
A Swiss army knife approach to DQ assessments

V. Cotik*, P. Luján, D. Scotton
and D. Yankelevich

Pragma Consultores,
San Martín 575 2 Piso,
(C1004AAK) Buenos Aires, Argentina
Fax: +54-11-4327-1999
E-mail: vcotik@pragmaconsultores.com
E-mail: plujan@pragmaconsultores.com
E-mail: dscotton@pragmaconsultores.com
E-mail: dyankele@pragmaconsultores.com
*Corresponding author

Abstract: The use of tools in data and information quality diagnosis and improvement projects is desirable to automate clerical tasks and in some cases it is mandatory to be effective. In this work we present a toolkit that we developed in the context of the NEAT methodology, that provides a systematic way of assessing Data Quality. The novelty of the proposal is given by the construction of many independent single-purpose tools, which are combined into a powerful kit, and its insertion in the context of a methodology that follows the best practices accepted in the community. We also present the outcome of its use in actual projects.

Keywords: quality; tools; NEAT methodology; data quality toolkit; DQT; Swiss army knife; DQ assessments; information quality.

Reference to this paper should be made as follows: Cotik, V., Luján, P., Scotton, D. and Yankelevich, D. (2007) 'A Swiss army knife approach to DQ assessments', *Int. J. Information Quality*, Vol. 1, No. 2, pp.145–161.

Biographical notes: Viviana Cotik is currently responsible of the Data Quality Practice of Pragma Consultores, a consulting firm specialised in software engineering. She received her Licenciatura (MsC) in Computer Science from Universidad de Buenos Aires (UBA), Argentina, in 2004. She has worked as Teaching Assistant in the Computer Science Department. She has conducted many DQ assessment projects in different vertical markets and she participated in research projects at the university.

Pablo Luján holds a BS in Computer Science from Universidad de Buenos Aires, Argentina, where he was also Teaching Assistant in the Computer Science Department from 2001 to 2004 and where he is currently working in his Licenciatura (MsC) Thesis. His research interests focus on the area of Software Quality. He is a Software Quality Consultant since 2001, currently managing several projects in Pragma Consultores.

Damian Pablo Scotton is currently an undergraduate student at the Universidad de Buenos Aires, Argentina and is expecting to graduate as Engineer in Computer Science. He has been working at the software development team of Pragma Consultores since 2005, where he has participated as Developer of the Data Quality Toolkit (DQT), among others.

Daniel Yankelevich is currently Director of Pragma Consultores. He received his PhD from Università di Pisa, Italy, in 1992, and his Licenciatura (MsC) from ESLAI, Argentina, in 1988. Before working in industrial projects, he was Professor at Universidad de Buenos Aires, Argentina, and held a postdoctoral position in NCSU, USA. He participated in many research projects in Argentina, Italy and the USA. He has more than 40 peer-reviewed publications in conferences and journals.

1 Introduction

Data Quality (DQ) issues are becoming more and more important as companies value their information as one of their most important assets. Billions of dollars are being spent for implementing applications to integrate the companies' corporate data, but these applications often fail because of DQ problems. According to a Gartner Group projection, over 50% of Customer Relationship Management (CRM), Business Intelligence (BI) and Enterprise Resource Planning (ERP) implementations in 2006 will fail or will suffer from limited acceptance because of poor-quality data (Dresner, 2002).

Bad DQ clearly affects decision making. But in many cases, it can also affect the core of the business. For example, let us consider the following case: in Nigeria, a tax dispute related to different oil operators' reserves position was decided based on the quality of the data used to justify the new reserves, according to the Nigerian Presidential Advisor on Petroleum and Energy Matters, Edmund Daukoru (McNaughton, 2004). In this case, the impact of the quality of data in the core of the business is clear.

Finally, the growing interest on corporate governance leads to an increasing examination of corporate controls and legal accountability of officers. A consequence is that security of systems and data becomes critical (Friedman, 2004; Exler, 2002). Poor DQ creates additional difficulties to sustaining compliance with legislative and regulatory laws, such as Sarbanes-Oxley (also known as SarOx compliance), a critical federal law (Griffin, 2006), that establishes new or enhanced standards for all US public company Boards, Management and public accounting firms.

Hence, as previously mentioned by other authors (see for instance Friedman and Kutnick, 2006), there is an evolution from an abstract notion of the importance of DQ in general, since information is considered as an asset, to specific projects where the quality of data can be a driver or a barrier for success, and finally to consider DQ as a critical part of the business.

The success of DQ initiatives relies in the use of a *general methodology* to measure and improve the quality of the data (Friedman and Kutnick, 2006). By *general methodology* we mean a methodology that applies to the complete data life cycle. Given the amount and complexity of datum, the use of tools to detect and correct DQ issues is a must. There are many tools available to do this. In order to get the most out of them, we believe that they should be applied in the context of a general methodology for DQ improvement.

In this paper, we present a family of tools developed by us, which we will call the Data Quality Toolkit or DQT, and its use in the context of the NEAT methodology (Bobrowski et al., 2002), which we have successfully applied in a number of projects.

At a high level, the NEAT methodology has three main phases: diagnosis, treatment and maintenance.

