Practical data quality certification: model, architecture, and experiences

Paolo Missier
School of Computer Science
The University of Manchester
Manchester, UK
pmissier@cs.man.ac.uk

Alessandro Oliaro
Department of Mathematics
Università di Torino
Torino, Italy
oliaro@dm.unito.it

Silvana Raffa
Consorzio per il Sistema
Informativo – Piemonte
Torino, Italy
silvana.raffa@csi.it

ABSTRACT

Queries to information systems return results whose quality is not described explicitly . We argue that data providers have both the incentives and the technical means to make quality of data explicit, and that data consumers may exploit such metadata to reach informed data acceptability decisions. We propose a data model and architecture for addressing this problem in a practical and scalable way, and report on our experience with a prototype implementation for a real industrial use case.

1. INTRODUCTION

The proper execution of data-intensive business processes relies upon the exchange of data among interconnected information systems. When data that is created and maintained by original sources propagates through downstream systems, so does its quality, in particular for data that is central to a number of applications. In the public sector, for example, vital records about local businesses are likely to be joined with related information in several other databases in order to support dedicated applications (eg for internal revenues management). In this case, poor quality of the central vital records may result in incorrect tax assessments, for example.

A number of approaches and methodologies, some of them quite successful, have been proposed for improving the quality levels of the data; it is not the goal of this work to recall or add to such collection. Rather, we note that even after cleaning has been performed on a data source, the problem remains that other systems which depend on that data are completely unaware of its computed quality levels. We argue that, in contexts like cooperative data exchange, or data marketing, providing explicit quality meta data alongside the data would benefit both provider and consumers.

Data exchanges in cooperative systems are usually defined as part of some over-arching process that crosses the individual system's interfaces. In this non-competitive scenario, data providers and consumers share a common interest in preventing process failures by supporting quality-aware data exchanges. Attempts have previously been made to design a brokering infrastructure to support quality-aware data exchange in cooperative systems [9, 10]. The same authors also sketched a data model for presenting quality metadata to the consumer, called a *quality certificate* [4]. To the best of our knowledge, however, there is no documentation of this model being put to the test on real use cases.

With data marketing, information is viewed as an asset with an associated market price, which can be sold to third parties for profit. In this case, the incentive for data providers to make quality assessments available comes from the market. As proposed in [3], for example, it is possible to link the price of goods to their quality, in such a way that uncertainty in quality determines a reduction in price, according to a contingency pricing scheme. Specifically, it is argued that the price of information can be a function of quality, provided that perceived quality is "objectively verifiable". In practice, "the contingency pricing framework works well when the information infrastructure enables easy quantification, capture, verification and dissemination of quality and performance information". Throughout their exposition, however, the authors leave the elements that make up quality meta-information unspecified –as expected of a general framework.

The work presented in this paper is motivated by the need to make such quality meta-information available to consumers in a precise and actionable manner. With traditional goods, such as food, legislation has been put in place to make sure that appropriate quality controls are applied to the production process in a documented and standard way; this results in labels being available to the consumer, eg carrying the expiry date of the product. As a result, when consumers buy food they effectively decide to trust this certification process, the labels being their only clue to quality.[1]

We argue that a similar certification process should be developed for information goods, to achieve a win-win scenario: data consumers should be provided with sufficient and fair quality information, so that sensible data acceptability decisions can be reached at the time data is obtained. At

[1]Of course, brand reputation also plays an important role.

IQIS 2006, June 30th, 2006, Chicago, IL, US

the same time, the provider benefits in terms of reputation and credibility. However, it should be clear that we are not advocating certification-by-legislation; since computing quality metadata is generally regarded as expensive, data providers should balance this cost with expected business benefits, and freely provide any type of quality metadata that they perceive will bring added value to their data.

One way to address the problem could be to augment a query language like SQL with features to specify constraints on quality metrics of interest. This is the approach taken for example in [2], where a novel framework for negotiating quality contracts is supported by extending a quality of service modelling language, QML [6]. This approach, however, assumes that users be aware of the available quality metrics, and that they have a realistic expectation about their values for the specific data resulting from a query. In this paper, we propose a practical way to convey this information to the users, so that they can pose reasonable follow-up, quality-aware queries.

