# Technical Report CS-00-02
# Automated Identification of Errors in Data Sets*

Jonathan I. Maletic
Andrian Marcus

The Department of Mathematical Sciences Division of Computer Science
The University of Memphis
Campus Box 526429
Memphis, TN 38152

jmaletic@memphis.edu, amarcus@memphis.edu

**Abstract:** The paper presents an overview of the current research and methods applied to the problem of data cleansing. It presents a tool for automated data cleansing of data sets. The tool is designed to be domain independent and constitutes the first part in a proposed framework for automated data cleansing. Development of a tool to address the entire framework is the ultimate goal of the research. The paper addresses the data cleansing problem from a different perspective, that is identification of errors in single data sets. This research ties together data quality and data mining issues. Existing outlier detection methods are utilized: clustering, pattern identification, and statistical methods. Real-world data is used for experiments. The methods and results are presented and analyzed. Refinements of these methods and new methods are proposed to address the data cleansing problem.

* This research is supported in part by a grant from the Office of Naval Research.

## 1. Introduction

The quality, correctness, consistency, completeness, and reliability of any large real world data set depend on a number of factors [Wang95], [Wang96], [Eish99]. But the source of the data is often times the crucial factor. Data entry and acquisition is inherently prone to errors both simple and complex. Much effort is typically given to this front-end process, with respect to reduction in entry error, but the fact often remains that errors in a large data set are common. Unless an organization takes extreme measures in an effort to avoid data errors the field errors rates are typically around 5% [Redman98], [Orr98]. Where the error rate is equal to the number of error fields over the number of total fields. There is a wide range of impacts to the organization including higher operational costs, poorer decision-making, increased organizational mistrust, and diversion of management attention.

The logical solution to this problem is to attempt to cleanse the data in some way. That is, explore the data set for possible problems and endeavor to correct the errors. Of course for any real world data set, doing this task "by hand" is completely out of the questions given the amount of person hours involved. Some organizations spend millions of dollars per year to detect data errors [Redman96]. A manual process of data cleansing is also laborious, time consuming and, prone to errors. The automation of the data cleansing process for large sets of data may be the only practical and cost effective way to achieve a reasonable quality level in a data set. While this may seem to be an obvious solution, little research has been directly aimed at this problem. Related research addresses the issues of data quality [Ballou89], [Redman98], [Redman96],

[Wang95] and tools to assist in "by hand" data cleansing (e.g., [EDD99], [ETI99], [Pak93], [QMSoft99], [Trillium99], [Vality99], [Wang93]).

### 1.1. Data Cleansing

Data cleansing is a relatively new field. The process is computationally very expensive, thus it is almost impossible to do with old technology. The new fast computers allow performing the data cleansing process in acceptable time on large amounts of data. There are many issues in the data cleansing area that researchers are attempting to tackle. They consist of dealing with missing data, determining record usability, erroneous data etc. Different approaches address different issues. Of interest to this research is the search context for what is called in the literature and the business world as “dirty data” ([Hernandez97], [Kimball96], [Fox94], [Flanagan98], [English99]).

There is no commonly agreed definition of the data cleansing. Various definitions depend on the particular area in which the process is applied. Three major areas include the data cleansing as part of their defining processes: data warehousing, knowledge discovery in databases (KDD), and total data quality management (TDQM).

### 1.2. Related Work

Within the data warehousing field, data cleansing is applied especially when several databases are merged. Records referring to the same entity are represented in different formats in the different data sets or are represented erroneously. Thus, duplicate records will appear in the merged database. The issue is to identify and eliminate these duplicates. The problem is known as the *merge/purge problem* [Hernandez97], [Galhardas99], [Moss98]. Instances of this problem appearing in the literature are called record linkage, semantic integration, instance identification, or object identity problem.

From this perspective data cleansing is defined in several (but similar) ways. In [Galhardas99] data cleansing is the process of eliminating the errors and the inconsistencies in data, and solving the object identity problem. The [Hernandez97] paper defines the data-cleansing problem as the merge/purge problem and proposes the basic sorted-neighborhood method to solve it. The proposed method is the basis for the DataBlade module of the DataCleanser tool [EDD99].

Data cleansing is much more than simply updating a record with good data. Serious data cleaning involves decomposing and reassembling the data. According to [Kimball96] one can break down the cleansing into six steps: elementizing, standardizing, verifying, matching, house holding, and documenting. Although data cleansing can take many forms, the current marketplace and the current technology for data cleaning are heavily focused on customer lists [Kimball96]. In this area, three companies dominate the data-cleaning marketplace [Kimball96], and all three specialize in cleaning large customer address lists. The three companies are Harte-

Hanks Data Technologies [Trillium99], Innovative Systems Inc. [Innovative99], and Vality Technology [Vality99]. Recently, companies have started to produce tools and offer data cleaning services that do not address specifically the customer address lists but do rely on domain specific information provided by the customer: Centrus Merge/Purge Module [QMSoft99], DataCleanser [EDD99], etc. A very good description and design of a framework for assisted data cleansing within the merge/purge problem is available in [Galhardas99].

TDQM is an area of interest both within the researchers and the business community. The data quality issue and its integration in the entire information business process are tackled from various points of view in the literature (e.g., [Fox95], [Fox94], [Levitin95], [Orr98], [Pak93], [Redman96], [Redman98], [Strong97], [Svanks84], [Wang96]). Other papers refer to the same problem as the enterprise data quality management [Flanagan98]. Probably the most comprehensive survey of the research in the area is available in [Wang95].

