ABNORMAL DATA FORMATS IDENTIFICATION AND RESOLUTION ON DATA WAREHOUSING POPULATING PROCESS

Anália Lourenço **Orlando Belo**

Universidade do Minho, Campus de Gualtar, 4710-057 Braga, Portugal
{analia,obelo}@di.uminho.pt

ABSTRACT

Data Warehousing systems are perhaps one of the most valuable assets that organisations possess today. They manage and sustain crucial, strategic information, granting the urging decision support. However, before taking advantage of this magnificent resource, there has to be set a plan to ensure its population. The process of extracting, transforming and loading data into the data warehouse is anything less straightforward. These scenarios are inherently heterogeneous. The idea of gathering every piece of information that is available and thought useful brings along different data models and data schemas to conciliate. Besides, within each single source, it is likely that several kinds of conflicts, inconsistencies and errors pump up. Therefore, tools capable of identifying and resolving these situations are in order.

This paper aims to bring some light into the subject, covering basic issues related with data cleaning, as well as, proposing a new computational platform - an agent-based abnormal data formats identification and resolution platform. The aim was set on assisting the process, learning from past experiences and thus, evolving 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.

KEYWORDS: Data Warehousing systems, data cleaning and integration, agent-based systems, FIPA, JADE, and Knowledge Discovery in Databases.

1. INTRODUCTION

The problem of conciliating data coming from multiple, heterogeneous information sources is an old acquaintance. There has always been this necessity and the appearance of new, larger and richer information sources only have made things worse. It is no longer just a matter of "digesting" flat files, paper documents and databases. The implantation and wide acceptance of the Information Systems within organizations impelled the generation and collection of huge, diverse volumes of data, as well as, the need for convenient analysis and consequently, their adequate processing [Naumann 2001] [Guess 2000].

Data entry and acquisition is inherently prone to errors. Most of the times, efforts are made to prevent these errors at the front-end level (i. e., at the entry point), but the fact is that errors often remain and have to be dealt with in the back-end through intensive data processing. The new generations of Information Systems due to their specificities in terms of data volumes, query response times and analysis requirements, strongly prohibit any manual attempts, urging the creation of automatic, intelligent extraction and integration platforms, namely wrappers [Wiederhold 1992] [Roth & Schwarz 1997] [Kushmerick et al. 1997] [Kuhlins & Tredwell 2002]. However, most of the available programs provide limited support for data cleaning, focusing mainly on data transformations for schema translation and integration [Doan et al. 2001] [Maletic & Marcus 2000a].

There are many kinds of abnormal situations, both instance-related and schema-related. From a single-source perspective, issues such as misspellings, duplicates, contradictory values, and the lack of integrity constraints are to be attended. However, the problem rises way up when trying to mingle multiple, eventually heterogeneous sources. All single-source problems are present along with the ones emerged during the conciliation attempt of different data models and schema designs.

Data warehousing scenarios are the perfect example as they are inherently heterogeneous. The idea of gathering every piece of information that is available and thought useful, brings along different data models and data schemas, as well as, the conventional data errors. When data is extracted from the sources, it is inspected and processed in order to obtain a single, homogenous, consistent data volume, ready to be integrated into the data warehouse. It is during this processing that wrappers encounter inconsistencies, conflicts and errors. Some they are able to repair, some they are not and, consequently, this will cause a loss of information. Within this scenario, an agent-based abnormal data formats identification and resolution platform is presented here. This platform is a Foundation for

Intelligent Physical Agents (FIPA)-compliant, Java Agent DEvelopment Framework (JADE) sustained multi-agent system and its aim is to optimize the feeding process of data warehouses, preventing the loss of as many pieces of data as it is possible.

The idea behind it is to learn from past processing experiences, detecting new kinds of abnormal situations and suggesting resolution schemas that will enhance the wrappers' knowledge base. In fact, agents split the work among them in such a way that there are some that examine the data – the analysers -, sustaining the regular Extraction, Transformation and Loading (ETL) data flow. When an abnormal situation is detected, a notification is sent to the solvers' group which will be in charge of trying to find a solution. If there is knowledge to resolve it, data is repaired and integrated into the main data flow again and otherwise, it will be discarded. These actions result in a report that is stored in the abnormal situations information database. When a mining task is deployed, the miner agents gather the information from this database and apply pre-established mining algorithms, searching for valuable knowledge. The mining results are then processed and, if possible, they are transformed into new identification-resolution schemas that will enrich the agents' knowledge base.

