ATTENUATING THE EFFECT OF DATA ABNORMALITIES ON DATA WAREHOUSES

Anália Lourenço and Orlando Belo
Department of informatics, School of Engineering, University of Minho
4710-057 Braga, Portugal

Keywords: Data warehousing systems, ETL Processes, Agent-Based Computing, Data Processing.

Abstract: Today's informational entanglement makes it crucial to enforce adequate management systems. Data warehousing systems appeared with the specific mission of providing adequate contents for data analysis, ensuring gathering, processing and maintenance of all data elements thought valuable. Data analysis in general, data mining and on-line analytical processing facilities, in particular, can achieve better, sharper results, because data quality is finally taken into account. The available elements must be submitted to an intensive processing before being able to integrate them into the data warehouse. Each data warehousing system embraces extraction, transformation and loading processes which are in charge of all the processing concerning the data preparation towards its integration into the data warehouse. Usually, data is scoped at several stages, inspecting data and schema issues and filtering all those elements that do not comply with the established rules. This paper proposes an agent-based platform, which not only ensures the traditional data flow, but also tries to recover the filtered data when an data error occurs. It is intended to perform the process of error monitoring and control automatically. Bad data is processed and eventually repaired by the agents, integrating it again into the data warehouse's regular flow. All data processing efforts are registered and afterwards mined in order to establish data error patterns. The obtained results will enrich the wrappers knowledge about abnormal situations' resolution. Eventually, this evolving will enhance the data warehouse population process, enlarging the integrated volume of data and enriching its actual quality and consistency.

1 INTRODUCTION

Information systems are the core of any organization. Every event occurred within organization's environment is registered and eventually processed. Information acts as a key differentiator. Competing businesses use information to grant customers' attention, preserving their market share. However, the value of information depends on quality, i.e., data quality sets the trust one can have on decisions made upon the available elements (Hipp et al. 2001). In fact, decision support systems inherited two main goals: to manage and make available valuable elements for analysis, and to ensure data quality of their contents through adequate processing (Naumann 2001)(Guess 2000).

One of the intentions of data warehousing systems is to provide adequate contents for data analysis, ensuring gathering, processing and maintenance of data elements that was considered relevant. Data analysis, in general, and data mining and on-line analytical processing facilities in particular, can achieve better, sharper results, because data quality is finally taken into account. Data consistency, integrity and non-volatile are main premises that are preserved at all cost.

However, the process of extracting, transforming and loading data into the data warehouse is anything less straightforward. The scenario is inherently heterogeneous. Gathering every piece of information that is available and thought useful brings along the conciliation of different data models and data schemas, as well as the usual single-source conflicts, inconsistencies and errors.

This paper presents an agent-based approach to the traditional *Extraction, Transformation and Loading* (ETL) process that populates data warehousing systems. The proposed platform not only ensures the traditional data flow, but also contributes to minimizing the loss of data. It performs error monitoring and control automatically, recurring to software agents. The aim was set on assisting the process, learning from past experiences

Lourenço A. and Belo O. (2004).
ATTENUATING THE EFFECT OF DATA ABNORMALITIES ON DATA WAREHOUSES.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 411-415
DOI: 10.5220/0002629604110415

and thus providing more knowledge to wrappers about how to deal with abnormal data formats.

In order to accomplish this, a multi-agent environment was conceived following the specifications and directives of the *Foundation for Intelligent Physical Agents* (FIPA). The intention is to turn the system as generic, standard, robust and flexible as possible. Moreover, the system was developed recurring to *Java Agent DEvelopment Framework* (JADE), a FIPA-compliant facility that delivers all basic features to the creation and management of a system of this nature. Eventually, this evolving will enhance the data warehouse population process, enlarging the integrated volume of data and enriching its actual quality and consistency.

