A rule-based methodology to support information quality assessment and improvement

Studies in Communication Sciences 4/2 (2004) 137-154

Cinzia Cappiello, Chiara Francalanci & Barbara Pernici*

A RULE-BASED METHODOLOGY TO SUPPORT INFORMATION QUALITY ASSESSMENT AND IMPROVEMENT

Data quality is an increasingly critical issue in the majority of information-intensive businesses. In such contexts, the quality of the information provided is a relevant component for the evaluation of the overall quality of service. A high data quality level is achievable by the adoption of a complete data quality management program that includes algorithms for measuring data quality and automatic techniques for recovering data when their quality decreases below acceptable values. This paper proposes a semi-automatic methodology to perform quality assessment and improvement and to provide support to organizations in achieving a high level of data quality. The methodology takes as input the desired levels of data quality to be obtained and maintained. Evaluation of quality and consequent improvement activities are triggered based on a set of predefined monitoring rules.

Keywords: data quality management architecture, rule-based data quality control.

*Department of Electronics and Information, Politecnico di Milano
cappiell@elet.polimi.it; francala@elet.polimi.it; pernici@elet.polimi.it

1. Introduction

Data quality is an increasingly critical aspect of the quality of service in information-intensive businesses. In such contexts, the primary goal of data quality assurance is the continuous control of data values and, possibly, their improvement.

The literature provides a variety of techniques for data assessment and improvement. The most straightforward solution suggests the adoption of data-oriented inspection and rework techniques, such as data bashing or data cleaning, to solve problems related to data accuracy and data consistency (English 1999). A fundamental limitation of these techniques is that they do not prevent future errors, so they are considered appropriate only when data are not modified frequently (English 1999). On the other hand, a more frequent use of data bashing and data cleaning algorithms involves high costs that can be difficult to justify. To overcome these issues, several experts recommend the use of process-oriented methods (English 1999; Redman 1996; Scannapieco et al. 2004a; Shankaranarayan et al. 2000; Wang 1998). These methods allow the identification of the causes of data errors and their permanent elimination through changes in data access and update activities. These methods are more appropriate when data are frequently created and modified.

This paper proposes a semi-automatic methodology to perform quality assessment and improvement. The methodology takes as input the levels of data quality to be obtained and maintained. Evaluation of quality and consequent improvement activities are triggered based on a set of predefined monitoring rules. Rules are defined with an initial data and process analysis and can trigger both process-oriented and data-oriented improvement actions.

The work presented in this paper extends the functionalities of the Quality Factory developed in the DaQuinCIS Project (Cappiello et al. 2003; Scannapieco et al. 2004b) with a new Monitoring module, which evaluates monitoring rules and either starts data improvement actions or alerts the organization's data quality manager if process improvement actions are needed to provide higher quality data. The paper describes the extended Quality Factory architecture and discusses the advantages and disadvantages of the rule-based methodology with reference to a case study.

The paper is organized as follows. Section 2 discusses data quality management issues in the literature. Section 3 presents the architecture

for data quality management. Section 4 explains the rule-based methodology on which the architecture is based and Section 5 reports details on its implementation and testing results.

2. Data Quality Management

The notion of data quality has been widely investigated in the literature. Several authors (Orr 1998; Wang 1998) define the quality of data as their "fitness for use", i.e., the ability of a data collection to meet user requirements or to be suitable for a specific process. Research on data quality has been carried out in several areas, including data warehousing, data cleaning, quality management in information systems and quality management on the Web. In each field, specific data quality dimensions have been defined, along with metrics, methodologies, and techniques for data quality measurement and improvement. The data quality literature provides a thorough classification of data quality dimensions, even if there are discrepancies on the definition of most dimensions due to the contextual nature of quality. The six most important classifications are Wand and Wang (1996), Wang and Strong (1996), Redman (1996), Jarke et al. (1999), Bovee et al. (2001), and Naumann (2002). By analyzing these classifications, it is possible to define a basic set of data quality dimensions including accuracy, completeness, consistency, timeliness, interpretability, and accessibility, which represent the dimensions considered by the majority of the authors. Timeliness is usually considered together with other time related dimensions, typically currency and volatility (Ballou et al. 1998).

