for this dataset is shown in Figure 3 below, which displays the number of possible options (dimensions) for each attribute.

We first construct a conceptual search space for a director D_1 that oversees seven (7) different managers "M" with the following Criteria:

- I) Hierarchy Structure (Graphical view is shown in Figure 4)

$$\left\{ \exists \text{ director } D_1: [M_1, M_2, M_3, M_4, M_5, M_6, M_7] \mid \forall \text{ manager } M: [D_1, D_2, D_3, D_4, D_5, D_6] \text{ and } \exists \text{ dept } D: [E_1, E_2, E_3, E_4, E_5] \right\}$$

II) Set of attributes and options to build a data set for each department

EMP(5), ScaleID(1), Gender(M, F, All),
 Ethnicity(W, AA, H, NH, All), Shift(PT, FT, Temp),
 Education(HS, Undergraduate(F,S, J,S), College,
 Graduate(M.S., M.E., M.A., M.D., PhD)

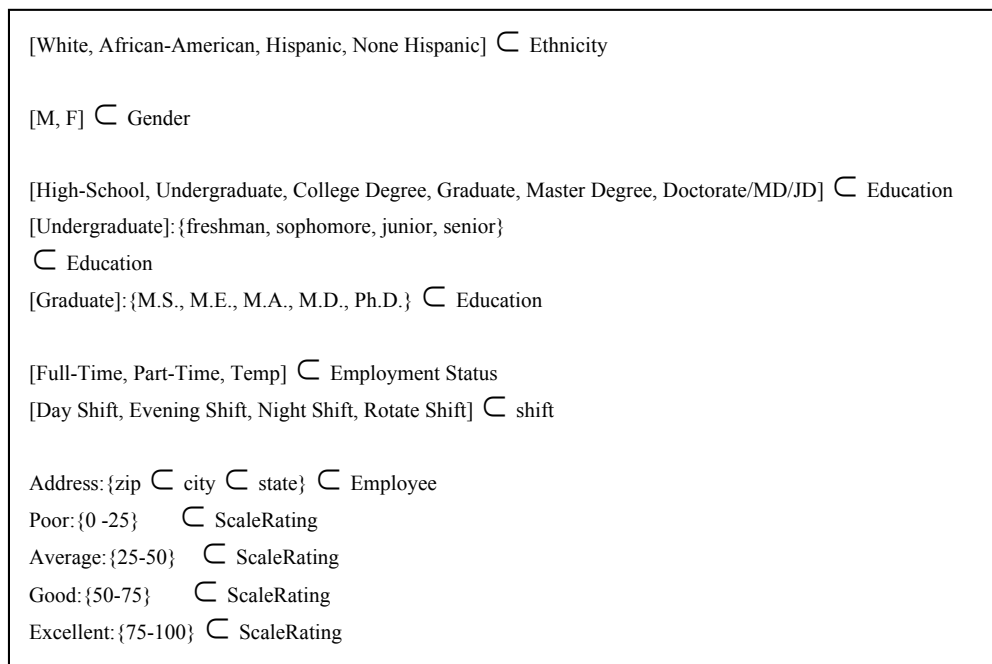


Fig. 3. Concept hierarchy table

Based on above example the search space contains 218 people and 103,950 possible legal combinations with two different hierarchical layers, where each combination is a logical path from director to the set of attributes and options that are shown in Figure 4.