2 DATA WAREHOUSING SYSTEMS: A HETEROGENEOUS SCENARIO

2.1 An Overview

Data heterogeneity is an unavoidable problem within data warehousing environments (Lee et al. 1999) (Jeusfeld et al. 1998). By definition, the purpose of these systems is to provide the most complete view of the organization. The available data sources are quite diverse and can range from conventional database systems to non-conventional sources like flat files, HTML and XML documents, knowledge systems and legacy systems. Thus, reuniting all distinct, available data elements and building up an unified content becomes a necessity.

While merging data coming from distinct data sources, several kinds of problems pump up. Besides “normal” single-source problems, there are now different data models and data schemas to take care of (Rahm & Do 2000). The occurrence of different categories of errors depends on the intervenient data sources and, more important, on their heterogeneity. The data quality of a certain data source largely depends on the schema and integrity constraints that control the permissible data values (Fox et al. 1994). When we are dealing with sources without schema, like flat files, the probability of occurrence of errors and inconsistencies is very high as there are not set any restrictions to the inputs. When sources are governed by some sort of data model, like it happens with database systems, part of the damage can be prevented.

2.2 Abnormal Data Formats

Abnormal data can be classified into the following categories: incomplete data, incorrect data, incomprehensible data, inconsistent data and schema conflicts (both naming and structural).

Records or fields that are missing and that, by design, are not being filled in, belong to the first category. Wrong (although valid) codes, erroneous calculations and aggregations, duplicate records and wrong information compose the second one. Incomprehensible data embrace situations like multiple fields put in one field, weird formatting, unknown codes and confusing many-to-many relationships. Inconsistencies may appear at different levels such as coding, business rules, aggregations, timing and referential integrity. Besides, it is not exactly a surprise that, sometimes, different codes are associated to the same object or that the same code assumes different meanings or even, different codes have the same meaning, as well as, there may be overlapping codes.

On the other hand, there are conflicts that emerge from the schema rather than from the data itself. Schema conflicts can be divided into naming conflicts and structural conflicts (Doan et al. 2001). Naming issues are related to the use of homonyms and synonyms, i.e., when the same name is used for two different concepts and when the same concept is described by two or more names, respectively. Structural conflicts appear as a result of a different choice of modelling constructs or integrity constraints, arising problems with existing types, dependencies, keys along with behavioural conflicts.

ETL processes are quite complex. A real-world system embraces a large number of programs, temporary data repositories and mapping tables for data transformation and key generation, among other items. All these elements have to be very well conciliated, preventing the occurrence of breaks along the way and ensuring the data compliance with the specified requisites. The implementation, and implantation, of abnormal data situations detection and control mechanisms demands for more resources and an additional effort from the system’s administrator. Typically, most detected error occurrences have to be worked out by the administrators themselves. Obviously, their primary concern is to ensure the non-stopping flow towards the data warehouse, without putting in risk the quality and consistency of the repository.

2.3 Relevant Resolution Schemas

Data cleaning (also known as data scrubbing or data cleansing) is a relative new research field (Maletic & Marcus 2000) (Marcus & Maletic 2000). Computationally, the process is very expensive, requiring leading technology that was not available till very recently. Researchers are only now attempting to tackle issues like dealing with missing data, determining record usability or resolving erroneous data.

There are still many kinds of problems that have no solution at sight. Not all kinds of errors can be eliminated using automated tools and the development of strategies is most of the times dependent of the interests of the application areas. Right now, data cleaning methods take care of five major error categories: missing values, outliers, inconsistent codes, schema integration and duplicates. Missing values can be worked out using a co-relation with another attribute, i.e., by finding some rule between the attribute containing missing values and another one to whom it is somehow related. Another possibility is to determine an adequate polynomial model capable of deriving the missing values. Outliers are an issue a little bit more complex. Almost all studies that consider outlier identification are performed within Statistics and follow one of the following strategies (Knorr & Ng 1997):

– the calculation of statistical values (averages, standard deviation, range), based on Chebyshev’s theorem and considering the confidence intervals for each field (Bock & Krischer 1998) (Barnett & Lewis 1994);
– the identification of the data patterns that apply to most records, combining techniques such as partitioning and classification;
– the appliance of clustering techniques based on the Euclidian distance (Miller & Mayers 2001).