Unfortunately, none of the mentioned papers refer explicitly to the data cleansing problem. Some of the papers deal strictly with the process management issues from data quality perspective, other with definition of data quality. The later category is of interest to this research. In the proposed model of data life cycles with application to quality [Levitin95] the data acquisition and data usage cycles contain a series of activities of: assessment, analysis, adjustment, and discarding of data. Although it is not specified in the paper, if one integrated the data cleansing process with the data life cycles, this series of steps would define it in the proposed model from the data quality perspective. In the same framework of the data quality, [Fox94] proposes four quality dimensions of the data: accuracy, current ness, completeness, and

consistency. The correctness of data is defined in terms of these dimensions. Again, a simplistic attempt to define data cleansing process within the framework would be as the process that assesses the correctness of data and improve its quality.

More recently, data cleansing is regarded as a first step or preprocessing step in the KDD process [Fayyad96], [Brachman96]. No precise definition and perspective over the data cleansing process is given. Various KDD and Data Mining systems perform data cleansing activities in a very domain specific fashion. In [Guyon96] discovering of informative patterns is used to perform one kind of data cleansing by discovering *garbage patterns* – meaningless or mislabeled patterns. Machine learning techniques are used to apply the data cleansing process in the written characters classification problem. In [Simoudis95] data cleansing is defined as the process that implements computerized methods of examining databases, detecting missing and incorrect data, and correcting errors. The Recon Data Mining system is used to assist the human expert to identify a series of errors types in financial data systems.

## 2. Problem Definition

This research is concerned with the following generic framework for automated data cleansing:

- Define and determine errors types
- Search and identify error instances
- Correct the uncovered errors

This paper is focused on the first two aspects of the generic framework for automated data cleansing. The data set currently under investigation is comprised of real world data supplied by the Navy Personnel Research and Development Center (NPRDC). The data set is part of the officer personnel information system including midshipmen and officer candidates. Similar data sets are in use at personnel records division in companies all over the world. The investigated data and its use represent a large problem space.

This particular data set is updated constantly through a large number of different mechanisms (other data bases, data entry and other personnel, etc.). The entry (and origin) of the data stretches over a long period of time and there are few rigid processes in place (i.e., TDQM) for quality assurance. The lack of quality metrics and measurements on the data set determine and unknown overall quality of the data set. Therefore, the possibility for errors in the data is high. Assessing the overall quality of such a data set, and correcting mistakes is of high priority.

The following are some of the characteristics of the data set and problem at hand:

- Exploitation of the data set and the data cleansing are performed off-line. Therefore, computation time is not a primary concern, but the data cleansing process should be scalable to larger data sets.
- The size of the data set is medium (45,000 records with 311 fields). The average size of a field is 8 bytes. However, it is large enough to prohibit transformations of the entire data set in the main memory of a usual computer.
- More than half of the elements in the data set are empty.

- Over 50% of the fields are date type, representing events. The rest of the fields represent domain specific attributes, which would vary among different problems.

For the experiments presented here, a part of the original data set, referred from now on as the *data set* is utilized. Also, only the date type fields are being examined by the methods presented. Data type data (data of date type representing events in time) fits very well the proposed goal of keeping the data cleansing process as domain independent as possible. All the date type data elements have more or less the same format, and the range is rather data driven than domain driven. The only issue that needs some domain knowledge is the fact that the first field in the data set now should contain the earliest date for each record. This however does not constrain the problem too much, since most data sets of this type contain a field corresponding to the date of birth, which is the earliest date that appears in an individual personnel record. This field is referred further in the paper as the *reference date*. If there is no such a field in the data set, an extra field can be added to the data set, which would satisfy this criterion.

A further step yields even more domain independency. All the date type values are converted to integer numbers. Since dates have different representations in different application, the set of tools was designed to recognize and convert twenty-eight types of date representations, grouped in five classes. The only restriction is that the data elements in the same data set should be represented in the formats from the same class. An integer type is utilized for memory concerns. Thus, the data cleansing tool set operates on any data set that has date type or integer type elements, with the only restriction that the first field of each record contains the smallest value in that record. Further research will prove if the set of tools could be used on non-data type data

sets. The current experiments were done only on the date type data set. Date type data allows the establishment of some hypothesis on data, as shall be seen further in the paper, that makes the definition and identification of possible errors easier, without using domain knowledge.

## 3. Error Types and Methods Utilized

The implemented data cleansing tool focuses primarily on outlier detection [Knorr97], [Barnett94]. As mentioned before, most of the existing tools and research is concentrated around the merge/purge problem, where the outlier detection is not a concern. More than that, all of these methods and most of the existing data mining methods consider outliers of no interest and to be removed. Almost all studies that consider outlier identification as their primary objective are in statistics [Knorr97]. The only other way in which outlier detection is currently used in data cleansing is purely by using data visualizing tools to actually see the outliers [Simoudis95]. That process is entirely manual, that is no automation in the outlier identification is done. The stated goal is to automate as much as possible this process. The implemented data cleansing tool is one of the first to automate a large part of the outlier detection process used in data cleansing. The resulting files, containing the outlier fields or records are much smaller than the original data set, so are much easier to be analyzed manually by the user.