The diagnosis phase has three steps. In the first step we elicit data from data consumers, data custodians and data creators to identify their DQ needs. In order to do that, first we identify those stakeholders, data that is considered critical, data's perceived quality and DQ goals (i.e., the quality dimensions that are priorities for that particular set, organisation and circumstance and that must be addressed in any improvement project (Dravis, 2005; Huang et al., 1999; Redman, 1996; Wang et al., 1998)). Technical and functional documentation, data models and data flows (which provide more objective information) are also analysed. As many publications state (Strong et al., 1997a, 1997b; Wang et al., 1998), each of the three mentioned stakeholders has different knowledge about the targeted systems. Data custodians are aware of some problems with data models, processes to be run to keep data consistent, and user complaints about data and information accessibility. Data creators know the problems they have in order to enter data, such as not knowing what to fill in and which criteria to use in order to do it correctly. Finally, data consumers find or imagine the problems in the quality of the data, such as having more than 50% of a mailing returned because of problems in client addresses. Data consumers may not rely in some subset of the data, may not have access to the data they need, and might also find that different creators are entering data with different criteria.

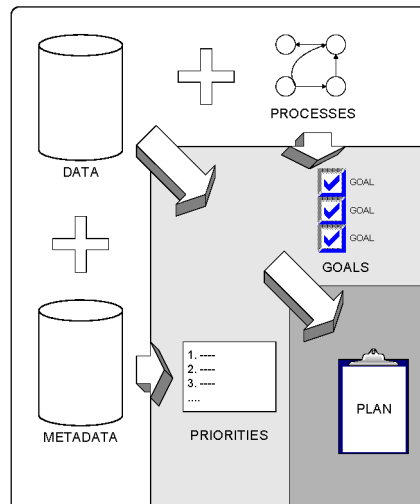
In the second step, our goal is to measure the quality of the data. Since the correctness, completeness and consistency of a data model can also be the cause of some problems, we also measure the quality of the model or metadata. The measurement makes it possible to understand the magnitude of the DQ issues. Metrics are obtained using the Goal Question Metric approach (Basili and Rombach, 1988), a widely used technique for metrics definition in the software engineering community. The goals in this case are the quality dimensions defined above. Then, each goal is defined by means of questions that must be answered to achieve the goal, and finally one or more metrics are defined for each and every question. Once defined, the metrics are implemented and executed. See Bobrowski et al. (1999) for a detailed description of the Goal Question Metric approach for DQ projects.

Metrics can be implemented in different ways. In a relational database SQL queries can be written. In some cases, specific functions have to be implemented (for instance similarity search). Here a DQ tool with query automation features, matching functions and storage and re-execution of metrics can be of great help. Many DQ controls, like the completeness check of different fields and the referential integrity verification, use the same queries but referencing other tables. These queries can be predesigned with some abstraction in mind, so that through the use of parameters different tables and fields can be reached. This allows the tool users to save some time and use it to perform more complex analysis.

Finally, with the results obtained in the two previous steps, an analysis of the cause of the DQ problems follows. This allows us to elaborate a corrective and preventive plan, which enables us to start with the treatment phase. Some of the activities that should be taken in this phase include the correction of data and of the data model, the correction of software bugs, the redefinition and communication of data flows and of the workflow, and user training in the applications and in the importance of the quality of the data. A schema of the NEAT methodology can be seen in Figure 1. It is important to note that data enrichment and correction, where tools are also very helpful, are just a part of the

universe of possible solutions, that usually include working with humans or processes if the real roots of the problems are to be addressed.

Figure 1 The NEAT methodology



Finally, all the improvements, as well as the metrics employed should be maintained in order to keep the quality of the data. Tools should be used to correct data in the treatment phase and to check the evolution in time of DQ metrics during the maintenance phase.

In either diagnosis or improvement of DQ projects there is the need to define different roles associated to specific processes. We usually organise an *assessment team* to conduct the DQ diagnose, treatment and improvement phases. Data custodians (in charge of the quality of the data) must be identified so that they can define and execute metrics to ensure the quality of the data and implement processes to correct it. Finally, data creators, equipped with the appropriate tools, should review the quality of the data immediately after entering it.

The rest of the paper is organised as follows. In the following section, we present the current state of DQ Software tools. While this paper does not provide a complete survey of tools, it provides an idea of the market situation and the variety of tools. Although, they are hard to compare because they serve different goals at different phases. After that, we present our proposal, the DQT. The novelty of the proposal is not the tool itself, but its organisation and its composition of many independent single-purpose tools, which are combined into a powerful kit, and its insertion in the context of a general methodology for DQ improvement. We describe the main functions of the toolkit through examples, and we also present two real-life cases where the tool has been used. The main goals of presenting these experiences are to evaluate the usefulness of the toolkit in actual projects and to show how it has been used in the context of the NEAT methodology. Finally, we present the conclusions and further work.

2 Data Quality (DQ) software tools

DQ Tools can be classified into several categories, according to their goals (Cotik, 2003; Friedman and Bitterer, 2006) (see Table 1). Some of them are *cleansing* or *analysis tools*, where parsing, standardisation, data enrichment and matching functions are implemented; *consolidation* or *integration tools*, that deduplicate data; *extraction tools*, that extract data from systems that are already in the production phase in order to correct them; *data profiling tools*, that provide for the analysis of the content and structure of the data and try to obtain model information from the data and *integrity tools*, that control referential integrity, among others.