Firstly, we propose a simple semi-structured model for sharing quality meta-information between data providers and consumers. This model accommodates a provider-defined collection of *quality indicators*, which provide insight into the expected quality of the delivered information. These may include measures of data completeness, accuracy, consistency, and so forth, according to established classifications for quality properties, or *dimensions* [13]. Indicators are often data-specific, for instance the expected average syntactic correctness of a set of business addresses.

While indicators are useful to convey the available quality meta-information about data, they are often expressed at a low level and may be difficult for users to understand; furthermore, they do not, by themselves, provide criteria for data acceptability, often summarized by the term "fitness for use" in the literature. Thus, our model also includes a way to specify a collection of user-defined, domain-specific functions that express the user's perception of data quality in a more intuitive way, by mapping indicators to a collection of *quality ratings*. The specific rating functions used in the prototype, described in detail in Section 4, are modelled as decision trees: each node in the tree evaluates an expression on the value of a single indicator, for instance the estimated syntactic correctness of an address; the outcome determines a partial rating and indicates which node is visited next further down the tree. A formula is used to assign an overall rating when one of the leaves is reached. Note that quality ratings have been used in the past, for instance in the context of data integration of heterogeneous and possible mutally inconsistent data sources [11, 1].

The primary original contribution of our work is in the definition of a data architecture for the implementation and delivery of quality metadata, and the practical demonstration of its feasibility. The user interaction model can be summarized as follows:

- When users issue queries to the provider, the query processor returns both the query result, and a collection of quality indicator values associated to each data item in the result;
- Users may request that one or more quality rating models of their choice be applied to the indicators;
- In order to make the quality meta-information practically manageable for large result sets, the provider computes a summary aggregate of the indicators and of the rating values, called data *summary quality profiles*;
- Along with the indicator values, the provider also returns a description of the types of indicators, which includes details on how their values are computed

We use the term *quality certification* to refer to all of these activities, namely: computing quality indicators, associating them to query results, computing quality rating functions and summary quality profiles, and finally, delivering this information to the data consumer. Note that we assume that providers are fair and do not attempt to provide false quality metadata; in the scenario supported by our current architecture, no third party authority is introduced to guarantee fairness.

A prototype of the architecture has been implemented in collaboration with a large consortium for information systems development for public sector data, based in Italy. The implementation is done as part of a pilot project to demonstrate the feasibility of quality-aware data exchange and marketing, and is based on actual production data describing regional businesses; indicators are obtained from quality assessment analysis that is already part of the consortium's data management procedures.

In the rest of this paper we describe our use case for certification (Section 2), followed by the data model for quality certification (Section 3) and its syntax; we introduce the concept of quality rating (Section 4) and the data architecture for delivering certified data, in Section 5. Our early experiments have highlighted issues regarding the applicability of quality certification principles to information systems on a larger scale, prompting us to analyse the overall cost of setting up quality certification in a systematic way. We conclude the paper with a brief analysis on reusability and cost-effectiveness of scaling our approach to the enterprise level.

2. A QUALITY CERTIFICATION SCENARIO

Our quality certification experiment is set in the context of a pilot project on data exchange in cooperative information systems, promoted by CSI-Piemonte ("Consortium for Information Systems", CSI for short), a public consortium with over fifty members serving the public sector ICT needs for Piedmont, one of Italy's largest regions. Currently the largest Italian ICT company involved in e-government projects, CSI develops and maintains information systems that manage most of the regional public sector data, ranging from healthcare, to taxation, education, environmental health, and so forth.

For CSI, system interoperability is critical to the deployment of cooperative processes, and efforts are under way to standardize its data exchange protocols; making quality metadata explicit is part of this effort. The current project

aims at demonstrating the practical feasibility of certification, by showing that (i) quality metadata can be created by data owners at reasonable cost; (ii) it may be passed on to downstream systems using a standardized format, and (iii) it adds value to the data by providing decision elements for its acceptability.