The three major approaches considered are:

1. Identifying outlier fields using statistical values (averages, standard deviation, range), based on Chebyshev's theorem [Barnett94], [Bock98], considering the confidence intervals for each field;
2. Identify outlier records that do not conform to existing pattern in the data. As mentioned above, a different approach in the character recognition field is discussed in [Guyon96]. Combined techniques (partitioning and classification) are used to identify patterns that apply to most records.
3. Identify outlier records using clustering based on Euclidian distance. Existing clustering algorithms provide little support for identifying outliers [Korn96], [Murtagh83], [Zhang]. A combined clustering method is utilized along with the group-average clustering algorithm [Yang99] considering the Euclidean distance between records. Two clusters are considered similar and merged together only in case they are each other's reciprocal nearest neighbors (RNN) [Murtagh83].

The architecture of the tool, the expected and the obtained results are presented further. Some of the methods did not produce the expected results. The cases are analyzed and solutions for fine-tuning the tool are proposed.

## 4. A Tool for Data Cleansing.

A general architecture for the tool and the framework at this stage is presented in Figure 1. A set of five independent subsystems was used in a pipe-filter architecture to search and identify the possible errors in a data set. C++ was chosen as implementation language. The system is such implemented to assure certain portability between platforms (e.g., Windows, Unix).

The implemented tool is designed to identified the following potential errors:

1. Non-numerical value in a field
2. A value that represents a date earlier than the reference date (the data element in the first field)
3. Missing value in the reference field
4. Too many empty fields
5. Outlier values in a record
6. Records that do not follow existing patterns in the data
7. Outlier records according to clustering

Analyzing the data and preparation of the data are still manual processes. A reusable data structure was created and implemented as a C++ class (*adate class*). The purpose of the data structure is to handle date type data in a standardized manner. It accepts as default date format year-month-day. It can work also with the other four date types and various date ranges. Altogether, the structure deals with twenty-eight date formats. The main feature of the data structures is that it converts dates to integers representing number of days passed between a

reference date and the given date. This type of conversion is used for each date in the data set, considering the first field as the reference date. Thus, each field is converted to an integer representing the number of days passed between the date in the field and the reference date, which is the first field.