Table 1 Different DQ tools categories, their description and examples of actual and more important DQ tools providers

<i>Category</i>	<i>Description</i>	<i>Examples Tool (company)</i>
Cleansing	Parsing (identifies and isolates individual elements in the information repository) Standardisation (converts data into an appropriate format for matching) Data enrichment (data enrichment based on available data, e.g., gender from name) Matching (identifies similar data, the objective is to know if data refer to the same or not). There exist different matching functions as soundexing, keyboard distance, character frequency, and so on	dfPower® (DataFlux) Trillium® (Trillium Software) DataRight® (Firstlogic, now Business Objects) DataLever® DQT (our tool)
Consolidation/ integration	Once duplicated information is found, this kind of tools allows us to decide which the correct data is and in some cases it may suggest which the correct value is. The incorrect data may be eliminated. They usually have the capability of analysing different data sources	dfPower® (DataFlux) Match & Consolidate® (Firstlogic, now Business Objects) DQT (our tool) (partially)
Extraction	Data is extracted from productive systems for being corrected	dfPower® (DataFlux)
Data profiling	The contents and structure of the data is analysed in detail. Information of the data model tries to be obtained from the data. The referential integrity, entity and column integrity is checked	Trillium® (Trillium Software) IQ insight® (Firstlogic, now Business Objects) DataLever® DQT (our tool)

Tools can be further classified into detection, correction and prevention tools, depending on their functionalities. Detection tools detect problems in the databases; correction tools correct problems and preventive tools detect problems when the user is entering bad quality data.

DQ software tools also differ in the data they can handle. For instance, most of the commercial tools are prepared to deal with householding data, but there also exist DQ tools specialised in specific businesses or vertical markets. Since in many cases it is not possible to deal with the semantics of the data, they use heuristics, usually in the form of knowledge bases, mostly in English language. DQ tools differ in their capabilities to customise the rules and in the presence of the functionalities listed above.

In a survey conducted in the past (Cotik, 2003) we came across very powerful DQ tools such as Firstlogic's (now Business Objects), DataFlux's and Trillium's. According to a recent report from Gartner Group, these three and IBM continue to be the market leaders (Friedman and Bitterer, 2006). But, although some of them were good at parsing, name and address corrections, they were not useful for our needs as a consulting company because the kind of data they were prepared to deal with (mainly names and addresses) was different from the data managed by many of our clients of different vertical markets (Oil & Gas and Telcos, among them). Furthermore, the tools were not adapted to work in Spanish. Although they can be adapted, this task requires an important amount of work. Finally, the cost of these tools was much higher than what most of our interested clients could afford at the moment.

These facts led us to decide the creation of our own DQ tool as one of our companies' Research and Development Area projects. The development was partially funded by the Government of the City of Buenos Aires, Argentina.

3 DQT

Instead of building a large, integrated, powerful and expensive tool, with heavy investment on the development of the human interface: we followed the strategy of building several simple tools that implemented single techniques that would be useful in different situations. These tools could be combined, but are meant to be used by consultants who would first define the strategy and then choose the tools accordingly. The notion would be that of a Swiss army knife for DQ: a naïve user could do very little with a Swiss army knife, but a trained consultant would probably choose the right tools for each specific problem. This is why we called it a 'toolkit'.

The coherence of the toolkit would be given by the underlying methodology and adequate training.

The use of the toolkit allows us to add quickly and inexpensively new functionalities as needed. When some functionalities were needed during the projects, we could add them very easily, without a substantial development effort. However, the use of the toolkit demands more manual work of an IT skilled professional than what a *larger, more integrated, more powerful and more expensive tool* would. Still the toolkit does not require an initial configuration by a specialist.

The DQT is a detection tool. It does not modify data, but it has the capability to generate SQL scripts to do so.

It has many modules, including standardisation and matching for Spanish and English languages, enrichment, integrity controls, domain-specific controls and geographical information checks. It analyses the information in a syntactic and semantic way and facilitates the finding of problems in the data model and the execution of queries to a database. Metrics are obtained from the executions and bad quality data can be seen in

detail. The executed queries and its results are stored and can be easily re-executed. Statistics of the evolution of the quality of data in time can thus be obtained.

The DQT can analyse data from different relational database management systems, through an ODBC connection and runs on Windows®. It uses a knowledge base as an oracle to control the correctness of some data and to enrich data. This knowledge base can be populated with information from different domains.

The tool is intended to be used by DQ Assessment Teams to create and execute metrics to measure the quality of the data at a specific point in time and by the Management to follow the status of the quality of the data. An additional module is being constructed to enable an automatic re-execution of metrics for Data Custodians.

Figure 2 shows the phases of the NEAT methodology and in which steps the tool DQT can be applied.

Next, we describe the main functions of the toolkit with the aim to show what a consultant could afford with the combination of the different simple modules of the toolkit.