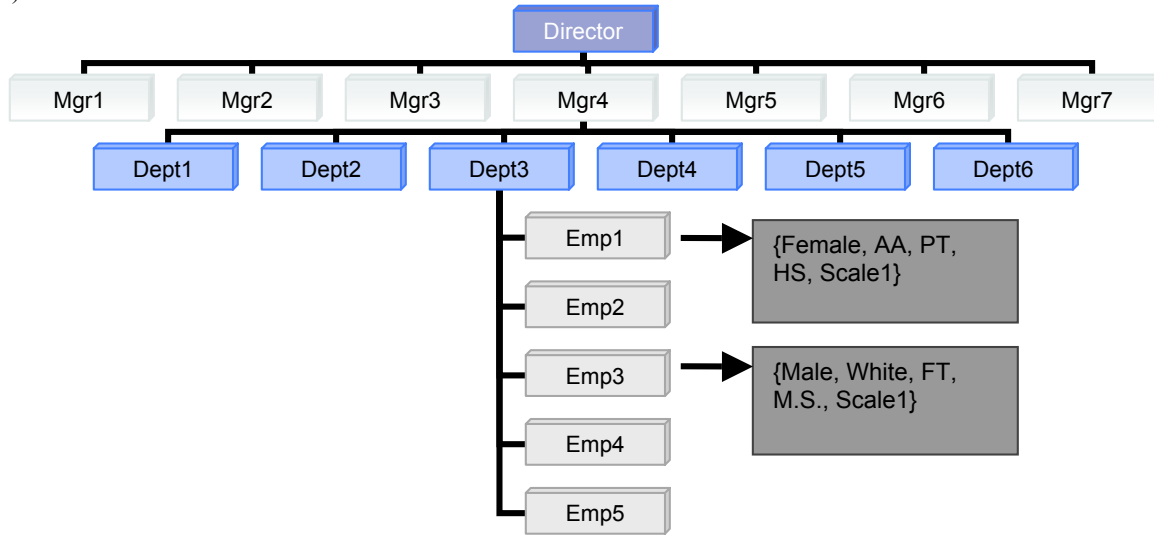


Fig. 4. Concept hierarchy table

One sees immediately the infeasibility of writing a query or using any existing features in current OLAP and data mining tools to discover different possible combinations of the data sets for an organization such as above example.

In reality, the AMS produces the possible legal combinations automatically similar to a crawler agent that finds and builds each combination and also applies statistical formulas and calculations based on user preference. The legal path for each combination is stored in a new table along with any numerical results derived from the calculations. A conceptual view of that table is shown in Figure 5 below (AMS_CONTROL).

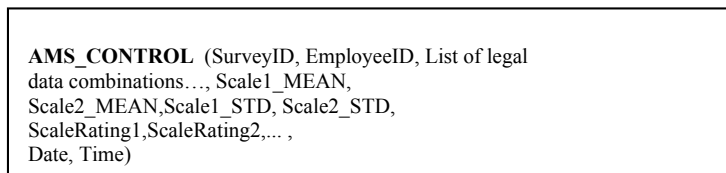


Fig. 5. Conceptual view of AMS table

As mentioned above, AMS attempts to convert multi-dimensional data to fewer dimensions, from a user perspective AMS creates a new table that contains all the information that is needed for searching a domain and running statistical reports. This table “AMS_Control” contains the general calculation such as mean, standard-deviation and others that have been defined by the user.

Consider in some detail a short scenario of illustrating how a user can interact with AMS. AMS requires each user to register and create a username and password. The user can then log in to the system via a web interface. After the user was authenticated, the system displays a menu for the user. Figure 6 is an example of an AMS menu.

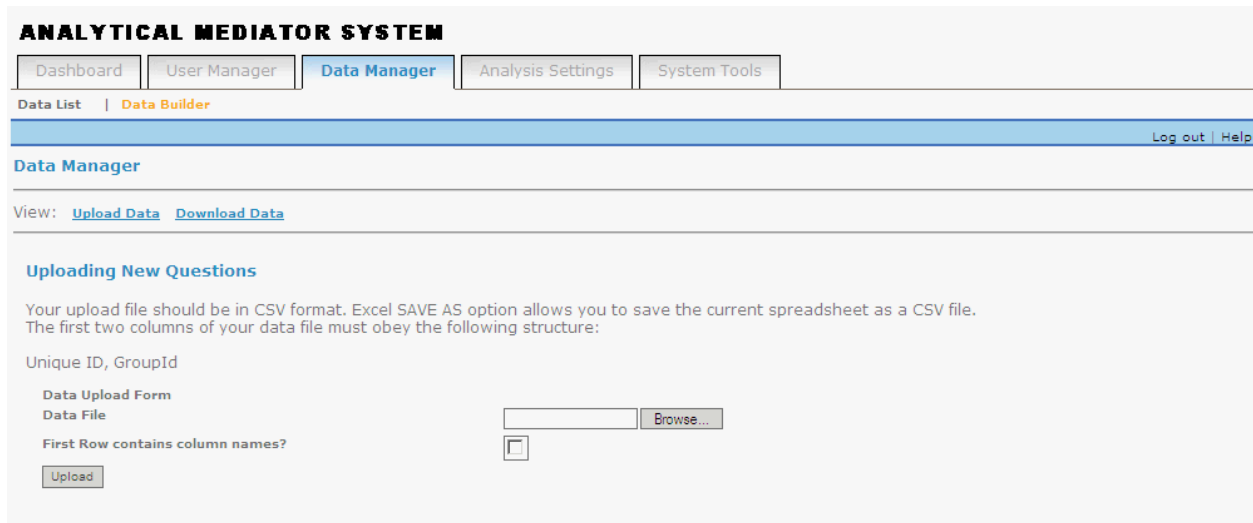


Fig. 6 AMS Menu (Data Manager)