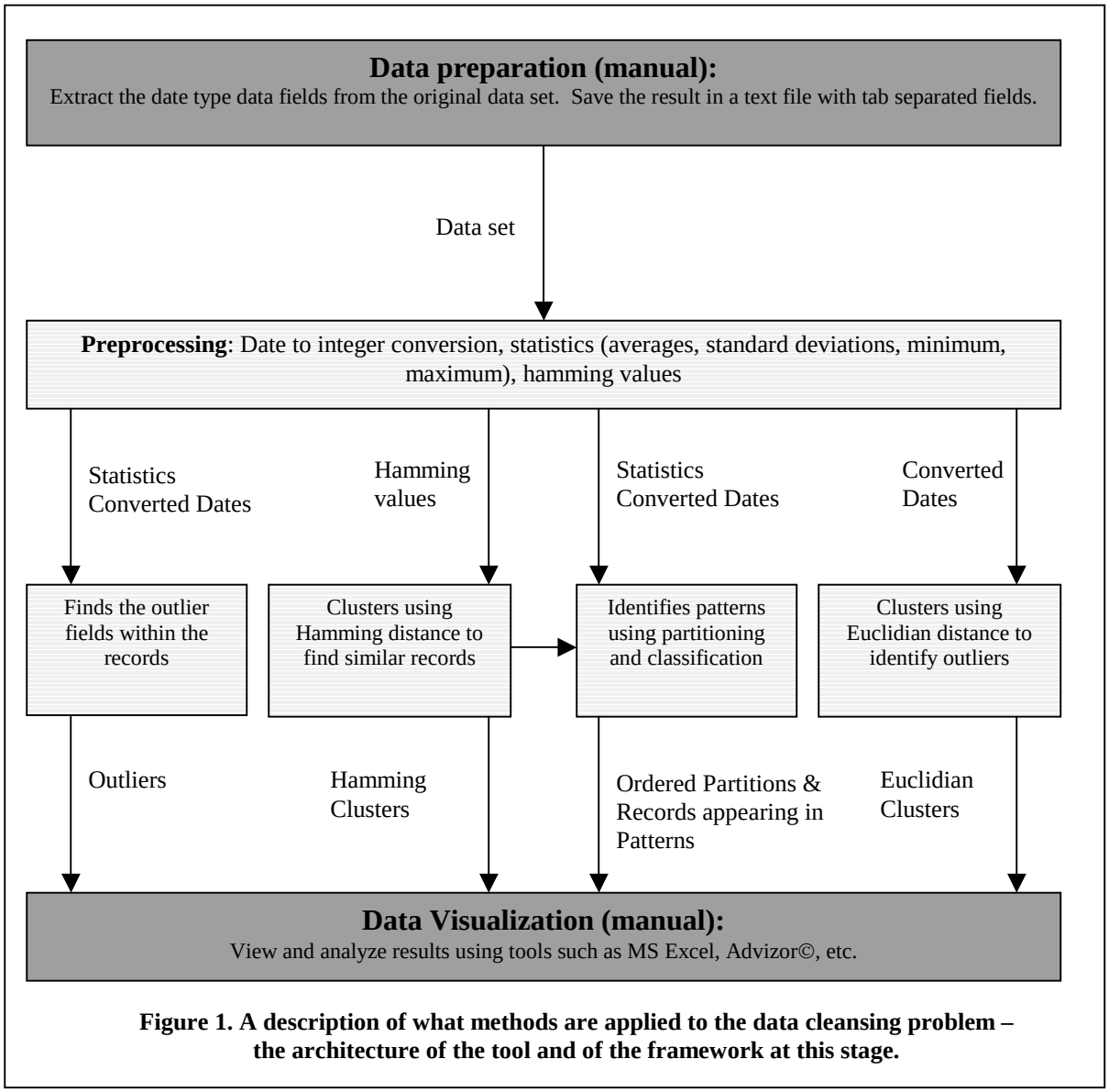


**Figure 1. A description of what methods are applied to the data cleansing problem – the architecture of the tool and of the framework at this stage.**

### 4.1. Preprocessing

For those dates that did not have a specified the month or the day, it is assigned 1 for each (January for the month and 1st for the day). The empty fields get a specific value, indicated by a constant (EMPTY). The fields with non-numeric values get a value specified by another constant (WRONG). Each field that contained the number 0 was considered empty. However, since this represents an inconsistency in the representation of the data it is mentioned as a possible error.

First preprocessing is performed to address the following tasks:

- Convert the date values into integers and output these values,
- Compute statistical values and output these values,
- Assign to each value a string of bytes representing the empty (0) and non-empty (1) fields in the record and output these values.

Each output is formatted in a tabular manner so that can be imported easily in any spreadsheet program or data visualizing tool. Only a single pass over the data set is required. The methods implemented in the adate class allow all the necessary conversions to be performed.

### 4.2. Identifying Outlier Values

Missing and outlier values for particular fields are identified based on the statistics and other information computed in the preprocessing. For each field the average and the standard deviation are utilized and based on Chebyshev's theorem [Barnett94] those records that have values in a given field outside a number of standard deviations from the mean are identified. The number of standard deviations to be considered is customizable. A range of five standard deviations in the tests is used. Confidence intervals are taken into consideration for each field. Also "wrong" values (non-numerical or negative) are uncovered.

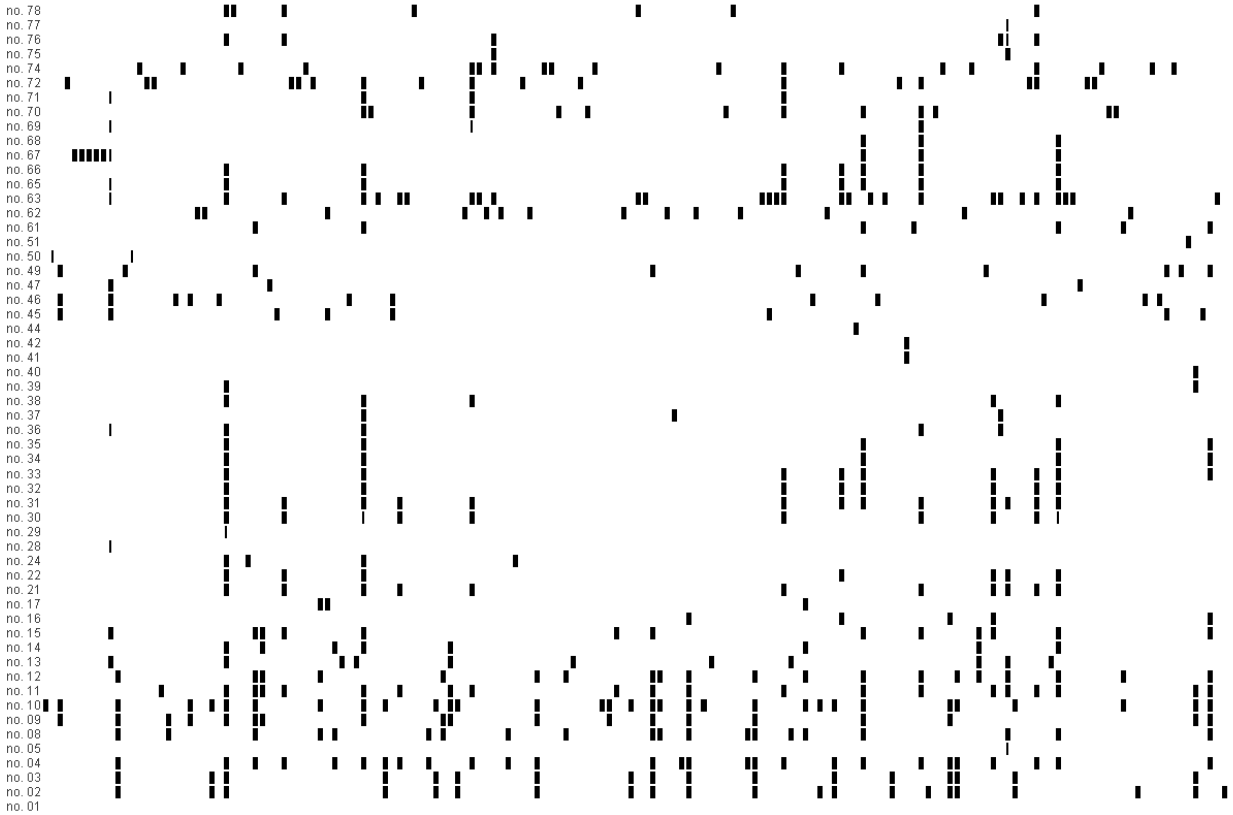


**Figure 2. Outliers and wrong values detected in the data set. X-axis: record number, Y-axis: field number. The thicker line in the figure indicates an outlier, and the thinner line indicates a "wrong" value. 164 records with such values are represented.**