The paper is organized in a way that first data processing issues are studied in general, theoretical terms and then, there is a full briefing about the conceived platform and its agents. Section 2 presents a description of the data cleaning process as a whole and a rough classification both of the existing data sources and the cleaning issues associated to each one of them. Then, some of the most common methods applied to major errors categories are presented. Section 3 is entirely devoted to the characterization of the platform. A general view of the abnormal situations mining facility is given and then, a closer look is directed towards its main agents groups. Conclusions will resume the presented work, pointing out future work.

2. DATA PROCESSING IN DATA WAREHOUSING SCENARIOS

Relatively, data cleaning, also called data scrubbing or data cleansing, is a new research field [Maletic & Marcus 2000b] [Marcus & Maletic 2000]. The process is computationally very expensive, requiring leading technology that was not available till very recently. In fact, there are many issues in data cleaning that researchers are only now attempting to tackle like dealing with missing data, determining record usability, resolving erroneous data, etc.

Within data warehousing scenarios, the urge is obvious due to the inevitable merge of distinct sources' information [Lee et al. 1999]. By definition, data warehouses are complex systems that have to deliver highly-aggregated, high quality data from heterogeneous sources to decision-makers [Jeusfeld et al. 1998]. 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. Moreover, the role of wrappers in the data warehousing context is enlarged. On one hand, there is the description of the data kept by each source in a common data model (the typical wrapper functionality) and, on another hand, there is the detection of changes occurred on the underlying data sources (the so-called change monitoring). The latest is a specific functionality required by data warehousing systems in order to ensure the incremental refreshment of their repository. This brings a new perspective to the integration process as quality goals like timeliness, accessibility and others [Strong et al. 1997] [Wang et al. 1996] [Wang et al. 1995], take the place of the previously absolute consistency goals which become now relative [Fox et al. 1994]. The main focus is set on identifying overlapping data, the so-called merge/purge problem, often instantiated in the literature as record linkage, semantic integration, instance identification, duplicate elimination or object identity [Hernandez & Stolfo 1998] [Hernandez & Stolfo 1995]. The problem resides in the fact that, often, records referring to the same entity are eventually represented in different formats within distinct data sources or are represented erroneously, generating duplicate records in the merged database. If this situation is not overcome, decision support could be seriously affected.

2.1 ERRORS CATEGORIZATION

The occurrence of different categories of errors depends on the intervenient data sources and, more important, on the presence or not of heterogeneous data sources [Rahm & Do 2000]. The data quality of a certain data source largely depends on the schema and integrity constraints that control the permissible data values. 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 ruled by some sort of data model, like it happens with database systems, part of the damage can be prevented. Schema-related issues pump up because of data model limitations, poor schema designs or an insufficient number of integrity constraints. It can be that the existing constraints do not properly describe the application's specificities. Probably, this is due to an inefficient analysis of requirements or an intentional cut to limit the overhead brought by integrity control. At the instance level, there are all the issues that could not be avoided at the schema level, such as misspellings, duplicates and contradictory values.

When multiple sources are reunited, each source's problems go along. Typically, sources are developed, deployed and maintained independently, serving specific needs. This gives rise to a large degree of heterogeneity in terms of data management, data models, schema designs and contents. Data may be represented differently, overlap or contradict across sources. Schema translation and integration are in order to deal with data model and schema design differences, while, at the instance level, all the individual data problems have now to share room with problems concerning different value representations and interpretations, varying aggregation levels and distinct timing.

In fact, the errors that call out for data cleaning intervention can be classified into the following categories: incomplete data, incorrect data, incomprehensible data, inconsistent data and schema conflicts (both naming and structural). Missing records or fields and records or fields that, by design, are not being filled in belong to the first type. 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 into a single 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, apart from the fact that there may be overlapping codes.

Finally, 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. 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. Structural conflicts appear as a result of a different choice of modelling constructs or integrity constraints, arising problems with existing types, dependencies and keys along with behavioural conflicts.

2.2 ERROR RESOLUTION SCHEMAS

Broadly speaking, data cleaning methods embrace five major error categories: missing values, outliers, inconsistent codes, schema integration and duplicates. It cannot be forgotten that not all types of errors can be eliminated using automated tools and that the development of strategies depends on the particular interests of the application areas.

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. For instance, if the total revenue field was not filled in, it is possible to calculate the correspondent value based on the items entries. Another possibility is to scan the attribute's values and determine an adequate polynomial model capable of deriving the missing values.

Outliers are a little bit more complex. 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. Almost all studies that consider outlier identification as their primary objective are in Statistics [Knorr & Ng 1997]. The three major approaches to the subject recur to one of this three strategies: 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]; existing data patterns, combining techniques such as partitioning and classification to identify the patterns that apply to most records; and, the appliance of clustering techniques based on the Euclidian distance [Miller & Mayers 2001].

Inconsistent codes can be resolved by using a codes' 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.

Duplicates pump up 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". A deeper analysis of these algorithms is far beyond the scope of this work and the reader should address the given references for further details.