Figure 2 The use of DQT in the context of NEAT



3.1 Data standardisation and data matching: an example and results

The DQT has a library of *matching* functions to identify redundant data in a data set, as soundex, a phonetic algorithm that takes a term as input and produces a string that identifies a set of words that are phonetically alike (Knuth, 1998), for English and Spanish, distance between texts and n -grams. It also has parsing and standardisation functionalities (see Table 1).

The standardisation module converts data into a standard value. For example, the values:

- ‘*eighteenth st.*’, ‘*eighteenth street*’, ‘*18th st.*’ and ‘*18th st.*’, represent the same street
- ‘*ESP*’, ‘*Electrical Submersible Pump*’, ‘*electrosubmergible pump*’ represent the same component.

Each of these values has the same meaning but is represented differently. The DQT knowledge base has information of standards for different domains, such as countries, cities and numbers. It also has the capability of storing standards in different languages. New standard definitions can be incorporated to the knowledge base. With this module the first set of data can be standardised to the value *18th street*. Standards are not easy to obtain, but many of them are more available than what one could think. Catalogues and directories might be sources of standardised data.

We will show an example of DQT main standardisation and data matching functionalities using a subset of a real-case clients table from a company. For the example we take the most significant records of clients living in Buenos Aires city (Ciudad de Buenos Aires), formerly known as Capital Federal.

Using the frequency count functionality (that groups similar concepts, and counts the amount of occurrences of them), we grouped the *primary city* field of the clients table (see Table 2).

Table 2 Clients table grouped by primary city

<i>Primary city</i>	<i>Count</i>	<i>Primary city</i>	<i>Count</i>
BS.AS	1	CAP.FEDERAL-PTO MADE	1
BUENO AIRES	3	CAPFED	1
BUENOES AIRES	1	CAPI	1
BUENOS AIRES	1	CAPITAL	2
BUENOS AIRES	152	CAPITAL FEDERAL	1
BUENOS AIRES – CAPIT	1	CAPITAL FED	1
BUENOS AIRES – CAPITAL FEDERAL	1	CAPITAL FEDERAL	33
BUENOS ARIES	3	CAPITAL FEDERAL – BUENO AIRES	1
CAP FED	35	CAPITAL FEDERAL – BUENOS AIRES	1
CAP FED REP	1	CAPITAL FEDRAL	1
CAP FED.	1	CAPITOL FEDERAL	1
CAP FEDERAL	1	CIUDAD DE BUENOS AIR	2
CAP. FED.	14	CIUDAD DE BUENOS AIRES	1
CAP.FED.	5	CPAITAL FEDERAL	1
		<i>Total</i>	268

In this case we can notice at a glance, that all the data correspond to the same city. If we were analysing data from the whole country, the whole world or a more difficult domain the task would be more time consuming. Our goal is to have the capability of correcting the data automatically or semiautomatically with the use of standardisation and matching functions.

We first use the standardisation function. This function standardises data based on the knowledge stored in the tools knowledge base. The knowledge base entries used in this example can be seen in Table 3. The function also deletes redundant blanks and deals with other punctuation marks.

The knowledge base can be modified, what allows users to enrich it in order to improve the obtained results. Once we ran the standardisation function, we obtained the result shown in Table 4. We can notice that although the results have improved, we still have many different values representing the same city.

Table 3 Knowledge base entries for cities. Standardisation for Buenos Aires

<i>Data</i>	<i>Standard</i>	<i>Data</i>	<i>Standard</i>
BS AS	BUENOS AIRES	CAP FEDERAL	BUENOS AIRES
BS AS.	BUENOS AIRES	CAP. FED	BUENOS AIRES
BS. AS	BUENOS AIRES	CAP. FED.	BUENOS AIRES
BS. AS.	BUENOS AIRES	CAP.FED	BUENOS AIRES
BS.AS	BUENOS AIRES	CAP.FED.	BUENOS AIRES
BSAS	BUENOS AIRES	CAPFED	BUENOS AIRES
BSAS.	BUENOS AIRES	CAPFED.	BUENOS AIRES
BUENOS AIRES	BUENOS AIRES	CAPITAL FED	BUENOS AIRES
CAP FED	BUENOS AIRES	CAPITAL FEDERAL	BUENOS AIRES
CAP FED.	BUENOS AIRES	CIUDAD DE BUENOS AIRES	BUENOS AIRES

Table 4 Standard values

<i>Standardised primary city</i>	<i>Count</i>
BUENO AIRES	3
BUENOES AIRES	1
BUENOS AIRES	247
BUENOS AIRES – BUENO AIRES	1
BUENOS AIRES – BUENOS AIRES	1
BUENOS AIRES – CAPIT	1
BUENOS AIRES REP	1
BUENOS ARIES	3
CAP.FEDERAL-PTO MADE	1
CAPI	1
CAPITAL	2
CAPITAL FEDERAL – BUENOS AIRES	1
CAPITAL FEDRAL	1
CAPITOL FEDERAL	1
CIUDAD DE BUENOS AIR	2
CPAITAL FEDERAL	1
<i>Total</i>	<i>268</i>

Now, the soundex functionality is applied to the standardised values obtaining the results shown in Table 5. This function enables us to index phrases according to their pronunciation. The DQT has an English and a Spanish soundex implementation.