Data quality has to be assessed and monitored continuously in order to guarantee high quality levels. To improve quality, organizations can adopt either data-oriented or process-oriented techniques. In particular, the former discover anomalies and inaccuracies by comparing values with benchmarks or by performing local analyses to detect inconsistencies and duplications. Process-oriented methods allow the identification of the causes of data errors and their permanent elimination through an observation of the whole process in which data are involved. Correction activities change data access and update procedures. They require a considerable effort for process analysis and redesign, but guarantee long-term benefits. The literature provides several frameworks that allow the representation of processes that manipulate data and the identification of the sub-processes in which data quality decreases (Shankaranarayan et al. 2000).

Several data quality programs are based on process oriented methodologies, such as TDQM (Total Data Quality Management) and TIQM (Total Information Quality Management) (English 1999; Wang 1998). These data quality programs in general require a considerable personalization effort before application. The TDQM methodology, for example, considers data as particular types of manufacturing products. The TDQM methodology cycle consists of four phases in which data quality dimensions are chosen and measured, quality problems are analyzed and improvement techniques are defined. The authors provide a series of guidelines that each organization can customize and apply by developing its own techniques and algorithms.

A fundamental open question is the association of dependable measures with data quality dimensions. The literature does not provide an exhaustive set of metrics that organizations can apply. Only a few algorithms have been developed for a subset of dimensions, such as accuracy, completeness, consistency and timeliness (Ballou et al. 1998; Naumann and Freytag 2003; Redman 1996). Quality assurance is instead faced with the need for objective measures of quality, since most users cannot judge the quality of data and simply trust data sources.

The architecture presented in this paper is aimed at providing a complete set of tools for a comprehensive data quality program that combines some of the successful techniques and algorithms presented in the literature with new algorithms and techniques. A primary goal of the architecture is to provide support in the assessment and improvement phases with a semi-automatic rule-based methodology. Rules are adopted in the assessment phase to manage the monitoring and improvement activities as described in the next sections.

3. The Data Quality Management Architecture

A total data quality management program can be achieved only with continuous data monitoring and improvement. Organizations need semi-automatic tools that can support these activities by tracking and identifying anomalies in data management. The architecture proposed in this paper, is based on the Quality Factory developed in DaQuinCIS (Data Quality in Cooperative Information Systems) project (Cappiello et al. 2003; Scannapieco et al. 2004b), in which data quality management has been addressed in the context of Cooperative Information Systems (CIS). Such systems involve multiple organizations that need to share

data in order to reach a common goal. Quality assurance in cooperative information systems is faced with the need for objective measures and evaluations of data quality are exchanged between organizations along with corresponding data. In addition, since interacting organizations may cooperate occasionally, data should be certified by the organization distributing them. In the same way in which a digital certificate authenticates a specific public key and a quality certificate (ISO 9002) guarantees specific levels of service, a data quality digital certificate authenticates the quality of data provided to a specific user by a specific source (Cappiello et al. 2004b).

The quality certification process includes different phases, such as the development and monitoring of suitable procedures to verify the level of different quality dimensions. In particular, in a cooperative information systems, each organization may have a different level of quality and in order to guarantee a consistently high level of data quality, it is necessary to design and implement a common architecture to control and improve the quality of data. The architecture proposed in the DaQuinCIS project is composed of an internal and an external infrastructure. The internal infrastructure is the Quality Factory and has to be implemented in each organization involved in the CIS to ensure a good internal data quality management. The external infrastructure is accessible to all the organizations involved in the CIS and it provides the following services: 1) it offers a set of data quality services and 2) it controls the data exchanged among organizations.

The Quality Factory is composed of four modules (Figure 1): Quality Analyzer, Quality Assessment, Monitoring, and Quality Certification. These modules interact with each other according to two different operation modes, called on-line and off-line evaluation (see Section 3.1). The functions performed by the four modules are separately discussed in the following.