3. THE ABNORMAL DATA FORMATS IDENTIFICATION AND RESOLUTION MODEL

Nowadays, the diversity of data sources is so great and data volumes are growing so fast that automation is an urge. In spite of all good intentions that decision support facilities and database management systems might have, very little can be accomplished without quality enforcement [Hipp et al. 2001].

Supporting automated data cleaning implies defining and determining error types, searching and identifying error instances and correcting the uncovered errors. Implementing error control and detection mechanisms within data warehousing environments is computationally consuming and demands an extra effort from the system's administrator. Most of the times, many of the unclosed errors are dealt with by the administrators themselves. In this sense, their primary concern is always to ensure the population process without jeopardizing the quality and consistency of the data warehouse.

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 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.

3.1 AGENT-BASED ABNORMAL SITUATIONS ANALYSIS

The conceived platform can be viewed as an add-on component of a regular data warehousing system. The usual tasks performed in any given data warehousing system are preserved and enforced, including now extra tasks related with abnormal situations treatment. In this sense, the platform keeps 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. It is at this moment that the abnormal situation is registered at the correspondent log and the resolution attempt is made. 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 inserted into the data warehouse's data flow again and if not, data will be discarded. However, there will always be a report about the abnormal situation and all the measures that were taken, trying to repair it. These reports are stored into the abnormal situations reports' database that is from time to time mined and thus, these events will be used to enhance the whole process.

Obviously, in order to sustain all these activities, agents have to be instructed conveniently. While implementing them, knowledge acquisition procedures must be launched. 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 follow the FIPA standards and protocols. The communication model stands over FIPA-compliant messages 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.

```
for each data source do
  getData( )
  getMetadata( )
  inspectData( )

  if abnormality? then
    sendToSolver()
  else
    moveToNextStep()
  end if

end for
```

Analyser

```
for each incoming abnormality do
  inspectAbnormality( )

  if known? then
    executeResolutionSchema()
    addToMainStream()
    saveReport()
  else
    saveReport()
  end if

end for
```

Solver

```
for each requested mining do
  gatherAbnormalitiesReports()
  executeMining()
  processResults()
  proposeNewSchema()

  if accepted? then
    sendResolutionSchema()
    saveReport()
  else
    saveReport()
  end if

end for
```

Miner

Figure 1. Agents' general procedures.

As Figure 1 illustrates, there are three main agent groups: the analyzers, the solvers and the miners. Each one of them performs its own tasks and establishes contact with the others when it is judges convenient. Basically, the Analyser inspects data, spotting possible error situations and when one pumps up, it is send to the Solver. Actually, the detected abnormal situations are checked against the resolution rules of the Solver (Figure 2). If it is capable of solving 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 this agent is incapable of resolving 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.

```
case missing_value:
  if relatedAttrib? then
    applyCorrelationRule()
  end if

case outlier:
  chooseResolutionSchema()
  case chebyshev :
    generateStats()
  case pattern-based:
    identifyPatterns()
    applyMostCommonPattern()
  case euclidean:
    cluster()

case inconsistent code:
  loadHash()
  checkCode()
...

case duplicates:
  chooseResolutionSchema()
  case sliding_window :
    deployN_gram()
  case neighborhood :
    deploySortedNeighborhood()
  case queue:
    deployPriorityQueue()

default:  sendToSolver()
```

Figure 2. The executeResolutionSchema() procedure.

Finally, there is the mining stage. The knowledge extraction process can be requested through a conventional interface agent, supporting both immediate and scheduled tasks. The Miner agent wakes up when an interface or an agenda lookup agent has a task for him. He receives the task and gets in touch with the abnormal situations reports' database to acquire the dataset. While performing mining activities, such as association rules discovery and interpretation, patterns recognition and clustering, or partitioning and classification, new knowledge about abnormality identification and resolution will eventually pump up. These elements are then used to refuel the knowledge bases of the analysers and solvers, improving their work as well as enforcing data quality.

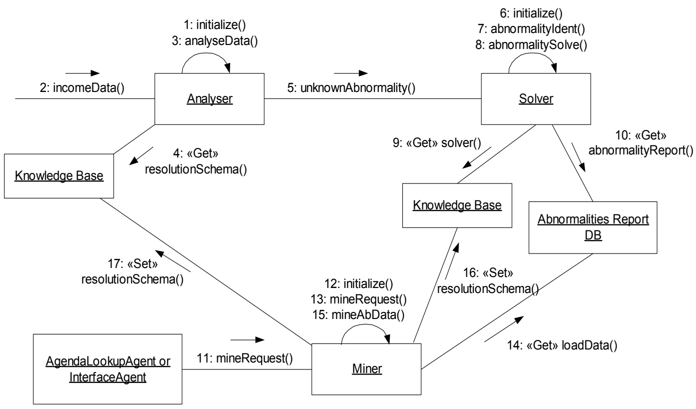

Figure 3. The agents' collaboration diagram.