In this way, we detected 255 rows that correspond to the same city (the ones with soundex code B526), that is 95.1% of true positives. There are no false positives and only 4.9% false negatives. These numbers depend on the completeness of the knowledge base and on the terms domains.

Table 5 Standardised primary city, with its soundex code. Records having the same soundex code are highlighted

<i>Standardised primary city</i>	<i>Soundex code</i>	<i>Count</i>
BUENO AIRES	B562	3
BUENOES AIRES	B526	1
BUENOS AIRES	B526	247
BUENOS AIRES – BUENO AIRES	B526	1
BUENOS AIRES – BUENOS AIRES	B526	1
BUENOS AIRES – CAPIT	B526	1
BUENOS AIRES REP	B526	1
BUENOS ARIES	B526	3
CAP.FEDERAL-PTO MADE	C136	1
CAPI	C1	1
CAPITAL	C134	2
CAPITAL FEDERAL – BUENOS AIRES	C134	1
CAPITAL FEDRAL	C134	1
CAPITOL FEDERAL	C134	1
CIUDAD DE BUENOS AIR	C331	2
CPAITAL FEDERAL	C134	1
<i>Total</i>		<i>268</i>

The process can be iterated more than once. Note that the soundex code C134 is repeated in many records. If we normalise the set, changing the C134 soundex code to Capital Federal, and re-execute the standardisation function, we obtain a new data set with only a 2.6% of false negatives (six more records are identified repeating the procedure once more). Actually, it is possible to automatically generate a corrective SQL script that modifies the values of the Primary City field having the same soundex, replacing them with the most frequent standardised value, or the one that the user selects as default (see Table 6). This technique is useful in many practical cases, as can be seen from the example.

Table 6 Corrected values

<i>Primary city</i>	<i>Count</i>
BUENO AIRES	3
BUENOS AIRES	255
CAP.FEDERAL-PTO MADE	1
CAPI	1
CAPITAL FEDERAL	6
CIUDAD DE BUENOS AIR	2
<i>Total</i>	<i>268</i>

The code B526 was replaced by the most frequent value for this soundex code and the code C134 by the value ‘CAPITAL FEDERAL’. Now reexecuting the standardisation function, Capital Federal could be changed to Buenos Aires, leading to 261 True Positives (in the table we highlighted the records that would be changed to BUENOS AIRES).

Even if it would be possible to automate or semiautomate the iterative process (for instance, looking for a ‘fixed point’), human input and domain knowledge is still key, and we rather adhere to the concept of toolkit: the human using the tool should decide whether to apply the same procedure a number of times or if it is time to apply a different algorithm or technique.

3.2 Integrity control, enrichment and other validation rules

The integrity control module has three main functionalities: referential integrity check, attribute check and primary key candidate check.

The referential integrity check detects, given two tables, if there are referential integrity problems (i.e., if the foreign keys of the *child table* are not primary keys of the *parent table*). This problem would lead to having orphan records in the *child table*. For example, all the owners of savings accounts of a bank should be clients of the bank. The referential integrity function can check whether there are some clients in the saving accounts table who do not belong to the clients table.

The attribute check provides for the detection of null values on one or more attributes of a database table and checks whether there are values that do not belong to a defined list, among others. As a result of the first check, we obtain:

- a metric informing the amount of null values/the amount of table records
- records that have null values
- SQL code executed.

As happens with every other function, the control and the metric results can be saved in the DQT database, with a user chosen name.

The primary key candidate check detects if some attributes of a table could be primary key (a set of attributes is a primary key candidate if it is unique and if none of its values are null). With this function we can easily detect suspicious data, for instance duplicated tax ids, in the cases where this attribute should unambiguously identify a person.

The enrichment module allows to augment and to enhance the database data. This is done with data taken from internal or external sources. Some functions of the DQT enable the user to derive first names, last names and gender from the full name field. In all these cases, the DQT generates a SQL script to add this new information to the database. This module has been used to determine the gender of each of our companies’ 260 employees in order to send women greetings on the international women day. We had only 0.8 % bad results that corresponded to French and Bosnian names. This module is based on the knowledge base content. For some names like Andrea, used for men in Italy and for women in Argentina, the results are not precise. To solve this, the DQT determines which gender is more commonly used for that name and suggests it as a ‘possible correct’ gender.

3.3 Ad hoc algorithms

For some domains, domain knowledge can be captured in specific algorithms, in order to validate and populate data sets. Two conditions should be fulfilled by these algorithms to be effective: first, they should implement heuristics, usually strongly

related to the domain and second, they should approach problems shared by many organisations/applications in the domain.

We have identified and included in the DQT a number of such algorithms. Among them, an algorithm to validate well-names in the Oil Industry. Their names are usually strongly regulated and must follow a strict structure, in most cases subsuming information that must be coherent with data describing the well.