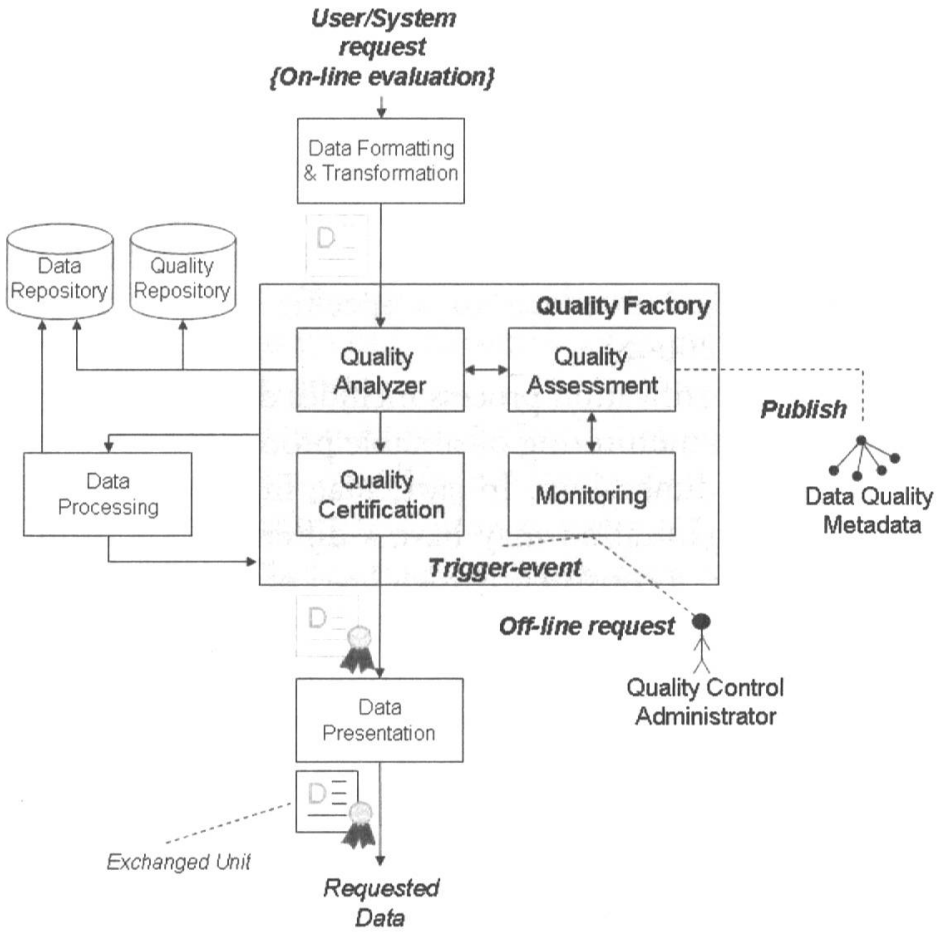

Figure 1: The Quality Factory

3.1. Evaluation processes

On-line evaluation is activated through an ad-hoc request. A data-quality oriented request is based on the submission of a query along with the specification of user quality requirements. For instance, a typical query that can be expressed is "return the name of the students attending the Master on Computer Engineering (completeness > 0.9)". The Quality Factory retrieves the data and provides an estimation of corresponding quality values.

Specifically, a generic user (internal or external or, in some cases, an application) sends a request to retrieve a given data set together with the specification of quality requirements. The request is processed by the *Data Formatting & Transformation* module that translates it into a format

that can be understood by the *Quality Analyzer*. The Quality Analyzer analyzes the request and retrieves the data from the *Data Repository*. Then, the Quality Analyzer considers user quality requirements. It extracts the information required for quality evaluation from the *Quality Repository*. The Quality Repository contains both the metadata needed for the evaluation and the results of the latest off-line quality evaluation. Assessment operations are executed by the *Quality Assessment* module which uses a set of internal measurement tools. The specific algorithms that are used by the Quality Assessment module to perform measurement operations are not presented here and the reader is referred to Cappiello et al. (2004a). The result of the evaluation, i.e., the quality metadata, is returned to the user together with the requested data set.