Among the 5000 records of the experimental data set, 164 contain outlier or wrong values. Figure 2 gives a graphical display of this data set produced by a data visualization tool ADVIZOR by Visual Insight Inc. The thicker line in the figure indicates an outlier, and the thinner line indicates a "wrong" value. The Y-axis contains the field numbers in which the error is found, and the X-axis the record number. The visualization tool allows the user to move the mouse pointer to any of the lines and it indicates the corresponding record and field number, as well as the contained value. Trying to visualize the entire data set to identify the outliers by hand would be impossible. If the user attempts this field-by-field could take quite some time. The data cleansing tool, in two cases, points out missing values: if the first field (reference date) is missing, or if only the first field is present and all the others in a record are empty.

### 4.3. Identification of Patterns

Clustering the record space according to number and position of the empty fields is done in an attempt to group records that might contain patterns. Here, strings of zeros and ones, referred to as *Hamming value* [Hamming80], are associated with each record. Each string has as many elements as the number of fields. A "1" in the string represents a non-empty field on the same position in the record as the 1 in the string. A "0" in the string represents an empty field on the same position in the record as the 0 in the string. Patterns in a set of records that have the same empty fields are identified. In order to do this, the data set has to be partitioned in subsets of records with the same number of non-empty fields on the same positions. The Hamming

distance [Hamming80] is utilized to cluster the records into groups of close records. Initially, clusters having zero Hamming distance between records are identified. Unfortunately, the number of identified clusters was too high (4631 clusters for 5000 records). The largest cluster had only 98 records and the second largest only 29. This step required a single pass over the output from the preprocessor. Since the results were not encouraging, a hierarchical clustering method will be implemented to determine clusters of records with a Hamming distance larger than zero. The result showed that although fields in each record represent the same event, real-life data often is very diverse.

Patterns are identified in the data according to the distribution of the records per each field. A pattern is defined by a large group of records (over 90% of the entire data set) that have values for each field in the same ranked partition. Initially the tool was supposed to use subsets of the data set containing the clusters with zero hamming distance, generated before. However, due to the small size of these clusters it was decided to apply the tool on the entire data set. In the future, the tool will use the clusters generated by the new clustering algorithm, mentioned above. The guiding element in this approach is the fact that only date type data is being examined, and the same field corresponds to the same event in each field.

In order to determine the existence of a pattern, the distribution of records according to each field is analyzed. The record set is partitioned in subsets surrounding the mean (the number of subsets used is six, one being used for empty fields). The diameters of the partitions depend on the standard deviation for each field. Each partition is classified according to the number of records it contains (i.e., partition number 1 has the largest size and so on). Then the following

hypothesis is considered: if there is a pattern that is applicable to most of the fields in the records, then a record following that pattern should be part of the partition with the same rank for each field. In other words, in a 2D representation of the records, there should be a shape that includes the values of the records respecting the pattern. Figure 3 contains such a representation for the first 10 fields in the data set. Only the values for 2500 records are shown. It is currently impossible to visually represent the entire data set in one screen. If a pattern exists, (a compact shape is identified) it would be defined by the shape's height.

This method was applied on the entire data set and a small number of records (0.3% of total number of records) were identified that followed the pattern for more than 90% of the fields.