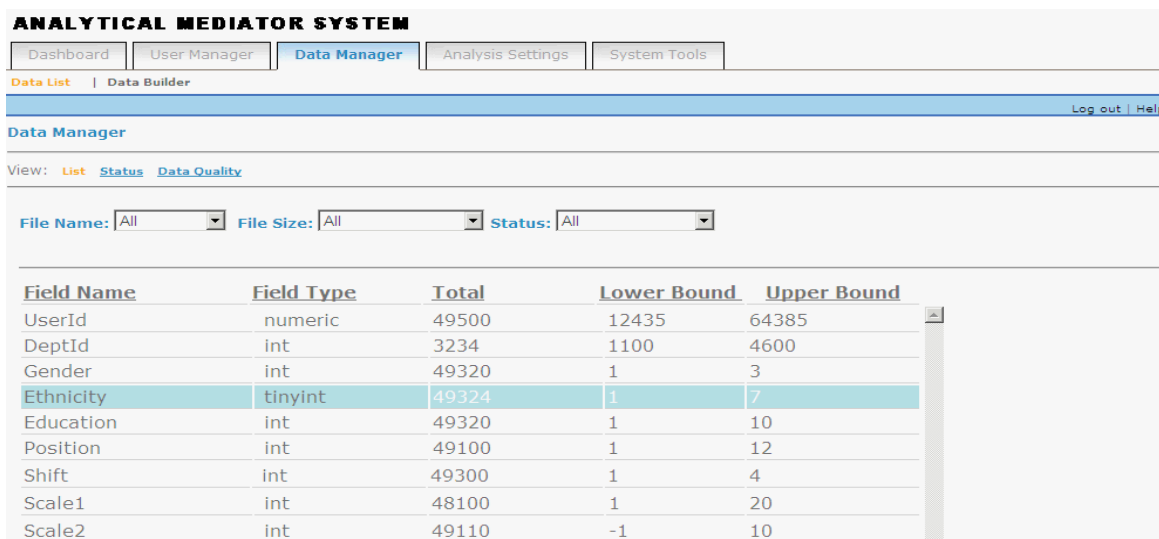
The first step is to upload some data. The user can upload any file that is in a comma separated version (csv) format. The user can either upload the file with headers or leave them blank and AMS will use its default headers. Figure 7 shows you an example of sample upload file which we will use through out this paper.

UserId	DeptId	Gender	Education	Position	Ethnicity	Status	Shift	Tenure	Teams	SurveyId
1111	2	M	3	2	1	2	3	1	2	2
MeasurementScale1		MeasurementScale2						MeasurementScale10	
2.4		5		7.6				7.3	

Fig. 7 AMS sample upload file

By clicking on the upload link under data manager section of AMS as shown in Figure6, AMS displays an upload format for the user. The user is required to follow the format or the upload

process might fail. For instance, the first field on each data needs to be a unique ID (in this case UserId). The 2nd field is used for creating hierarchy structure; the user can either leave this field blank or have numbers that represent its structure (in this example DeptId). The other fields require no order; they will be used to find combinations and set criteria for the search. The user can view the uploaded information by clicking on “Data List” as shown in Figure 8. This page will display schema information for the uploaded data in AMS.



The screenshot shows the 'ANALYTICAL MEDIATOR SYSTEM' interface. At the top, there are navigation tabs: 'Dashboard', 'User Manager', 'Data Manager' (selected), 'Analysis Settings', and 'System Tools'. Below the tabs, there are links for 'Data List' and 'Data Builder'. The main content area is titled 'Data Manager' and includes a 'View:' section with options for 'List', 'Status', and 'Data Quality'. Below this, there are three dropdown menus for 'File Name', 'File Size', and 'Status', all set to 'All'. The main part of the interface is a table with the following data:

Field Name	Field Type	Total	Lower Bound	Upper Bound
UserId	numeric	49500	12435	64385
DeptId	int	3234	1100	4600
Gender	int	49320	1	3
Ethnicity	tinyint	49324	1	7
Education	int	49320	1	10
Position	int	49100	1	12
Shift	int	49300	1	4
Scale1	int	48100	1	20
Scale2	int	49110	-1	10

Fig. 8 AMS Data List

This schema is used for setting up the analysis section of AMS and to define the search space for the uploaded data. The user can define the search space by going to the analysis settings of AMS. Under this menu the user can view current profiles or create a new profile. The default profile is set to all the potential paths that have been discovered for the upload data file after the processing time. Figure 9 shows an example of the screen that use can setup his/her profile. The “create profile” runs a process on the backend of AMS system (similar to display data list section) and defines the following information for each numerical field:

- 1) Lower bound
- 2) Upper bound
- 3) Data type ? Integer or Double

The rest of the fields are treated as char. AMS also displays how many unique values each field contains. For example:

UserId int(11) (1500)
Gender Char(1) (1457)
MeasurmentScale1 int(11) (9) lower: 0 upper:9
MeasurmentScale2 int(11) (20) lower: 1 upper:21

The create profile interface allows user to define:

- 1) Search Criteria
- 2) Combinations sets

Presently, AMS allows only numerical fields for the search criteria. The user can either select all the possible fields for the search or select a smaller set. For each new data set that is uploaded to AMS, the user should expect some processing time to create the table that includes all the possible combinations. After that table is built the user can view or filter any possible groups that exist in his/her data.