This approach involves an additional cost to compute the result of the query, since, differently from generic data, metadata are not simply retrieved from a local database, but are calculated with an evaluation algorithm. Furthermore, if the request involves a large amount of data, response time can be too high. For requests of large data sets, it could be more appropriate to consider the results provided by the off-line evaluation process, as discussed in the following.

Another fundamental task of the Quality Assessment module is the comparison of metadata with the quality requirements specified by the user. In the context of cooperative information systems, if an organization cannot satisfy a request with its own data, then it sends the request to the external infrastructure of the Quality Factory which verifies whether another organization in the CIS owns suitable data. This action is called *on-line improvement* and it allows an organization to increase the quality of its own data through the cooperation with other organizations. If user requirements cannot be satisfied, data are sent to the user with an alert message associated.

The results of the Quality Assessment module are also used as an input for the *Monitoring* module: if the values of one or multiple quality dimensions do not satisfy requirements, the Quality Assessment module sends an alert message to the Monitoring module, which evaluates whether quality improvement actions are needed. The message contains the data that do not satisfy quality requirements and related quality metadata. These messages are a useful source of information for the off-line improvement process, as described in Section 4.

On the contrary, if the values of all quality dimensions are acceptable, the Quality Assessment notifies the quality values to the Quality

Analyzer and a quality certificate is associated with the requested data by the *Quality Certification* module.

Before data are sent to the user, they are sent to the *Data Processing* module, which cooperates with other software applications that are in charge of preparing the final response to the user. The *Data Presentation* module sends the response to the user according to a specific format.

The *off-line evaluation* differs from the on-line evaluation since it is independent of the execution of a particular query. The Quality Factory calculates the quality metadata associated with the data belonging to an organization's databases. This calculation is triggered by the rules discussed in Section 4. Rules may trigger the evaluation periodically or as a consequence of a given internal event. An off-line evaluation can be also requested by the Quality Control Administrator to analyze a specific situation that has not been detected automatically.

With the off-line evaluation, the Monitoring module must communicate to the Quality Assessment module the data that have to be controlled and the set of data quality dimensions that have to be evaluated. The metadata that measure the results of the evaluation are stored in the Quality Repository.

The periodic assessment and storage of quality metadata allow the enterprise to answer users' queries by accessing both the database containing the requested data and the database containing corresponding quality metadata. This reduces the response time to the queries of users, but it may provide out-of-date information about the quality of data. In fact, the stored quality metadata do not take into account all the changes performed in the time interval between two periodic assessments. A critical issue is the definition of the time interval between two periodic assessments in order to maximize the currency of quality metadata.

The two evaluation methods described above differ in the events that trigger evaluation procedures, the granularity of input data, and the improvement methods that can be performed (Table 1).

Table 1: Comparison between on-line and off-line evaluation approaches

	On-line evaluation	Off-line evaluation
Evaluation input	Query submission	Periodically Upon specific events Upon request of data quality administrator
Data granularity	Small amount of data (query results)	Large amount of data (usually, entire d atabases)
Improvement methods	Search for an alternative source	Data or process oriented improvement methods

3.2. Assessment functions

In order to design the assessment module and to guarantee the standardization of quality evaluation procedures, an analysis of data quality dimensions has been performed. The set of quality dimensions that have been considered have been classified as follows:

- *Objective dimensions:* this category includes all dimensions that consider data values without taking into account the process in which data are involved. The most relevant dimensions in this category are accuracy, completeness, and consistency.
- *Architectural dimensions:* this category includes all dimensions that consider the architectural features of the database management system. The only dimension in this category that we consider is data availability.
- *Process dimensions:* this category includes all dimensions that consider the processes in which data are involved. These dimensions are evaluated by considering the end-to-end process, that is all the operations that affect data from their extraction to their delivery. The most relevant dimensions in this category are Relevance, Access Security, Timeliness (including volatility and currency), History, and Cost.