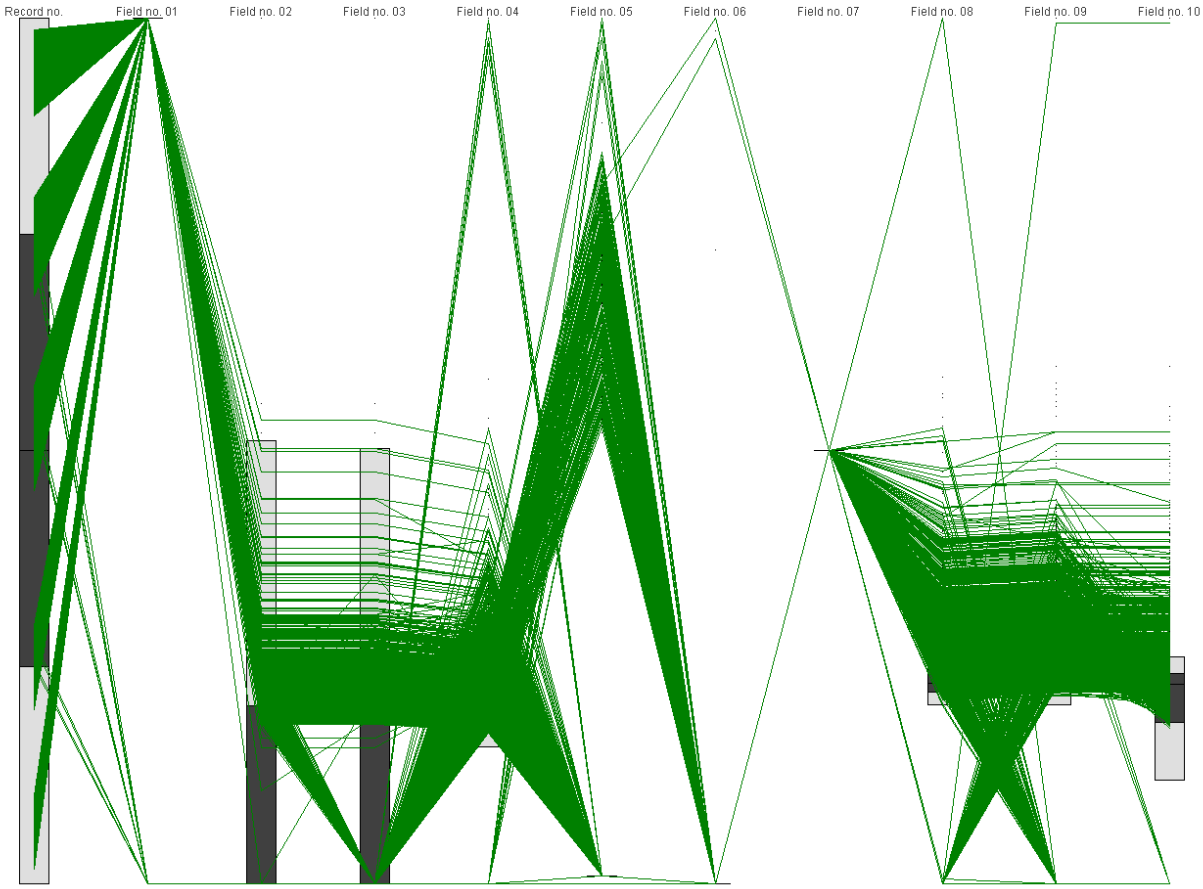


**Figure 3. Representation of the first 10 fields of 2500 selected records from the data set. The lack of a compact shape proves the absence of a pattern. Too many outlier values are observed.**

If a representation, such as the one in Figure 3, is hard to interpret due to the large dimensionality of data, one can study the distribution of the values for one field at a time. Figure 4 shows the distribution of the records for the first 10 fields of the data set. The same records as in Figure 3 are highlighted and show that the values in fields 2 and 3 disrupt an eventual pattern that would apply to all the fields.