Other examples are algorithms used to assess the correctness of tax id numbers in Chile (called RUTs) and Argentina (called CUITs). The CUIT is also an example of an identifier that subsumes information: it can be calculated from the number of the identity card of the citizen plus his/her gender. Hence, in a data set containing gender, CUIT and identity card number the algorithm can be applied to identify mismatches or to populate the table. Moreover, an informed user can weigh the different columns and decide whether the mismatch corresponds to an incorrect CUIT, identity card or gender. The DQT function that, given a name, guesses its gender can be combined with the CUIT function to perform more sophisticated validations.

3.4 *Knowledge base*

An alternative way to capture domain knowledge is by using the Knowledge Base. The DQT provides a function that allows a user to check the values on a data set against a table of the Knowledge Base. For instance, in the example above, one could check the Primary City against a table of Cities of Argentina of the Knowledge Base, to find that CPAITAL FEDERAL and CAPI (among others) are not valid cities. The Knowledge Base is dynamic and grows with the use of the toolkit and when standardised information becomes available. For instance, we bought a database with all the location information of the City of Buenos Aires (streets, numbers and geographical information) and included it in our Knowledge Base. These data could be easily used to validate addresses from a database and even to suggest candidates when addresses are identified as non-existent or wrong.

3.5 *Georeferenced data analysis*

Many industries, in particular Oil & Gas and Telecommunications, have sensitive geographical information. Having the capability of ensuring the detection of problems in the integrity of that data can be of great value.

The DQT Knowledge Base supports geographical information. Thus, it is possible to represent districts such as countries, provinces and neighbourhoods using polygons meaning the cartographic coordinates. Therefore, these polygons can show whether a set of records with latitude and longitude fields are in or outside a district.

Based on this, the DQT can detect inconsistencies in geographical data (for instance between Province field and Latitude and Longitude fields).

It also represents the record location drawn on a map, allowing the user to detect visual inconsistencies.

Figure 3 shows a Microsoft© Live Local© map representing part of two Argentina's provinces, Chubut and Santa Cruz. The pushpins are examples of some Chubut's oil wells, stored as records in a database using Latitude and Longitude fields (see Table 7). The black ones are offshore wells, and the white and grey ones are onshore wells. By representing the wells on a map, it is easy to locate incorrect data (Wells 3 and 5).

Additionally, if the GIS Knowledge Base database stores the polygons of Chubut and Santa Cruz borders, by querying the database the output of Well 6 coordinates would be Santa Cruz, although the well belongs to the Chubut district.

Figure 3 Table 7 wells plotted in a Microsoft live local map

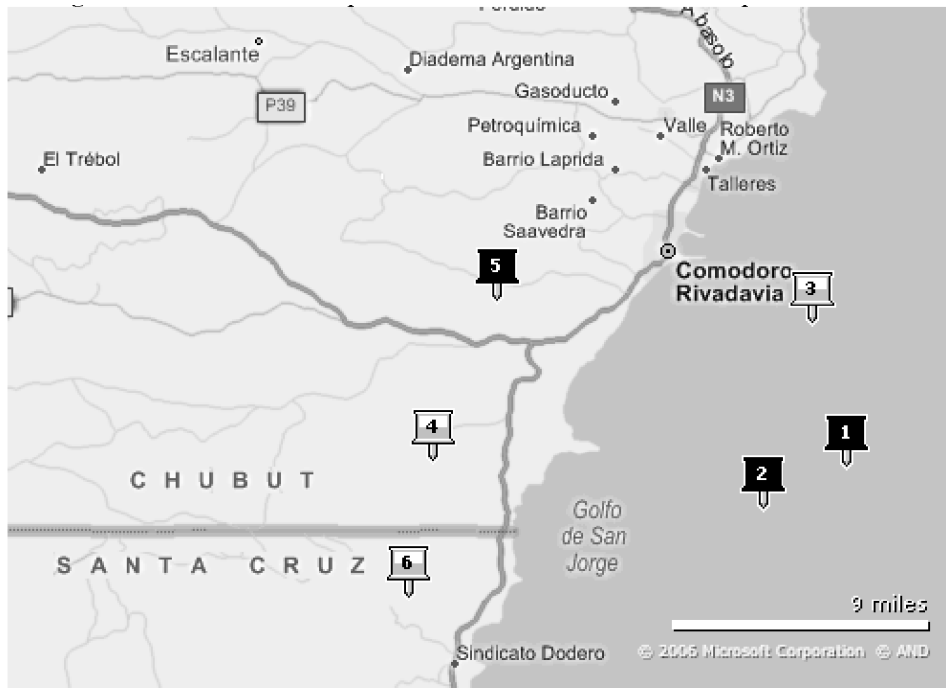


Table 7 Wells, well type and province

Well name	Well type	Well province	Latitude	Longitude
1	Offshore	Chubut	-45.96890833	-67.35483611
2	Offshore	Chubut	-45.98957222	-67.41477222
3	Onshore	Chubut	-45.89564444	-67.37914167
4	Onshore	Chubut	-45.95831667	-67.65584167
5	Offshore	Chubut	-45.88393889	-67.60993889
6	Onshore	Chubut	-46.03311389	-67.67387778

The database engine chosen for the implementation was Postgres. Although GIS packages are a common feature among many other databases, the PostGIS plug in is both easy to install and quick to start working with. The database has to be loaded with polygons and names of the districts.