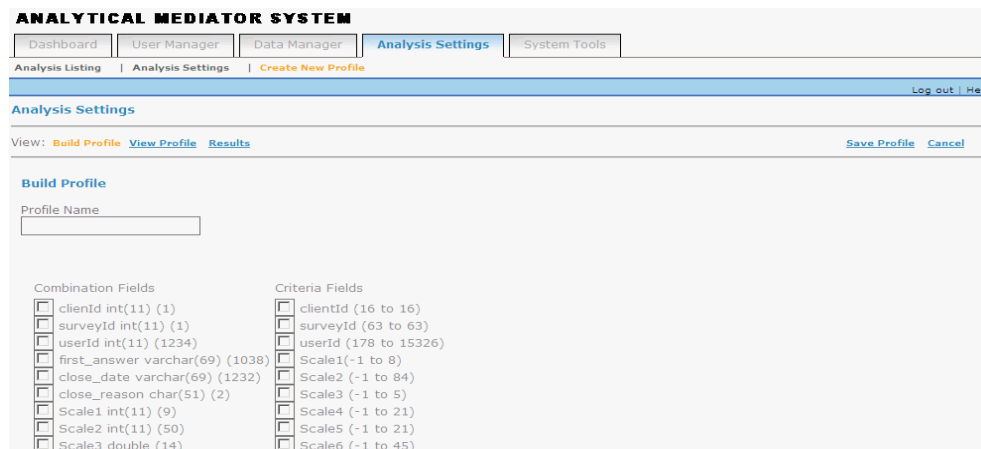


Fig. 9 Create profile menu

At a high level, the AMS backend data processing system generates dynamic queries based on all possible fields and criteria in AMS data analyzing table and writes all available data

combinations to the AMS_Control table as shown above. Figure 10 show a brief over of AMS algorithm for data crunching. (xx mazdak – the font below is too small.)

- 1) Generate and store all possible criteria
- 2) Generate and store all possible combination groups
- 3) Construct queries to retrieve all data that match the criteria (candidate)
- 4) Get all data for any valid values in the criteria (Space)
- 5) Filter combination groups from candidates
- 6) Build combination sets based on the number of options in each field
- 7) Search through the combinations list for valid match
- 8) If the valid match found store it, else remove that path (pruning)

Fig. 10. Overview of AMS backend algorithm

At this time we are trying to develop different ways to reduce the processing time, because, as the dataset size increases, the number of possible combinations increases and so does the processing time. Our front end interface is written in PHP and the backend data processing is written in C++.

Figure 11 shows sample results from the example above. In this case, the user was looking for a combination of 7 fields where each group had more than 2 members. As an example, figure 3, “Gender” can have three different values (M, F, Unknown) and, “Ethnicity” can have 6 values (African American, Asian, Caucasian, Hispanic, Native American / Alaska Native, Unknown), and so forth.

For this scenario, the total number of possible groups was 14817600. AMS discovered 49 different paths where the group size was greater than 2.

It took less than 2 minutes and it was run on a Pentium® processor 1500 MHz 1.5 GHz with 513 MB of RAM. The time dropped with a more powerful processor.

The screenshot shows the ANALYTICAL MEDIATOR SYSTEM interface. At the top, there are navigation tabs: Dashboard, User Manager, Data Manager, Analysis Settings (selected), and System Tools. Below the tabs, there are links for Analysis Listing, Analysis Settings, and Create New Profile. The main content area is titled "Analysis Settings" and includes a "View" section with links for Build Profile, View Profile, and Results. The "Results for sample-profile 1" section contains a table with two columns: "Combination Group" and "Group size".

Combination Group	Group size
Education (M.S.), Employment Status (UNKNOWN)	3
Employment Status (UNKNOWN) , Shift (FT-D)	4
Gender (F), Education (PhD/MD), Shift (FT-Flex)	5
Position (Nurse), Gender (F), Ethnicity (Hispanic)	3
Dept. 111 - Employment Status (PT-T)	3
Dept. 111 - Shift (Pr)	3
Dept. 222 - Employment Status (PT-T)	3
Dept. 333 - Education (B.S.)	3
Dept. 444 - Education (B.S)	4
Dept. 444 - Education (college), Employment Status (PT-Benefits)	3
Dept. 444 - Education (college), Shift (PT-D)	3

Fig. 11. Partial results

VI. AMS ARCHITECTURE

The high-level of AMS is shown in Figure 1 above and the brief description of each component is described below.