We evaluate these dimensions in a distributed environment, where data may be duplicated in multiple DBs of a single organization or in separate DBs of cooperating organizations. According with this assumption, we have also analyzed the redundancy issue inside an organization and enriched the Quality Factory assessment module with a functionality that can measure the impact of the IS architecture on data quality, called *Architectural Evaluation*. This functionality considers the degree of data

redundancy in the information system and analyzes the degree to which the replication of data can worsen quality due to delays in data updates caused by the periodical realignments of databases. Not all organizations have implemented fully integrated software platforms. Most often, information systems are composed by loosely coupled applications that access their own databases, inevitably causing data redundancy. The same information could be contained in more than one source and can be read and updated from different users at different times. In order to guarantee the alignment and consistency of redundant information, synchronization mechanisms are required. In most cases, data are periodically realigned, but in the time interval between two synchronizations the quality of information is not ensured. Architectural Evaluation provides a model that can help organizations to analyze the impact of software architectural choices on the currency, accuracy, and completeness of data (Cappiello et al. 2004a).

4. Rule-Based Methodology

The architecture performs assessment operations as described above, but it also provides support for the improvement process. The semi-automatic behaviour of the Quality Factory is supported by rules that trigger both the assessment and improvement phases. The rules trigger the interaction among internal modules (*Internal rules)* and between the Quality Factory and the external components, in particular the Quality Control Administrator *(External Rules).*

4.1. Internal Rules

Internal Rules mostly trigger the assessment phase and establish the communication among the modules belonging to the Quality Factory. The Internal Rules are completely automatic and human support is limited to the initialisation of a few parameters. The following sections describe the most relevant rules that manage the interaction between the main modules.

4.1.1. Interaction between the Quality Analyzer and Quality Assessment modules

In the on-line evaluation process, retrieval of data quality values can be performed in two ways, as described in Section 3.1. Let us consider the data answering a query submitted by the user. Corresponding quality val-

ues can be calculated by activating assessment algorithms or can be gathered from the result of the last off-line evaluation stored in the Quality Repository. The Quality Analyzer communicates to the Quality Assessment module the way in which the quality values have to be provided. The discriminating factor between the two approaches is the Computation Time necessary to perform assessment operations. The sum of Computation Time and Delivery Time (*td*), that is the time necessary to send data to the user, must be lower than the acceptable maximum Service Time (*ts*) for the user. Let us consider a user performing a query Q extracting k tuples ($tuple_{i1}$,...$tuple_{ik}$) from a database db_i specifying n requirements on corresponding data quality dimensions qd_j. If tc_j is the estimated computation time of quality dimension qd_j for a single record, the assessment phase must satisfy the following rule:

$$(\sum_{j=1}^{n} tc_j) * k + td < ts \Rightarrow \forall j \in [1,n], \forall z \in [1,k], Perform\_assessment(tuple_{iz}, qd_j)$$

If the rule is not satisfied, then $\forall j \in [1,n]$ and $\forall z \in [1,k]$ the action *Retrieve_quality_values* ($tuple_{iz}$, qd_j) is performed.

4.1.2. Interaction between Quality Assessment module and Monitoring module

The Quality Assessment module and the Monitoring module cooperate to perform the off-line evaluation. The Monitoring module activates the off-line evaluation based on a set of rules that can be classified as Temporal and Functional:

- *Temporal Rules* are designed to assess data periodically. To define temporal rules, it is necessary to establish the evaluation period, that is the time interval between two off-line evaluations. The evaluation period should be defined by considering the update frequencies of source databases. Frequent data changes should be associated with a shorter evaluation period. Let t and t_1 be the current time and the time instant of the last off-line evaluation, respectively, db_i a database, qd_j a quality dimension and Δt_{ij} the evaluation period associated with db_i and qd_j. An example of temporal rule is:

$$t \geq t_1 + \Delta t_{ij} \Rightarrow Perform(offline\_evaluation(db_i, qd_j))$$

Note that the off-line evaluation can be triggered on a subset of a database by specifying a query in the corresponding temporal rule. It can also be triggered on a specific set of quality dimensions with different time periods by defining multiple temporal rules.