The map drawing functionality provided by Microsoft Live Local (<http://local.live.com>) and Google Maps (<http://maps.google.com>) enable a friendly way to visualise a point or set of points. Just clicking the option menu in the DQT, a form that launches a browser window showing the results will be displayed.

Some of the limitations of the tool, that we are planning to overcome as a future work, are following; it is not prepared to work with a very big amount of data, as heavy production data from an important bank; in order to compare or check data from two different databases the user has to do some manual tasks and although the needed information for the execution of statistics is gathered, statistic functions have not been implemented yet. Finally, the knowledge base has to be populated with data and its quality should be revised on a periodic basis.

4 DQT use: case studies

The DQT has already been used in several projects. The feedback from each experience was used to enhance the toolkit functionalities according to the detected needs of use. We will briefly describe two experience reports where the tool has been successfully used.

4.1 Case 1

Case 1 was a DQ Assessment project we did for one of the dependencies of a South American Government. The dependency worked with a register of the training actions taken by the employees of this country's companies. So the main data they use consists of course names, companies and employees data.

The DQT was used in the measurement step of the NEAT methodology. We found several advantages of using it, among them:

- It took us less time to develop queries to the database.
- All queries and metrics results were stored in the DQT database. This allowed us to repeat execution (in a sort of 'regression test' for DQ, to find an analogy) and to elaborate statistics.
- The DQT countries knowledge base, helped us discover some non-existent countries that populated the clients database (such as *Britanica*) very easily.
- The integrity control helped us discover countries that were repeated in the countries parameter table, a problem that led statistics to have erroneous results.
- Different people with the same tax ids, duplicated entries of the same person, empty e-mail addresses, are some other problems we found by using the entity and matching functions.

Besides the obvious advantage of automating clerical tasks and making the diagnosis much faster (the country matching is an example), we also find very useful the ability to repeat tasks with almost no effort.

4.2 Case 2

Case 2 is being executed in an Oil & Gas company, as a part of the migration project of a legacy drilling information management system to a new one. This migration includes an important change in the database model, functionalities and data display.

Oil companies usually have different drilling groups, organised in many ‘operations’. In particular, this company has many operations in different countries, each with different loading criteria, and it is part of the project to unify the data of all the operations. This involves data integration from different sources and with different criteria. Usual problems, such as semantic shifts (when the same object means different things to different people), semantic gaps (when the gap between two descriptions of an object by different representations is not uniformly solved), mismatches (because of the structure change) were found during this integration. Besides that, major changes in the database model, such as normalising data that was not normalised, made it necessary to correct the quality of some critical data before the migration.

Even though some queries have been previously prepared for checking the quality of the data, the additional matching and standardisation functionalities of the DQT helped us detect further problems. Critical values for which parameter tables existed were checked against these parameter tables (by including them as tables of the Knowledge Base). Finally, DQ controls were stored and re-executed, once the data had been corrected, to know if it really was. Concurrent access to the DQT database allowed the whole team to use the elaborated queries.

In this case, the main contribution of the DQT was to help to make more systematic the controls and tasks related to DQ. While it is possible to be systematic without the use of a tool, the functions mentioned would have required a lot of effort from the group that could be directed to more useful tasks.

5 Conclusions

As an alternative to the use of integrated DQ Tools, we propose the use of a toolkit that combines many single tools, as if it was a Swiss army knife for DQ professionals. The coherence in the use of the tool is given by the adherence to a well-defined methodology, which follows the best practices accepted in the community (Wang et al., 1996).

The DQT kit includes Data Standardisation and Data Matching functions, Integrity Control, Enrichment, Georeferenced data analysis and also ad hoc algorithms for specific domains. The use of a Knowledge Base allows us to continuously improve the tool – without further coding and to capture knowledge from different trusted sources to be used as oracles.

The DQT was used in many real-life diagnosis projects, where the level of quality of production data (data of systems in the production state) was assessed for companies in different vertical markets and Government organisations.

From these projects, we have learnt that the use of the right tool at the right time can be very helpful for detecting an automatic or semiautomatic way DQ issues faster. We have also learnt that the use of a tool in isolation from a general methodology does not yield good results if the roots of the problems are to be addressed. That is, the use of a tool without the context of a methodology that applies to the whole data life cycle rarely addresses the deeper causes of the DQ problems and tends to concentrate only on symptoms.

The flexible structure of the toolkit also allowed us to aim for organic growth. When a new algorithm or technique is developed, it can be included as an independent ‘blade’ of the Swiss army knife: as a stand-alone tool that will interact with the other tools but also it could be used on its own. Some of the tools are of no use in a particular domain or project, while in others they could be critical. This characteristic allowed us to add quickly and inexpensively new functionalities even as we were using the toolkit in different projects. Although the use of the toolkit demands some manual work of an IT-skilled professional, its deployment is much easier and faster than the one of a *larger, more integrated, more powerful and more expensive tool* and it does not require an initial configuration by a specialist. Moreover, the overall methodology provides for a context of use which is crucial: without the big picture, one could easily be lost in the use of detailed techniques without understanding the goal of the specific phase one is following. The improvement of the quality of data can not be obtained just by applying sophisticated matching algorithms. We must target the overall process of the data life cycle, and the diagnosis phase can guide and simplify the improvement of data over time.