The experiment is centered around the strategic AAEP system[2], which maintains vital data on regional businesses (i.e. their tax ID, official names, addresses, legal status, and so forth). The database contains just below two million records. Note that this large number, well above the number of actual regional businesses, is due to the fact that AAEP records each business location for each company separately (including local branches of more global companies). AAEP receives batches of data from five upstream providers, which are informally endorsed with different authoritativeness on the data, and at various frequencies. The quality of AAEP's data content is affected by these providers, and it propagates downstream to multiple consumer systems – see Figure 1. These systems include analytics applications (eg for business demographics), but also, more critically, business tax revenue management applications. As the figure shows, multiple data feeds supply these systems with redundant data about businesses' vitals; among these feeds, AAEP is considered to be the one with the highest quality, and is used as a reference, allowing completeness and consistency to be enforced. Making its data feed quality-certified would provide consumers with valuable information for quality assessment purposes.

A number of systems are periodically updated by AAEP using large batches of data, which may realistically include most of the database contents. Thus, the following certification scenario has been envisioned. To begin, a human operator for a downstream system requests a large update from AAEP. We have considered two main batches, in the range of about 1.8 million and .58 million records.[3] Along with the update request, operators may specify one of a set of pre-defined quality rating models that should be applied to the batches. These models are implemented and maintained by AAEP. In addition to normal query processing, the data flow implemented by AAEP now involves an additional lookup into a quality indicators repository, where the metadata is indexed by data identifier; a summary quality profile is computed using the available indicator values, and the requested quality ratings are computed. The resulting metadata is returned to the operator along with the query result, encoded as a separate XML document; finally, a dedicated graphical interface displays the quality profile.

As mentioned, the summary profile only includes statistics about the indicators' values, rather than the value of the indicators themselves. For this reason, once users have determined appropriate thresholds for data acceptability, a second query is issued to AEEP, which fetches all and only the data that satisfy the threshold criteria. This second interaction with the system is not part of the current prototype.

[2]AAEP – "Anagrafe Attività Economiche e Produttive"

[3]These two batches includes the records for all active businesses, and for comparison, the subset of those records that originated from the more authoritative of the data sources.

3. DATA MODEL FOR INDICATORS

Data quality certification is about computing certain types of indicator values for the data, annotating them with suitable descriptors, and encoding them in a recognizable format to be returned alongside the results of a query. We now describe the underlying data model defined for this purpose.

3.1 Available quality indicators

Quality indicators are measurable quantities that are associated as metadata to domain data, and that are deemed useful to compute certain quality features. The most common types of indicators that are used in the data quality practice are available either as the result of deliberate quality assessment analysis carried out on the data, for example on completeness estimation, or as *provenance* information, e.g. a formal description of the processes that produced the data. For this work, we are only going to consider the first type of indicators, which are more immediately associated with traditional quality dimensions. Among these, we include those that are routinely computed on the AAEP database as part of the CSI data quality practice. With reference to the traditional distinction among quality dimensions offered in [13], the following types of indicators are considered:

- completeness of a value, i.e. non-null field value in a record. This may be important to assess, for example, how many addresses are available (some of the records in our test data set are missing values in fields that should be used as keys);
- syntactic correctness of a field with respect to the parsing rules for its domain;
- semantic correctness of a field with respect to value validation rules;
- internal consistency of a collection of fields within a record, according to domain-specific rules;
- uniqueness of a record within the database.

Rules vary, of course, for the syntactic and semantic correctness, as well as for internal consistency of different data domains. A business tax ID, for example, is a code composed of a number of sub-fields, for which internal consistency can sometimes be established; it also contains a checksum digit, which allows one to test the integrity of the whole code. In all the cases considered, these rules are well-defined and quality analysis of individual records will always return a value.