- *Functional rules* are defined to capture events occurring as a consequence of a modification in a database. These events are captured by monitoring update, create, and delete operations. Usually, the off-line evaluation is not performed at each modification of a database, due to the high costs of assessment operations and system unavailability. It is more advisable to perform the off-line evaluation after a predefined number of modifications *N*. This number can refer to the number of data updates, creations, and deletions. Alternatively, functional rules can include different thresholds for different types of modifications. In the Quality Factory, modifications are associated with records, as opposed to fields. Therefore, functional rules count a single modification even if multiple fields are changed in the same record. An example of functional rule is:

$$number\_of\_updates(db_i) > N_{update} \Rightarrow Perform(offline\_evaluati\ on(db_i, qd_j))$$

where N_{update} represents the number of update operations triggering the off-line evaluation.

Temporal and functional rules can be combined as follows:

$$(number\_of\_updates(db_i) > N_{update}) \vee (t \geq t_1 + \Delta t_{ij}) \Rightarrow Perform(offline\_evaluation(db_i, qd_j)).$$

In this case, the count of update operations should start from t_1.

Alert messages can also flow from the Quality Assessment to the Monitoring module. The Quality Factory includes a set of rules, classified as *Process Rules,* that support the verification of user quality requirements by the Assessment module. As described in Section 3, the on-line evaluation process implies that if user requirements are not satisfied, the

Assessment module sends an alert message to the Monitoring module. As an example, we can consider a user performing a query Q extracting k tuples ($tuple_{i1},\ldots tuple_{ik}$) from a database db_i. Let us suppose that the user requires data completeness to be equal to 80%. The Assessment module considers the following rule:

$$\forall n \in [1,k],\ if\ Completeness(tuple_{in}) < 80\% \Rightarrow$$
$$Send\_Alert(dimension = completeness\ tuple_{in}\ completeness(tuple_{in}), 80\%).$$

For each tuple that does not satisfy user requirements, an alert message is sent to the Monitoring module specifying the name of the quality dimension that has been assessed, the tuple that does not satisfy requirements, the actual and target values of the quality dimension. The Monitoring module stores all alert messages in a file that can be used for subsequent aggregate analyses supporting the improvement process.

4.2. External Rules

External rules manage the communication between the internal and external components of the Quality Factory. The rules that support the improvement phase are particularly relevant. They trigger the communication between the Monitoring module and the Quality Control Administrator. Note that the improvement phase is not completely automatic. The Quality Factory can suggest the type of improvement actions to undertake and the time period to perform them. The Quality Control Administrator should analyze critical situations and perform the most appropriate improvement actions.

In the previous section, it has been described how the Monitoring module stores alert messages generated by low quality values. When the number of user requests that are not satisfied is high, data should be improved. An overall low quality of data for a high number of users can affect the quality of business services (Redman 1996) and the cost of improvement actions can be justified. When an improvement process is needed, the Monitoring module sends a request to the Quality Control Administrator suggesting a thorough analysis of data. When the user request that extracts k tuples ($tuple_{i1},\ldots tuple_{ik}$) from a database db_i is not satisfied and the alert message is sent to the Monitoring module, the Monitoring module counts the number of alerts on all the detected

tuples and activates the analysis request message considering the following rule:

$$\forall n \in [1,k],\ if\ number\_of\_alerts(dimension = qd\_name\ tuple_{in}) > N_{alert} \Rightarrow Send\_Analysis\_Request(dimension = qd\_name\ tuple_{in}),$$

where N_{alert} represents the number of alert messages indicating a critical quality situation that requires a detailed analysis.

The analysis performed by the Quality Control Administrator should evaluate whether the divergence between user requirements and actual quality values is critical. If it is considered critical, an improvement action is required. As discussed in Section 2, improvement actions are based on either data-oriented or process-oriented techniques. The former are appropriate when data are not modified frequently, as they are expensive and have short-term effects. If data are changed frequently, they need the application of inspection and rework techniques, such as data bashing or data cleaning. It is possible to define rules to support the analysis conducted by the Quality Control Administrator in order to identify the most suitable improvement technique and to activate it. For example, when the Monitoring module sends to the Quality Control Administrator an analysis request for a tuple, the Quality Control Administrator can activate rules to evaluate whether data-oriented tools are suitable for those data and trigger them. If the frequency of updates is calculated within ΔT, a rule activating improvement tools is:

$$\frac{number\_of\_creations(db_i) + number\_of\_deletions(db_i) + number\_of\_updates(db_i)}{\Delta T} < F$$
$$\Rightarrow Activate(data\_oriented\_tool(db_i)),$$

where F measures the critical number of modifications to db_i.