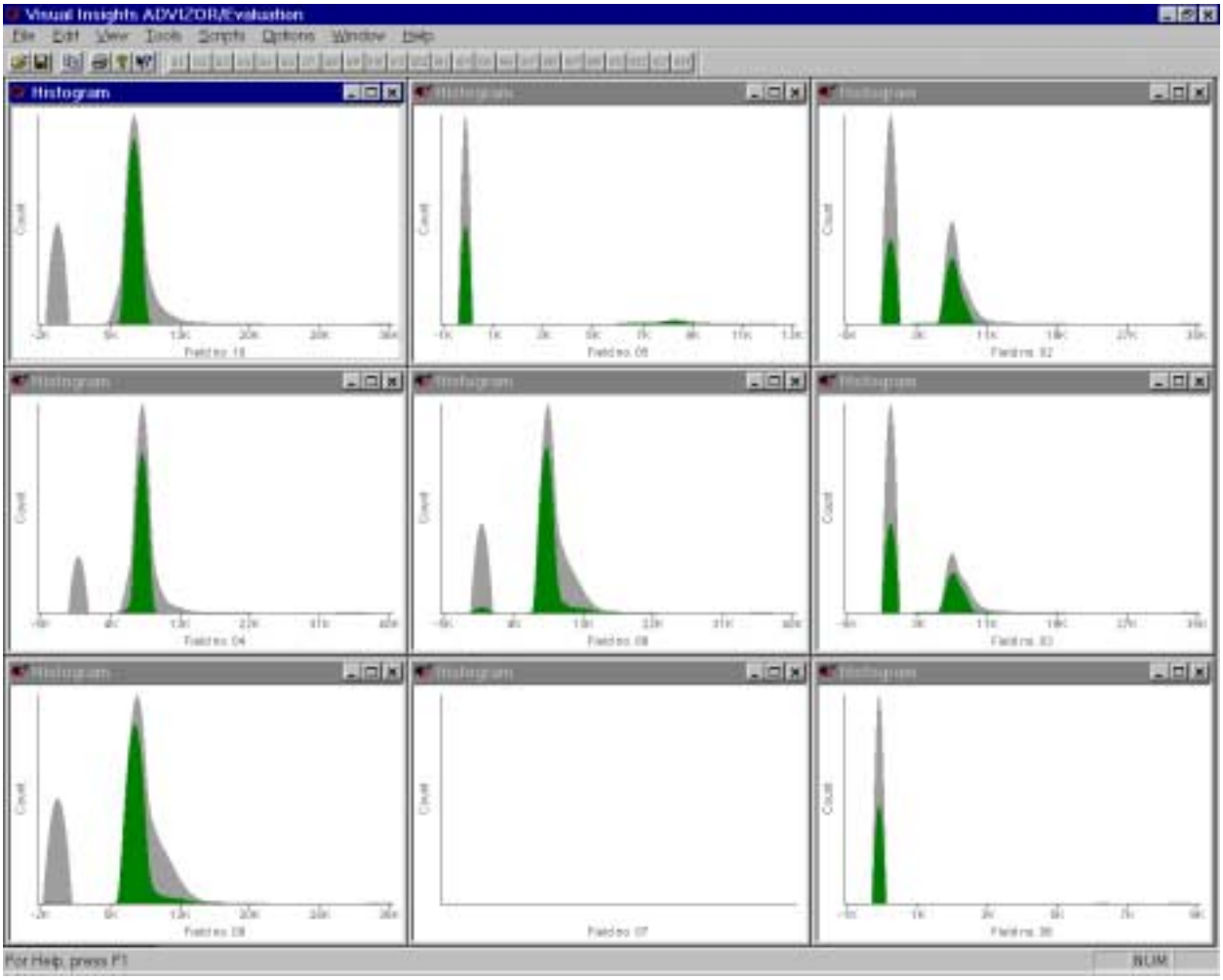


**Figure 4 Distribution of the record for fields 2-10. The same 2500 records as in Figure 3 are highlighted. The values in fields 2 and 3 (most empty) disrupt the pattern – records are not part of the same ranked partitions.**

Trying to "see" an existing pattern is impossible due to the size of data. Even if one uses only few fields and a part of the records this step cannot be done by hand. The data visualization tool

only helps to confirm or disconfirm the achieved result. As mentioned, no significant pattern was identified, during these experiments, which apply to each field. The tool will be adapted and applied on cluster of records generated using the Hamming distance, rather than on the entire data set. Chances of identifying a pattern will increase since records in clusters will already have certain similarity and have approximately the same fields empty. Also, in the future two different approaches are envisioned for this tool: use a clustering method instead of a user driven partitioning and identify patterns in subspaces. Thus, the similarity measure would depend more on the data. Again, real-life data proved to be highly non-uniform.

### 4.4. Clustering and Outlier Detection

The tool also implements a group-average-based hierarchical clustering method on the entire data set. The clustering algorithm was run several times adjusting the maximum size of the clusters. Ultimately, the goal was to identify as outliers at least those records that were identified before as contain outlier values. However, computational time prohibits multiple runs in an every-day business application. After several executions on the same data set, it turned out that the larger the threshold value for the maximum distance allowed between clusters that are to be merged together the better the outlier detection. However, the method failed to identify the records found by the previously described methods. This fact is most likely due to the fact that a large number of fields does not allow detection of not-so-obvious outliers (records with less than 1% of the fields having outlier values), or the Euclidian distance is not a very good similarity

measure in this case. A faster clustering algorithm could be utilized that may allow automated tuning of the maximum cluster size, as well as scalability to larger data sets.

A data visualization tool (ADVIZOR by Visual Insight Inc.) and Microsoft Excel were used to verify and analyze the results. The presented figures were obtained using ADVIZOR.

## 5. Conclusions and Future Work

The data cleansing tool that has been developed and tested represents the first parts of the automated data-cleansing framework. It addresses a rather general domain of problems. In order to maintain its generality, it makes little use of the domain knowledge. The only significant element extracted from the domain is that the first field in the data set should always have the lowest value in the record. However, this is a condition that does not greatly restrict the domain.

The approach taken – using outlier detection methods to identify potential errors in the data – is different from the existing data cleansing tools and methods. It does not regard the database integration problem, but addresses a single database in use with unknown quality of its data. This is the case in many businesses inside and outside the data-warehousing field. The current tool can be used to identify seven types of potential errors in the data set.

As part of the automated data cleansing process, this tool is an essential part but human interaction is still needed to prepare the data set and interpret the some of the results. However,

the tool as it is, already performs better than a human expert even on small data sets. Experiments showed that existing methods in outlier detection are applicable to data cleaning. The experiments also highlighted some shortcomings of a number of these methods (such as hierarchical clustering) when applied to large data sets in the attempt to identify not-so-obvious outliers.

The results generated can not only be used to identify the potential errors and attempt to correct them, but can be used as a basis to define a number of data quality metrics. The error correction issue that will be addressed in future research and will differ form the current general approach since extensive domain specific knowledge will be necessary.

## 6. References

[Ballou89] Ballou, D. and Tayi, K.: "Methodology for Allocating Resources for Data Quality Enhancement", *Communications of the ACM*, Vol. 32, No. 3, 1989, pp. 320-329

[Barnett94] Barnett, V., Lewis, T.: *Outliers in Statistical Data*, John Wiley and Sons, 1994

[Bock98] Bock, R.K., Krischer, W.: "The Data Analysis Briefbook", Springer, 1998

[Brachman96] Brachman, R. J. and Anand, T.: “The Process of Knowledge Discovery in Databases: A Human-Centered Approach”, in *Advances in Knowledge Discovery and Data Mining, MIT Press/AAAI Press*, 1996, pp. 97-58

[EDD99] EDD. Home page of DataCleanser tool: http://www.npsa.com/edd/, last accessed 01/15/2000

[English99] English, J.: Column "Plain English on Data Quality", *DM Review*: http://www.dmreview.com, last accessed 02/10/99

[ETI99] E. T. International. Home page of ETI-Data Cleanse tool. http://www.evtech.com/products/dc2.html, last accessed 01/15/2000