6.1) User Interface

The goal was to make the user interface intuitive and flexible enough so the user can explore and use it fairly easy. This User Interface(UI) was developed as a standalone client, written in PHP so the user can easily access the tool from anywhere and use the system. As shown in Section 5, the user can upload the formatted data set using this interface to the Data Builder. The UI also allows the system admin to assign different security levels to different users.

The link between a user's category and set of possible actions and features was also constructed in the tool. The similar interface with different features, which are based on user's category was developed to analyze the data and allow user to run different reports and use different knowledge based component.

6.2) *Data Builder*

This component uses a database and data processing/analyzing algorithms that have been implemented. The Data builder will search, calculate and construct possible sets of data paths and patterns based on different factors such as user's category, number of attributes, set of options for each attribute and the domain that user is working.

6.3) *Repository*

The Repository is the core of AMS system. It stores all of the semantically related data, in addition to providing data retrieval facilities, serving query requests, and performing computations.

6.4) *Data Analyzer Engine*

This component is responsible for query classification and query execution of the system. It basically provides numerical answers according to the query and its classification. A query can be classified to different categories based on the query condition and the information that is being asked. For instance, queries on organization development can be categorized based on director level and managerial level of organization according to the conditions given in the query. It also can calculate the approximate size of a category, the % match between two or more categories, the statistically relevant attributes of a category or intersection of categories, and so on. Also, a set of heuristic rules will be specified behind the scenes based on user category, query category, the chosen domain and the relationships between attributes and conditions.

6.5) *Data Analyzer Interface*

The Data Analyzer (DA) Interface gives the users the ability to build their own queries. It also allows them to use available set of queries (query templates) based on the user category. Essentially, the user creates a query using this interface and upon execution of the query, the query object is forwarded from the interface to the DA Engine on the Repository. Next, the DA Engine makes a number of important decisions based on the contents of the query. Lastly, the DA Engine assembles the raw data into a result object, which is returned to the user.

VII. CONCLUSIONS AND FUTURE WORK

Query languages such as SQL have been used to get information out of a database. The major limitation for naïve users is uncertainty about the kind of information that is needed from the database and the unfamiliarity with the grammar of such query languages. The goal of AMS was to create a structure to allow users to access data and find potential legal groups from the database. We are also trying to build a query builder to allow users transform their sentences through its parser (machine learning algorithm) and, based on different set of rules and constraints (i.e. privacy, security rules), applicable create a legal and logical approximate query based on those features. Approximation methods have been used in query processing aspect of database systems. Where one of the method is to precalculate values in the database, then process queries using sampling or other statistical techniques to produce approximate answers, e.g., [87-89].