Duplicates make their entrance during the merge of different sources, and there are three algorithms considered particularly suitable for large volumes of data (Hernandez & Stolfo 1998) (Monge 1997) (Hernandez 1996): the “N-gram sliding window”, the “sorted-Neighborhood method” and the “domain independent Priority-Queue algorithm”. Inconsistent codes can be resolved by using a code repository. The number of existing codes is quite small when compared to the overall volume of data. Therefore, it is feasible to prepare a hash table of codes, checking each appearing code against the table’s entries and verifying its correctness.

3 THE AGENT PLATFORM

The proposed platform is intended to perform the process of error monitoring and control automatically, recurring to software agents (Jennings 2000) (Wooldridge & Ciancarini 2001). The usual tasks performed in any given data warehousing system stay and share now room with abnormal data formats resolution tasks.

The aim was set on assisting the process, learning from past experiences and thus, evolving wrappers knowledge about abnormal situations’ resolution. In order to accomplish this, a multi-agent environment was conceived and modelled following the specifications and directives of the FIPA. By doing so, it is intended that the system is as generic, standard, robust and flexible as it is possible. Moreover, the system was developed recurring to JADE, a FIPA-compliant facility that delivers all basic features to the creation and management of a system of this nature. Eventually, this evolving will enhance the data warehouse population process, enlarging the integrated volume of data and enriching its actual quality and consistency.

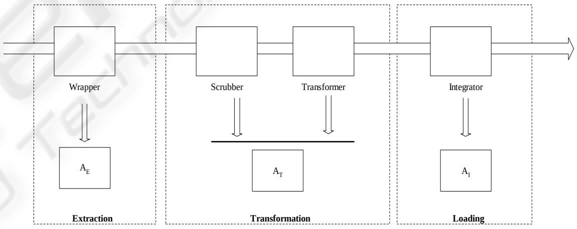

Figure 1: The distinct abnormal data processors

As Figure 1 illustrates, each platform embraces two different processing modules:

– the usual tasks related to the data warehouse feeding;
– the tasks involved in the abnormal data formats treatment.

In the particular case of the extraction platform, there are two main software agent classes: the data extraction agents and the error analysis and treatment agents. The first ones are typical wrapping programs that extract data according to pre-defined user directives and execute error detection and classification procedures. The analysis agents are concerned with the abnormal situations notified by the extraction agents. They manage these situations, proposing possible solutions and elaborating reports about the abnormal situations’ resolution attempts.

When a wrapper spots such a situation, i.e., detects a given piece of data that does not conform to the established quality standards, the process is deployed. If agents are able to recover the affected data, data will move on to the next ETL stage. Otherwise, the occurrence will be reported to the analysis agents and it will be up to them to keep the treatment process going. If they are able to solve the error, data will be back on track and if not, data will be discarded.

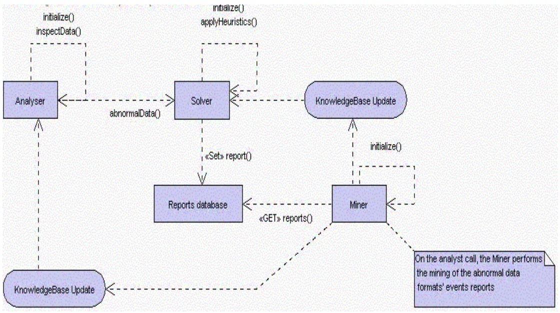

Figure 2: The main agents’ general procedures.