If a database contains critical data and it is frequently accessed by users, process-oriented approaches to data improvement are more appropriate, since they prevent future errors with a long-term effect. If the rule above is not satisfied, that is the frequency of changes is greater than F, then process-oriented improvement initiatives should be activated by the Quality Control Administrator to identify the causes of data errors and eliminate them permanently. This requires the observation of the

processes in which data are involved. Improvement actions change data access and update activities through process analysis and redesign. In our architecture, this type of improvement methods is supported by the History dimension. This dimension tracks the time evolution of the quality of a data set in order to identify which operations have improved or worsened quality values and, thus, build a historical database that can be used for statistical evaluations and process improvement. The History dimension stores all the events that have caused a data modification, from creation to deletion. For each data modification, the name of the user performing the modification, date, hour and type of operation (creation, update or deletion) and the percentage of data quality variation are stored in the History. The analysis of the History file helps the Quality Control Administrator to identify the processes that have to be analyzed and redesigned to obtain a permanent improvement of data quality.

5. Evaluation of the Methodology

The proposed approach has been implemented with a Web interface usable by a common Web browser that allows accessing services, realized as Web services, through an Intranet or the Internet. The architecture has been implemented using JAVA Server Pages (JSP) and JAVA Servlet with Microsoft Access and SQL for data management. In order to provide efficient and scalable Web services, the system has been designed with independent data, application and presentation layers. Not all the quality dimensions listed in the previous section have been implemented. So far, the implementation is complete for timeliness, completeness, history, and architectural evaluation. Current work is focused on the implementation of the accuracy dimension.

The architecture has been tested with the case study of multichannel financial information systems. A sample financial database has been created and data on customer behaviour for the architectural evaluation have been collected with an ad hoc empirical study.

As discussed in the previous sections, quality dimensions can be evaluated in on-line or off-line mode. Testing has pointed out that the on-line evaluation can provide real-time quality values, but if the assessment algorithms are performed on-line on large amounts of data, time response can be excessively high. In this respect, the off-line evaluation is preferable, although it provides quality values valid at the time of the last assessment. Specifically, in multichannel financial institutions users

require low response times, especially for home banking and trading operations. On the other hand, transactions involve databases that contain millions of records. Our analyses have pointed out that for certain quality dimensions, such as timeliness and completeness, the average response time grows linearly, since the execution of the query and the algorithms evaluating quality dimensions are characterized by linear complexity (Cappiello et al. 2004a). On the contrary, for the architectural evaluation dimensions, it is always better to perform the off-line evaluation, as calculations consider all the records in the databases and algorithms are exponentially complex (Cappiello et al. 2004a).

In the test case, the adoption of the rules defined in Section 4 has been extremely useful to manage the results of the architectural evaluation. First of all, the architectural evaluation has allowed us to validate the hypothesis that data redundancy impacts quality (Cappiello et al. 2004a). Further, the automatic monitoring system and the notification of data quality problems to the Quality Control Administrator supports the analysis of the processes and tasks that decrease the quality of data and the definition of the most suitable improvement actions. For instance, using the Quality Factory, we have verified that the data inaccuracy depends in part on the behaviour processes of users, in particular on their pattern of access to financial services. Further, from the results of the architectural evaluation, organizations can understand whether quality problems are caused by the information system architecture itself. A recommended improvement action in this case is the selection of the most suitable synchronization period to limit access to out-of-date data.

Concerning the improvement phase, a fundamental result is that it is not necessary to undertake improvement actions every time that the system identifies a quality problem. It is important to estimate the returns from an improvement action by conducting a cost-benefit analysis. First of all, it is necessary to consider the cost of improvement in terms of expenses related to software tools, hardware, and human resources. On the other hand, it is necessary to estimate the benefits that can be obtained by satisfying requests. The benefit from a satisfied request is equal to the benefit from a satisfied customer that can be calculated on the basis of his/her value in the organization. For instance, in a financial institution the value of the customers is calculated on the basis of their turnover, their financial activities and their switching costs. Finally, the evaluation of the benefits should also consider the total number of requests that the organization will be able to satisfy after the improve-

ment process. In general, the improvement action should be performed only if the benefits can be perceived by a consistent number of users.