In Figure 3, there is the collaboration diagram concerning the referred agents. It can be observed the interactions sustained among them and the sequence of actions deployed both internally and between agents.

4. CONCLUSIONS AND FUTURE WORK

Data warehousing systems are crucial pieces of today's organizational scenarios. Decision support is possible thanks to their contents that embrace the whole spectrum of information of the organisation. In this sense, the urge is set in guaranteeing data quality, consistency and timeliness of these repositories. Data warehouses must be always up to date and their data has to be indisputable.

The ETL process is responsible for all data management and treatment related to the migration of data from the available sources to the data warehouse. As sources heterogeneity comes with the territory, problems with data model and schema conflicts have to be taken into account along with the always existing errors, inconsistencies and conflicts in data. The problem is complex and extremely consuming in terms of resources. Data volumes keep a steady growing and new kinds of data are always pumping in. Therefore, automatic cleaning and treating has become an urge. Clearly, on one hand, there is the need of ensuring a steady flow of data to feed the data warehouse and, on the other hand, data losses should be prevented. So, apart from the conventional data processing tasks, new tasks should be introduced towards the identification of abnormal situations, their eventual resolution and thus, the recovery of data.

The agent-based abnormal data formats identification and resolution platform that has been presented throughout this paper aims to target this problem. Its agents ensure the normal ETL data flow and, at the same time, try to deal with the "dirty" data at an independent level. The multi-agent environment was conceived and modelled following the FIPA's specifications and directives. By doing so, it is intended that the system is as generic, standard, robust and flexible as it is possible. Additionally, the system is set upon the JADE facility, which makes it possible the easy creation, management and control of the community across multiple platforms. As future work, knowledge acquisition procedures must be enforced, collecting more information about all sorts of known abnormal situations and the attempts made towards their resolution. At the same time, mining techniques must be evaluated in order to identify the most adequate to each particular task. In this sense, it is intended the appliance of the platform test to real-world situations, figuring out what has to be enhance or re-arranged.

REFERENCES

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

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

[Bellifemine et al. 1999] Bellifemine, F., Poggi, A., Rimassa, G. 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. 1999.

[Bock & Krischer 1998] Bock, R.K., Krischer, W. The Data Analysis Briefbook. Springer. 1998.

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

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

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

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

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

[Hernandez & Stolfo 1995] Hernandez, M. A., Stolfo, S. J. The Merge/Purge problem for large databases. In Proceedings of the ACM SIGMOD, pp. 127-138. 1995.

[Hipp et al. 2001] Hipp, J., Guntzer, U., Grimmer, U. 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. 2001.

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

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

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

[Kuhlins & Tredwell 2002] Kuhlins, S., Tredwell, R. Toolkits for Generating Wrappers - A Survey of Software Toolkits for Automated Data Extraction from Web Sites. In Mehmet Aksit and Mira Mezini and Rainer Unland (Ed.s), Objects, Components, Architectures, Services, and Applications for a Networked World, International Conference NetObjectDays, NODe~2002, Vol. 2591, pp. 184-198. Erfurt, Germany. 2002.

[Kushmerick et al. 1997] Kushmerick, N., Weld, D. S., Doorenbos, R. B. Wrapper Induction for Information Extraction. In the International Joint Conference on Artificial Intelligence (IJCAI), pp. 729-737. 1997.

[Lee et al. 1999] Lee, M.L., Lu, H., Ling, T. W., Ko, Y. T. 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. 1999.

[Maletic & Marcus 2000a] Maletic, J., Marcus, A. Data Cleansing: Beyond Integrity Analysis. In Proceedings of the Conference on Information Quality (IQ2000), pp. 200-209. Massachusetts Institute of Technology, Boston, MA, USA. 2000.

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

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

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

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

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

[Pitt & Bellifemine 1999] Pitt, J., Bellifemine, F. 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. 1999.

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

[Roth & Schwarz 1997] Roth, M. T., Schwarz, P. M. Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources. In Proceedings of the 23rd VLDB, pp. 266-275. 1997.

[Strong et al. 1997] Strong, D. M., Lee, Y. W., Wang, R. Y. Data Quality in Context. Communications of the ACM, 40(5):103-110. 1997.

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

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

[Wiederhold 1992] Wiederhold, G. Mediators in the architecture of future information systems. In IEEE Computer 25, 3, pp. 38-49. 1992.

[Wooldridge & Ciancarini 2001] Wooldridge, M., Ciancarini, P. 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. 2001.