There are three main agent groups (Figure 2): analyzers, solvers, and miners. Analysers inspect data, spotting possible error situations and when one pumps up, they send it to a solver agent. Actually, the detected abnormal situations are checked against the resolution rules (based on heuristics) of a solver agent; if a analyser is capable, by itself, to solve the issue, the work is done, recorded in the abnormal situations reports’ database and, the correct data is goes back to the analysis level (main data stream). On the contrary, if a analyser is incapable of solving the problem, it posts a help message to the rest of the community and a record gives entry in the abnormal situations reports’ database. If another agent has the expertise to solve the case, the announcement will be made. There will always be a report about the abnormal situation and all the efforts that were taken in order to repair it. These reports are stored in a database that, from time to time, is mined, trying to understand the occurrences. By doing so, it will be possible to evolve agents’ knowledge bases towards sharper detection and more skilled resolution.

Obviously, in order to sustain all these activities, the agents have to be conveniently instructed, through specialized knowledge acquisition procedures. It is necessary to collect information concerning the different kinds of errors that we what to cover and the correspondent resolution schemas. All this information is then integrated into the knowledge bases of the analysis and solvers agents.

Moreover, the agents have to be instructed about their communication patterns, i.e., they have to know with whom they can talk to and when each interaction situation is in order. This implies the existence of a robust communication medium and an adequate communication language, capable of sustaining all inter-agent communication acts.

In this sense, the choice was to set a communication model that stands over FIPA-compliant messages (Figure 3) and the multi-agent environment is integrated in the JADE facility (Pitt & Bellifemine 1999), a middle-ware that complies with the FIPA specifications and delivers a set of tools that support the debugging and deployment phase (Bellifemine et al. 2001) (Bellifemine et al. 1999). It allows the spread of the agents across multiple platforms which is essential for this kind of application area, as well as, supports the agents’ configuration control via a remote GUI. Moreover it sustains distinct communicative acts and allows the creation of ontologies according to the application area.

```
(request-resolution-schema-begin,                   // message type
  ['wrapper', '999.999-99-99'],                     // sender identification
  '2003-06-01 14:00:00',                            // date
  [('extraction-error-manager', '999.999-99-00')],  // receiver identification
  [('wrapper', '999.999-99-99')],                   // receiver for the answer
  'FIPA-Based-ACL',                                 // communication language
  [resolution-schema-suggestion],                   // expected message
  [(*'information-source', '*system', '*table',     // message’s body
       155,’*attribute3’, ’NULL’ )]
)
```

Figure 3. A wrapper’s message example.

However, resolving these kind of situations is anything less straightforward. Sometimes, the local abnormal situations’ manager has not enough knowledge to work around the problem. When this happens, the manager takes the initiative of posting the wrapping problem to a “higher” manager, hoping that this helping effort will permit data recovery.

4 CONCLUSIONS

This paper presented an agent-based platform that targets both regular ETL process and abnormal data formats identification and resolution within the data warehousing scenario. The aim is set on optimising the volume and more important than that, the quality of the data that populates these systems. Unquestionably, ETL processes are quite complex, consuming a large amount of processing resources. Data keeps evolving and new abnormal data formats are always pumping in. It is no longer feasible to sustain such a process manually, urging for automatic monitoring and control tools.

The software agents are able to balance work, setting specialised workers for certain tasks (such as abnormal data formats recognition and repair). Moreover, it is intended to learn from the past experiences, evolving the cooperative work. Therefore, each event is recorded in a database that, from time to time, it is mined. The obtained results will be studied in order to better understand the abnormal data formats that are appearing and thus, keeping up-to-date the wrappers and the solvers knowledge bases.

This platform is a FIPA-compliant, JADE sustained multi-agent system. It allows the spread of the agents across multiple platforms, as well as, supports the agents' configuration control via a remote GUI. Moreover it sustains distinct communicative acts and allows the creation of ontologies according to the application area.

In the future, new resolution rules for abnormal data formats identification and resolution will be integrated. Also, the other two platforms, concerning the transformation and the loading stages of the process, will be deployed in a similar way.

REFERENCES

Barnett, V. and Lewis, T., 1994. Outliers in Statistical Data. John Wiley and Sons.