6. Conclusions

In this paper, we have presented a rule-based methodology to support the assessment and improvement phases in a data quality program. Future work will complete the implementation of data quality dimensions and other approaches to quality assessment along multiple dimensions will be experimented in order to evaluate the sensitivity of certification results. In addition, since the significant design effort involved by the permanent implementation of a data quality program and the additional management costs organizations necessarily incur over time may represent a barrier to its implementation, these topics will be studied in future research.

Acknowledgments

This work has been partially supported by the Italian FIRB Project MAIS. Particular thanks are expressed to the people involved in the Italian MIUR-MURST-COFIN DaQuinCis project for their assistance and their useful suggestions.

References

BALLOU, D. P. & PAZER, H. L. (1985). Modeling Data and Process Quality in Multi-input, Multi-output Information Systems. *Management Science* 31/2: 150-162.

BALLOU, D. P. et al. (1998). Modelling Information Manufacturing Systems to Determine Information Product Quality. *Management Science* 44/4.

BOVEE, M.; SRIVASTAVA, R.P. & MAK, B. (2001). A Conceptual Framework and Belief-Function Approach to Assessing Overall Information Quality. Proceedings of the Sixth International Conference on Information Quality. Boston, MA: MIT.

CAPPIELLO, C.; FRANCALANCI, C. & PERNICI, B. (2003). Data Quality Assurance in Cooperative Information Systems: a Multi-dimension Quality Certificate. Proceedings of the International Workshop on Data Quality in Cooperative Information Systems, Siena. Rome: Università La Sapienza.

CAPPIELLO, C. et al. (2004a). Time-Related Factors of Data Quality in Multichannel Information Systems. *Journal of Management Information Systems* 20/3: 71-91.

CAPPIELLO, C. et al. (2004b). Representation and certification of data quality on the web. Proceedings of the Ninth International Conference on Information Quality. Boston, MA: MIT.

ENGLISH, L.P. (1999). Improving Data Warehouse and Business Information Quality. New York: John Wiley & Sons.

JARKE, M. et al. (1999). Architecture and Quality in Data Warehouses: an Extended Repository Approach. *Information Systems* 24/3.

NAUMANN, F. & FREYTAG, J.C. (2003). Completeness of Information Sources. Proceedings of the International Workshop on Data Quality in Cooperative Information Systems. Siena. Rome: Università La Sapienza.

NAUMANN, F. (2002). Quality-Driven Query Answering for Integrated Information Systems. LNCS 2261. Berlin: Springer Verlag.

ORR, K. (1998). Data Quality and Systems Theory. *Communications of the ACM* 41/2: 66-71.

REDMAN, T.C. (1996). Data Quality for the Information Age. Boston: Artech House

SCANNAPIECO, M.; PIERCE, E. & PERNICI, B. (2004a). IP-UML: Towards a Methodology for Quality Improvement based on the IP-MAP Framework. Advances in Management Information Systems, Monograph on Information Quality, in press.

SCANNAPIECO, M. et al. (2004b) The DaQuinCIS Architecture: a Platform for Exchanging and Improving Data Quality in Cooperative Information Systems. *Information Systems* 29/7.

SHANKARANARAYAN, G.; WANG, R. Y. & ZIAD, M. (2000). Modeling the Manufacture of an Information Product with IP-MAP. *P*roceedings of the 6th International Conference on Information Quality. Boston, MA: MIT.

WANG, R.Y. (1998). A Product Perspective on Total Data Quality Management. *Communications of the ACM* 41/2.

WANG, R.Y. & STRONG, D.M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. *Journal of Management Information Systems* 12/4.

WAND, Y. & WANG, R.Y. (1996). Anchoring Data Quality Dimensions in Ontological Foundations. *Communications of the ACM* 39/11.