VIII. REFERENCES

1. Frawley, W.J., Piatetsky-Shapiro, G., and Matheus, C. Knowledge Discovery In Databases: An Overview. In Knowledge Discovery In Databases, eds. G. Piatetsky-Shapiro, and W. J. Frawley, AAAI Press/MIT Press, Cambridge, MA., 1991, pp. 1-30. .
2. U.M. Fayyad, G.P.-S., P. Smyth, and R. Uthurusamy, AAAI Press/The MIT Press, Menlo Park, CA., 1996, pp. 1-34. .
3. Goggin, M., et al., *The extent of undiagnosed HIV infection among emergency department patients: results of a blinded seroprevalence survey and a pilot HIV testing program.* J Emergency Medicine., 2000. **19**: p. 13-19.
4. Trepka, M.J., A.J. Davidson, and J.M. Douglas, *HIV seroprevalence and undiagnosed infection in an acute-care hospital: implications for assessing the extent of the epidemic and for routine screening.* Am J Prev Med, 1996. **12**: p. 195-202.
5. Steele, A., et al. *Encoded guidelines for targeted latent tuberculosis screening using an electronic medical record.* in Annual AMIA Symposium. 2003. Washington, DC.
6. Alfonsi, G., et al. *Chlamydia testing in a public healthcare system before and after implementation of screening guidelines.* in National STD Prevention Conference. 2002. San Diego, CA.
7. Davidson, A.J., et al., *Immunization registry accuracy: improvement with progressive clinical application.* Am J Prev Med, 2003. **24**: p. 276-80.
8. Hambidge, S.J., et al., *Strategies to improve Immunization rates and well-child care in a disadvantaged population: a cluster randomized trial.* Arch Pediatr Adolesc Med, 2004. **158**: p. 162-169.
9. Hedegaard, H., A.J. Davidson, and R.A. Wright, *Factors associated with screening mammography in low-income women.* Am J Prev Med, 1996. **12**: p. 51-56.
10. Turner, S., J. Westfall, and A. Davidson. *Receipt of diabetic-related preventive services in an urban health care system: comparison by race/ethnicity.* in North American Primary Care Research Group. 2003. Banff, Alberta.
11. Lasater, L.M., et al., *Glycemic control in English- vs. Spanish-speaking Latinos with type 2 diabetes mellitus.* Arch Intern Med, 2001. **161**: p. 77-82.
12. A. Motro. Management of uncertainty in database systems. In W. Kim, e., Modern Database Systems: The Object Model, Interoperability, and Beyond, pages 457–476. ACM Press, New York, 1994. .
13. Berners-Lee, T., J. Hendler, and O. Lassila, "The Semantic Web," in Scientific American, May 2001, 2001.
14. Hendler, J., *Is There an Intelligent Agent in Your Future?* Nature March 1999.
15. E. Zim'anyi. Query evaluation in probabilistic relational databases. Theoretical Computer Science, January 1997.
16. C. Olston and J. Widom. Offering a precision-performance tradeoff for aggregation queries over replicated data. In Proc. of the 26th Intl. Conference on Very Large Data Bases.
17. N. Fuhr. A probabilistic framework for vague queries and imprecise information in databases. In Proc. of the 16th Intl. Conference on Very Large Databases, p., Brisbane, Australia, August 1990.
18. N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In Proc. of the 30th Intl. Conference on Very Large Databases, T., Canada, August 2004.
19. D. Barbar'a, H.G.-M., and D. Porter. The management of probabilistic data. IEEE Trans. On Knowledge and Data Engineering, 4(5):487–502, October 1992.
20. B. Buckles and F. Petry. A fuzzy model for relational databases. International Journal of Fuzzy Sets and Systems, 1982.
21. R.S. Barga and C. Pu. Accessing imprecise data: An approach based on intervals. IEEE Data Engineering Bulletin, June 1993.
22. R. Cheng, D.V.K., and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In, p. Proc. of the 2003 ACM SIGMOD Intl. Conference on Management of Data, San Diego,, and J. California.
23. R. Fagin. Combining fuzzy information: An overview. ACM SIGMOD Record, June 2002.
24. T. Imielinski and W. Lipski. Incomplete information in relational databases. Journal of the ACM, October 1984.
25. S.K. Kwan, F.O., and D. Rotem. Uncertain, incomplete, and inconsistent data in scientific and, e. statistical databases. In A. Motro and P. Smets, Uncertainty Management in Information Systems:, and B. From Needs to Solution. Kluwer Academic Publishers, 1996.
26. S.K. Lee. An extended relational database model for uncertain and imprecise information. In Proc. of the 18th Intl. Conference on Very Large Databases, p., Vancouver, Canada, August 1992.
27. L.V.S. Lakshmanan, N.L., R. Ross, and V.S. Subrahmanian. ProbView: A flexible probabilistic database system. ACM Transactions on Database Systems, 22(3):419–469, September 1997.
28. I. Lazaridis and S. Mehrotra. Approximate selection queries over imprecise data. In Proc. of the 20th Intl. Conference on Data Engineering, p., Boston, Massachusetts, March 2004.
29. K.-C. Liu and R. Sunderraman. Indefinite and maybe information in relational databases. ACM Transactions on Database Systems, March 1990.
30. A. Ola and G. Ozsoyoglu. Incomplete relational database models based on intervals. IEEE Trans. on Knowledge and Data Engineering, April 1993.
31. F. Sadri. Modeling uncertainty in databases. In Proc. of the 7th Intl. Conference on Data Engineering, p., Kobe, Japan, April 1991.
32. M. Shapcott S. McClean, B.S.A.o.i.a.u.i.i.d.I.T.o.K.a.D.E., 13(6):902–912, November 2001.
33. Q. Yang, C.L., J. Wu, C. Yu, S. Dao, H. Nakajima, and N. Rische. Efficient processing of nested fuzzy SQL queries in fuzzy databases. In Proc. of the 11th Intl. Conference on Data Engineering, pages 131– 138, Taipei, Taiwan, March 1995.
34. <http://protege.stanford.edu/>. *Protégé-2000*. 2004 [cited; Available from: <http://protege.stanford.edu/>].
35. <http://www.cougaar.org/>, *Cognitive Agent Architecture*. 2004.
36. Broekstra, J., A. Kampman, and F.v. Harmelen, *Sesame: An Architecture for Storing and Querying RDF Data and Schema Information*, in *Spinning the Semantic Web*, D. Fensel, et al., Editors. 2003, MIT Press: Cambridge, MA. p. 197-222.
37. <http://www.openrdf.org/>, *RDF*.
38. <http://www.3rdmill.com/initiatives/downloadsKMS.htm>, *Data Centric Knowledge Management System*. 2004.
39. <http://fmridc.org/>, *fMRI Data Management Tool*. 2004.
40. Böhnlein, M., M. Plaha, and A.U.-v. Ende, *Visual Specification of Multidimensional Queries based on a Semantic Data Model*.
41. Ahmedi, L. *Directory-Based Ontologies for Integrating XML Data.* in *Workshop on Foundations of Models and Languages for Data and Objects*. 2001.
42. Eder, L., *Managing Healthcare Information Systems with Web-Enabled Technologies*. 2000: Idea Group Publishing.