The *uniqueness* indicator refers to the classic record linkage problem, and deserves an explanation. The problem is to establish the probability that two records in a database represent the same real-world entity. Most of the techniques that have been developed to address the problem [14, 15, 5, 8] usually provide a classification of each pair of candidate records into one of three classes (match, non-match, or undecided). Our project, on the other hand, has adopted the record clustering approach implemented in the SAS/DataFlux package for data quality analysis.[4] In this

[4]DataFlux: http://www.dataflux.com/

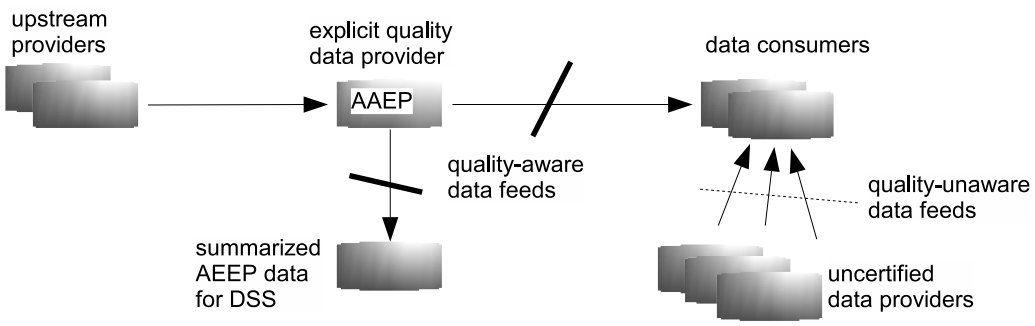

Figure 1: Use case scenario for certified data exchange

approach a single hash value, called *match code*, is computed for each record, independently of all others, based on the values of some of its fields and on some configurable similarity rules; domain experts may configure the way field values are used in the hash function (for instance, they may specify that only some prefix of a value should be used). Records with the same hash value are considered duplicates, and a collection of clusters of such duplicate records is returned.

While there is no guarantee that the choice of hash function, and of configuration, yields genuine duplicates, past experiments on AAEP data have led to some configuration for which the fraction of false positives and false negatives is acceptable. As it does not need to iterate on a quadratic number of candidate pairs, the method is linear in the number of records in the database, thus trading speed for accuracy.

A list of the actual set of AAEP indicators is shown in Table 1, along with a brief description of their associated rules, and the domain for their values. It is important to note that all of these indicators are periodically recomputed for the AAEP database, independently of any user query.

3.2 Properties of indicators

Indicator are described by a small set of properties that help the user interpret their values. Firstly, a distinction is made between exact and statistical indicators. Most of the indicators shown in the table are computed using automated procedures; in this case, exact indicator values can easily be associated to each record. However, computing indicator values can be expensive in some cases. The semantic correctness of a business address, for example, is validated either by cross-referencing the addresses with other databases, or by direct inquiry to the company officers. To limit the cost of the assessment, the value for these indicators may be obtained using statistical sampling, rather than extensive computation. When sampling is performed by drawing from the entire population without distinction, then a single estimate for the value, expressed using standard statistics (average, standard deviation) is associated to each record. However, to achieve better precision it is sometimes possible to partition the population (i.e., the database) into *strata* according to some natural criteria; for example, the population of regional businesses may be stratified according to their province. The term stratification in statistics refers to the division of a population into sub-populations. It is normally used when the overall population is heterogeneous, and at the same time homogeneous sub-groups, or strata, can be identified. The intended effect of stratification is to improve the precision of the estimates for each stratum.

A second distinction is the *scope* within which the indicator value applies. For the most part, indicators are record-oriented, so that the repository can be indexed using the data identifier. However occasionally, indicators are associated to the entire database, for example an estimate of its completeness relative to one of the primary data sources for AAEP. Finally, the procedure used to compute the indicator, or its *provenance*, is also described. For instance, it may be relevant to know which reference database was used to assess address correctness, or that duplicate detection was performed using the "match code" technique.

Indicator descriptors containing these properties are associated to the actual indicator values returned with the query result. As for the values themselves, in principle, the vector of values associated to each record in the query result could be returned. However, we choose to provide a more succinct information, consisting of the frequency distribution of values of each indicator over the query result. One of the reasons was mentioned earlier: for statistical indicators, the same value may be shared by a large number of data items. Furthermore, for large results such as those used in our examples, carrying an additional vector of indicator values would add greatly to the data volume. More importantly, it is unclear whether, in a realistic setting, a provider would be willing to release the entire details of its quality analysis, and whether the users would actually be interested in such a level of detail.