Bellifemine, F., Poggi, A. and Rimassa, G. 2001. Developing multi agent systems with a FIPA-compliant agent framework. In *Software - Practice And Experience*, no. 31, pp. 103-128.

Bellifemine, F., Poggi, A. and Rimassa, G. 1999. JADE – A FIPA-compliant agent framework. CSELT internal technical report. Part of this report has been also published in Proceedings of *PAAM'99*, pp.97-108. London, United Kingdom.

Bock, R.K. and Krischer, W. 1998. The Data Analysis Briefbook. Springer.

Doan, A., Domingos, P. and Levy, A. 2001. Reconciling Schemas of Disparate Data Sources: A Maching-Learning Approach. In *SIGMOD*, pp. 509-520.

Fox, C. J., Levitin, A. and Redman, T. 1994. The Notion of Data and Its Quality Dimensions. Information Processing and Management 30(1): 9-20.

Guess, F. 2000. Improving Information Quality and Information Technology Systems in the 21st Century. Invited talk and paper for the *International Conference Statistics in the 21st Century*.

Hernandez, M. A. 1996. A Generalization of Band Joins and the Merge/Purge Problem. Ph.D. thesis, Columbia University.

Hernandez, M. A. and Stolfo, S. J. 1998. Real-world data is dirty: Data Cleansing and the Merge/Purge problem. *Journal of Data Mining and Knowledge Discovery*, 2(1):9-37.

Hipp, J., Guntzer, U. and Grimmer, U. 2001. Data Quality Mining - Making a Virtue of Necessity. In Proceedings of *the 6th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD 2001)*, pp. 52-57.

Jennings, N. 2000. On Agent-Based Software Engineering. *Artificial Intelligence*, 117 (2) 277-296.

Jeusfeld, M.A., Quix, C. and Jarke, M. 1998. Design and analysis of quality information for data warehouses. In Proceedings of *the 17th Int. Conf. On Conceptual Modeling*, pp. 349-362. Singapore, China.

Knorr, E. M. and Ng, R. T. 1997. A unified notion of outliers: Properties and computation. In Proceedings of *the KDD Conference*, pp. 219-222.

Lee, M.L., Lu, H., Ling, T. W. and Ko, Y. T. 1999. Cleansing Data for Mining and Warehousing. In Proceedings of *the 10th International Conference on Database and Expert Systems Applications (DEXA)*, pp. 751-760. Florence, Italy.

Maletic, J. and Marcus, A. 2000. Automated Identification of Errors in Data Sets. The University of Memphis, Division of Computer Science, Technical Report.

Marcus, A. and Maletic, J. 2000. Utilizing Association Rules for the Identification of Errors in Data. Technical Report TR-14-2000. The University of Memphis, Division of Computer Science, Memphis.

Miller, R. and Myers, B. 2001. Outlier Finding: Focusing User Attention on Possible Errors. In the Proceedings of *UIST Conference*, pp. 81-90.

Monge, A. 1997. Adaptive detection of approximately duplicate database records and the database integration approach to information discovery. Ph.D. Thesis, University of California, San Diego.

Naumann, F. 2001. From Databases to Information Systems - Information Quality Makes the Difference. In Proceedings of *the International Conference on Information Quality*.

Pitt, J. and Bellifemine, F. 1999. A Protocol-Based Semantics for FIPA '97 ACL and its implementation in JADE. CSELT internal technical report. Part of this report has been also published in Proceedings of *AI*IA*.

Rahm, E. and Do, H. H. 2000. Data Cleaning: Problems and Current Approaches. Bulletin of the Technical Committee on Data Engineering, vol. 23, n.4, pp. 3-13. IEEE Computer Society.

Wooldridge, M. and Ciancarini, P. 2001. Agent-Oriented Software Engineering: The State of the Art. In Paolo Ciancarini and Michael Wooldridge (editors), Agent-Oriented Software Engineering. Springer-Verlag Lecture Notes in AI Volume 1957.