43. Staab, S., et al. *An Extensible Approach for Modeling Ontologies in RDF(S)*. in *Proceedings of ECDL 2000 Workshop on the Semantic Web*. 2000.
44. Adiba, M. and N. Quang, *Historical Multi-Media Databases*. 12th International Conference on Very Large Databases, 1986.
45. Al-Khatib, W., et al., *Semantic Modeling and Knowledge Representation in Multimedia Databases*. IEEE Transactions on Knowledge and Data Engineering, 1999. 11(1).
46. Hoschka, P., *Synchronized Multimedia Integration Language*. 1998, W3C Proposed Recommendation.
47. Liu, L., C. Pu, and Y. Lee. *An Adaptive approach to query mediation across heterogeneous databases*. in *Proceedings of the Int. Conf. on Cooperative Information Systems*. 1997.
48. Barja, M., et al., *A Mediator for Integrated Access to Heterogeneous Information Sources*. CIKM, 1998.
49. Wiederhold, G., *Mediators in the architecture of future information systems*. IEEE Computer Vol. 25 No. 3, 1992.
50. Garcia-Molina, H., et al., *The TSIMMIS approach to mediation: Data models and Languages*. Journal of Intelligent Information Systems, 1997.
51. Chawathe, S.S., et al., *The TSIMMIS Project: Integration of Heterogeneous Information Sources*. IPSJ, 1994.
52. <http://www.cerebellumsoft.com>, *Cerebellum Portal Integrator*. 2003.
53. Serge Abiteboul, R.A., Phil Bernstein, Mike Carey, Stefano Ceri, Bruce Croft, David DeWitt, Mike Franklin, Hector Garcia Molina, Dieter Gawlick, Jim Gray, Laura Haas, Alon Halevy, Joe Hellerstein, Yannis Ioannidis, Martin Kersten, Michael Pazzani, Mike Lesk, David Maier, Jeff Naughton, Hans Schek, Timos Sellis, Avi Silberschatz, Mike Stonebraker, Rick Snodgrass, Jeff Ullman, Gerhard Weikum, Jennifer Widom, and Stan Zdonik. *The Lowell Database Research Self-Assessment Meeting*. May 2003
54. Daconta, M.C., L.J. Orbrst, and K.T. Smith, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. 2003: John Wiley & Sons. 312.
55. Davies, J., U. Krohn, and R. Weeks. QuizRDF: search technology for the semantic web. in WWW2002 workshop on real world RDF & Semantic Web Applications, 11th International WWW Conference. 2002. Hawaii, USA.
56. Fensel, D., J. Hendler, H. Lieberman, and W. Wahlster, eds. *Spinning the Semantic Web*. 2003, The MIT Press: Cambridge, Massachusetts.
57. Thuraisingham, B., *XML Databases and the Semantic Web*. 2002: CRC Press. 336.
58. Heflin, J. and J. Hendler. *Searching the Web with SHOE*. in *Artificial Intelligence for Web Search. Papers from the AAAI Workshop*. 2000. Menlo Park, CA: AAAI/MIT Press.
59. Maedche, A., S. Staab, Stojanovic, R. Studer, and Y. Sure. SEAL - A framework for developing SEMantic portALs. in *Proceedings of the 18th British National Conference on Databases*. 2001. Oxford, UK: Springer-Verlag.
60. Staab, S., J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure, *Semantic community Web portals*. Computer Networks, 2000. 33(1-6): p. 473-491.
61. S. Agrawal, S.C., G. Das, and A. Gionis. Automated ranking of database query results. In *Proc. of the First Biennial Conference on Innovative Data Systems Research (CIDR '03)*, Pacific Grove, California, January 2003.
62. McGuinness, D., H. Manning, L.A. Resnick, and T.W. Beattie, *FindUR: Knowledge-enhanced online search*. 1998. Available at: <http://www.research.att.com/~dlm/papers/findur-chi98.ps>.
63. Lober, W.B., et al., *Roundtable on Bioterrorism Detection: Information System-based Surveillance*. J Am Med Inform Assoc, 2002. 9: p. 105-15.
64. Lazarus, R., et al., *Using automated medical records for rapid identification of illness syndromes (syndromic surveillance): the example of lower respiratory infection*. BMC Public Health, 2001. 1(1): p. 9.
65. Lazarus, R., et al., *Use of automated ambulatory-care encounter records for detection of acute illness clusters, including potential bioterrorism events*. Emerg Infect Dis, 2002. 8(8): p. 753-60.
66. <http://www.insightful.com/products/splus/>, *S-PLUS*.
67. <http://www.satscan.org/>, *SATSCAN*.
68. <http://www.terraseer.com/products/clusterseer.html>, *CLUSTERSEER*.
69. McClung, M.W., et al. *Using respiratory-related calls to a nurse advice line to predict pediatric upper respiratory infection-related healthcare utilization*. in *2003 Annual AMIA Symposium*. 2003. Washington, DC.
70. McClung, M., et al. *Evaluating data sources for syndromic surveillance*. in *American Public Health Association Meetings*. 2001. Atlanta, GA.
71. Wagner, M.M., et al., *The emerging science of very early detection of disease outbreaks*. J Public Health Manag Pract, 2001. 7(6): p. 51-9.
72. Lober, W.B., et al. *Communicable disease case entry using PDAs and public wireless networks*. in *2003 Annual AMIA Symposium*. 2003. Washington, DC.
73. Russell, S.J.a.P.N., *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. 2003: Prentice Hall.
74. Witten, I.H.a.E.F., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. 2000, San Francisco, California: Morgan Kaufmann Publishers.
75. <http://www.autonlab.org/tutorials>, S.D.M.T., Andrew Moore.
76. A.E. Monge. An efficient domain-independent algorithm for detecting approximately duplicate database records. In *Workshop on Research Issues on DataMining and KnowledgeDiscovery (DMKD'97)*, T., Arizona, May 1997.
77. S. Chaudhuri, K.G., V. Ganti, , and R. Motwani. Robust and efficient fuzzy match for online data cleaning. In *Proc. of the 2003 ACM SIGMOD Intl. Conference on Management of Data*, pages 313–324, San Diego, California, June 2003.
78. H. Galhardas, D.F., D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. In *Proc. of the 27th Intl. Conference on Very Large Data Bases*, pages 371–380, Rome, Italy, September 2001.
79. M.A. Hernandez and S.J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*.
80. V. Raman and J.M. Hellerstein. Potter's wheel: An interactive data cleaning system. In *Proc. of the 27th Intl. Conference on Very Large Data Bases*, p.
81. S. Sarawagi, e.S.I.o.D.C., *IEEE Data Engineering Bulletin* 23(4), December 2000.
82. S. Sarawagi and A. Bhamidipaty. Interactive deduplication using active learning. In *Proc. of the 8th ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining*, p.
83. W. E. Winkler. Matching and record linkage. In B. G. C. et al, e., *Business Survey Methods*, pages 355-384. New York: J. Wiley,