Summarising, the toolkit has many of the features that according to Friedman and Bitterer (2006) should be evaluated when choosing a DQ tool:

- domain-agnostic DQ, for managing different kinds of non-customer data
- international data format support, required for global companies
- ease of implementation and usability
- completeness of functionality and a single tool that can handle various DQ activities such as profiling, matching and cleansing.

6 Further work

We are planning to improve the toolkit by means of the organic growth model we have already described. We are aiming at improving the Knowledge Base, and we have a number of new features that will be developed in parallel. One of the improvements under analysis is to provide functions to enable the Data Custodians to automatically re-execute metrics. This would allow the users to control the quality of the data with the previously defined metrics on a periodic base without an additional effort. The module would show results ordered by data and dimension. We are also planning statistics and visualisation modules. The statistics module would allow the users to keep track of trends over periods of time, which could enhance their analysis capabilities of DQ root causes. Visual display of data has been identified as a powerful way to find errors and to improve quality (Tufte, 1983) and we will build on this by using graphical displays to find error candidates or to quickly analyse large amounts of data in a semiautomatic way. A more automated connection and analysis module between different databases and the treatment of bigger amount of data are also being considered as future improvements.

The use of the tool in new projects and domains will probably enrich the experience and foster the development of new ‘blades’ or modules to the toolkit.

Acknowledgements

We want to thank Juan José Cukier for his help with an early version of this paper and the anonymous referees for their helpful comments.

Work partially supported by Grant 70-097/2004 from the Government of the City of Buenos Aires.

References

- Basili, V. and Rombach, H. (1988) 'The TAME project: towards improvement-oriented software environments', *IEEE Transactions on Software Engineering*, Vol. 16, No. 6, pp.758–773.
- Bobrowski, M., Marré, M. and Yankelevich, D. (1999) 'A homogeneous framework to measure data quality', *Proceedings of IQ'99*, Boston.
- Bobrowski, M., Marré, M. and Yankelevich, D. (2002) 'A NEAT approach for data quality assessment', in Piattini, M., Calero, C. and Genero, M. (Eds.): *Information and Database Quality*, Chapter 7, Kluwer, p.1.
- Cotik, V. (2003) *Survey de Herramientas y de Datos Disponibles en la Región*, Technical Report, Pragma Consultores, Buenos Aires, Argentina.
- Dravis, F. (2005) 'Why categorize data quality problems?', *Business Objects Data Quality Weblog*, <http://eimblog.businessobjects.com/dravis/2005/8/31/why-categorize-data-quality-problems.html>.
- Dresner, H. (2002) 'Implementing business intelligence to succeed, not fail', *Strategic Planning*, SPA-18-8622, Gartner Group, http://www.gartner.com/DisplayDocument?ref=g_search&id=381068&subref=simplesearch.
- Exler, R. (2002) *Sarbanes-Oxley: Corporate Governance. IT Headache?*, Research Note, Robert Frances Group, <http://www.rfgonline.com/reprints/ibm/112702nt.html>.
- Friedman, T. (2004) *Data Quality: More Important than Ever*, Gartner Group, http://weblog.gartner.com/weblog/blog_rss.php?blogid=11&itemid=469.
- Friedman, T. and Bitterer, A. (2006) *Magic Quadrant for Data Quality Tools*, Gartner Group, http://www.gartner.com/DisplayDocument?ref=g_search&id=491510&subref=simplesearch.
- Friedman, T. and Kutnick, D. (2006) 'The cost of poor data quality', *Gartner Group Online Podcast*, http://www.gartner.com/it/products/podcasting/asset_145611_2575.jsp.
- Griffin, J. (2006) 'Information strategy: improve data quality', *DM Review Magazine*, http://www.dmreview.com/article_sub.cfm?articleId=1044284.
- Huang, K., Lee, Y. and Wang, R. (1999) *Quality Information and Knowledge*, Prentice-Hall, Upper Saddle River, NJ, USA.
- Knuth, D. (1998) *The Art of Computer Programming*, Addison-Wesley, Vol. 3, Boston.
- McNaughton, N. (2004) 'From data to financials – a small step indeed', *Oil IT Journal*, http://www.oilit.com/2journal/2article/0402_3.htm.
- Redman, T. (1996) *Data Quality for the Information Age*, Artech House, Norwood, MA, USA.
- Strong, D., Lee, Y. and Wang, R. (1997a) '10 potholes in the road of information quality', *IEEE Computer*, Vol. 30, No. 8, August, pp.38–46.
- Strong, D., Lee, Y. and Wang, R. (1997b) 'Data quality in context', *Communications of the ACM*, May, Vol. 40, No. 5, pp.103–110.
- Tufte, E. (1983) *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, CT, USA.
- Wang, R., Lee, Y., Pipino, L. and Strong, D. (1998) 'Manage your information as a product', *Sloan Management Review*, Vol. 4, pp.95–105.
- Wang, R., Strong, D. and Guarascio, L. (1996) 'Beyond accuracy: what data quality means to data consumers', *Total Data Quality Management Program*, Vol. 4, No. 4, p.5.