[Fayyad96] Fayyad, U. and Piatetsky-Shapiro, G. and Smyth, P.: “From Data Mining to Knowledge Discovery: An Overview”, in *Advances in Knowledge Discovery and Data Mining, MIT Press/AAAI Press* 1996, pp. 1-36

[Hernandez97] Hernandez, M. A. and Stolfo, J. S.: “Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem”, *Journal of Data Mining and Knowledge Discovery*, No. 2, 1998, pp. 9-37

[Flanagan98] Flanagan, T. and Safdie, E. (Eds.): “A Practical Guide to Achieving Enterprise Data Quality”, http://www.techguide.com/ last accessed, 12/01/99

[Fox95] Fox, C., Levitin, A., Redman, T., "Data and Data Quality", in *Encyclopedia of Library and Information Science*, 1995

[Fox94] Fox, C. and Levitin, A. and Redman, T.: "The notion of Data and Its Quality Dimensions", *Information Processing and Management*, Vol. 30, No. 1, 1994, pp. 9-19

[Galhardas99] Galhardas, H. and Florescu, D. and Shasha, D. and Simon, E.: “An Extensible Framework for Data Cleaning", *Technical Report, Institute National de Recherche en Informatique ét en Automatique*, 1999

[Guyon96] Guyon, I. and Matic, N. and Vapnik, V.: "Discovering Information Patterns and Data Cleaning", in *Advances in Knowledge Discovery and Data Mining, MIT Press/AAAI Press* 1996, pp. 181-203

[Hamming80] Hamming, R. W.: "Coding and Information Theory", Prentice-Hall, New Jersey, 1980.

[Innovative99] Innovative Systems Inc.: http://www.innovgrp.com last accessed 09/15/99

[Kimball96] Kimball, R.: “Dealing with Dirty Data”, *DBMS*, Vol. 9, No. 10, September 1996, p. 55

[Knorr97] Knorr, E. M. and Ng, R. T.: “A Unified Notion of Outliers: Properties and Computation”, *Proceedings of KDD 97*, pp. 219-222

[Levitin95] Levitin, A. and Redman, T.: "A Model of the data (life) cycles with application to quality", in *Information and Software Technology*, Vol. 35, No 4, 1995, pp. 217-223

[Moss98] Moss, L.: "Data Cleansing: A Dichotomy of Data Warehousing", *DM Review*, February 1998

[Murtagh83] Murtagh F., “A Survey of Recent Advances in Hierarchical Clustering Algorithms”, *The Computer Journal*, Vol. 26, No. 4, 1983, pp. 354-359

[Orr98] Orr, K.: "Data Quality and Systems Theory" *Communications of the ACM*, Vol. 41, No. 2, February 1998, pp. 66-71

[Pak93] Pak, S. and Pando, A.: "Data Quality Analyzer: A Software Tool for Analyzing Data Quality in Data Manufacturing Systems", on-line: http://web.mit.edu/tdqm/papers/93/pass1/93-10.html, 1993, last accessed 02/10/99

[QMSoft99] Qualitative Marketing Software. Home page of Centrus Merge/Purge Module: http://www.qmsoft.com/solutions/Merge.htm last accessed 01/15/2000

[Redman98] Redman, T.: "The Impact of Poor Data Quality on the Typical Enterprise", *Communications of the ACM*, Vol.41, No.2, February 1998, pp. 79-82.

[Redman96] Redman, T.: *Data Quality for the Information Age*, Artech House, 1996

[Simoudis95] Simoudis, E. and Livezey, B. and Kerber, R.: "Using Recon for Data Cleaning", in *Proceedings KDD 1995*, pp. 282-287

[Strong97] Strong, D. and Yang, L. and Wang, R.: “Data Quality in Context”, *Communications of the ACM*, May 1997, Vol. 40, No. 5, pp. 103-110

[Svanks84] Svanks, M.: "Integrity Analysis: Methods for Automating Data Quality Assurance", *EDP Auditors Foundation*, Vol. 30, No. 10, 1984, pp. 595-605

[Trillium99] Trillium Software System for data warehousing and ERP: http://www.trilliumsoft.com/products.htm last accessed 01/15/2000

[Vality99] Vality Technology. Home page of the INTEGRITY Data Re-engineering Environment: http://www.vality.com/ last accessed 01/15/2000

[Wang93] Wang, R. and Reddy, M., P. and Gupta, A.: "An Object-Oriented Implementation of Quality Data Products", in Proceedings WITS, 1993

[Wang95] Wang, R. and Storey, V. and Firth, C.: "A Framework for Analysis of Data Quality Research", *IEEE Trans On Knowledge and Data Engineering*, Vol. 7, No. 4, August 1995, pp. 623-639

[Wang96] Wang, R. and Strong, D. and Guarascio, L.: "Beyond Accuracy: What Data Quality Means to Data Consumers", in *Journal of Management Information Systems*, Vol. 12, No 4, Spring 1996, pp. 5-34

[Yang99] Yang, Y. and Carbonell, J. and Brown, R. and Pierce, T. and Archibald, B. T. and Liu, X.: "Learning approaches for Detecting and Tracking News Events", *IEEE Intelligent Systems*, Vol. 14, No. 4, July/August 1999

[Zhang97] Zhang, T. and Ramakrishnan, R. and Livny, M.: "BIRCH: A New Data Clustering Algorithm and Its Applications", *Data Mining and Knowledge Discovery*, no. 1, 1997, pp. 141-182