1995. available from <http://www.census.gov/>.
84. W. Cohen and J. Richman. *Learning to match and cluster entity names*. In ACM SIGIR' 01 Workshop on Mathematical/Formal Methods in Information Retrieval.
 85. W. E. Winkler. The state of record linkage and current research problems. RR99/04, h.w.c.g.s.p.p.r.-p., 1999.
 86. P.J. Haas and J.M. Hellerstein. Online query processing. In Proc. of the 2001 ACM SIGMOD Intl. Conference on Management of Data, S.B., California, May 2001.
 87. S. Acharya, P.B.G., V. Poosala, and S. Ramaswamy. The Aqua approximate query answering system. In Proc. of the 2001 ACM SIGMOD Intl. Conference on Management of Data, pages 574–576, Philadelphia, Pennsylvania, June 1999.
 88. D. Barbar'a, W.D., C. Faloutsos, P.J. Haas, J.M. Hellerstein, Y.E. Ioannidis, H.V. Jagadish, T. Johnson, R.T. Ng, V. Poosala, K.A. Ross, and K.C. Sevcik. The New Jersey data reduction report. IEEE Data Engineering Bulletin, 20(4):3–45, December 1997.
 89. M.N. Garofalakis and P.B. Gibbons. Approximate query processing: Taming the terabytes. In Proc. of the 27th Intl. Conference on Very Large Data Bases, R., Italy, September 2001.