A complete list of elements for the quality certificate can be obtained from the XML schema used for its encoding, shown in Figure 2 (where the attributes are omitted). The attributes for the qualityIndicator element include its unique name, the scope (called "applicability"), and its purpose, along with an optional reference (a URI) to external documentation on the indicator. The core information is the quality profile, captured as a exactValues or statisticalValues sequence structure. Each data point in the sequence is sim-

Short name	Description	Exact / statistical	Values
COER_DATI_CESSAZ_AZ	consistency between two fields	exact	boolean, N/A
PRES_DATA_COST	completeness of a date field	exact	boolean
PRES_DENOM	completeness of a field	exact	boolean
PRES_NAT_GIUR	completeness of a field	exact	boolean
PRES_ATECO	completeness of a field	exact	boolean
COER_NOME_SEDE	consistency between two fields	exact	boolean, N/A
COMPL_SEDE_LEG	completeness of a multi-part field	exact	code indicating which parts are present
CORR_FORM_COD_FISC	syntactic correctness of a structured field	exact	boolean
COER_DATI_CESS_SEDI	consistency between two fields	exact	boolean, N/A
COER_DATI_ATT_SEDI	consistency between two fields	exact	boolean, N/A
PRES_DUPL_SEDE_LEGALE	uniqueness / record linkage	exact	see note (1)
ACC_SINT_IND	syntactic correctness of a structured field. See note (2)	exact	boolean. See note (3)
ACC_SEM_IND	semantic correctness / currency of a field	statistical	probability estimate of correctness
PRES_DUPL_SEDI	uniqueness / record linkage	exact	see note (1)

Table 1: Indicators used in the AAEP use case

Notes:
(1) Value is "false" if no duplicates are detected; otherwise, the value is a list of the data sources from which the duplicates originate.
(2) Correctness uses address normalization followed by a match against a reference database, which is assumes to be correct and complete.
(3) Value is boolean; however, a third value "fix" indicates that the correct value has been found during the analysis, and it is available. Note that, for legal reasons, the wrong value may not be replaced.

ply a (name, value) pair, indicating either the exact or the statistical distribution of values over the data set. When the values are statistical, additional information is included. Here is an example:

```
<qc:qualityIndicator name="acc_sem_ind"
      purpose="Address_semantic_accuracy"
      applicability="record">
   <qc:statisticalValues
          stratum_condition="All"
          stratum_size="1807320"
          sample_size="1000">
      <qc:dataPoint label="NO" value="0.4" />
      <qc:dataPoint label="OK" value="0.6" />
   </qc:statisticalValues>
</qc:qualityIndicator>
```

More sequences of values may be added for multiple strata. Additionally, the optional algorithm element may be used to describe the algorithm used to compute the values. This is particularly useful when reference data sets are involved in the quality assessment. The following example also shows that different algorithms, and different reference data sets, may be used depending on specific *enabling conditions*. This is designed to provide great flexibility in the description of the computation method for indicators:

```
<qc:qualityIndicator name="acc_sint_ind"
      purpose="Address_syntactic_accuracy"
      applicability="record">
   <qc:algorithm name="Match_vs_reference_street_atlas"
enabling_condition="Normalized_address">
     <qc:reference_dataset
          enabling_condition="Comune_in_Piemonte">
          <qc:database name="SITAD" ref="" />
     </qc:reference_dataset>
   </qc:algorithm>
   <qc:exactValues>
     <qc:dataPoint label="Correct"
          value="979825" percent="54.2" />
     <qc:dataPoint label="Incorrect"
          value="654104" percent="36.2" />
     <qc:dataPoint label="Fixable"
          value="173387" percent="9.6" />
   </qc:exactValues>
```

This fragment indicates that the SITAD reference database is only used for towns in the Piedmont region. It must be emphasized that, at this stage in the project, the conditions are expressed using a small vocabulary of keywords that are assumed known and understood by all parties involved. These are simply annotations on the indicators, and no automated computation is currently expected on them. In future work (please see Section 6), our plan is to provide a more formal definition of such shared vocabulary, for the purpose of automatic interpretation. Standard provenance metadata may also be associated to each indicator, using the meta-inf element. Dublin Core metadata[5] is currently used for this purpose.

[5]Dublin Core: http://www.dublincore.org

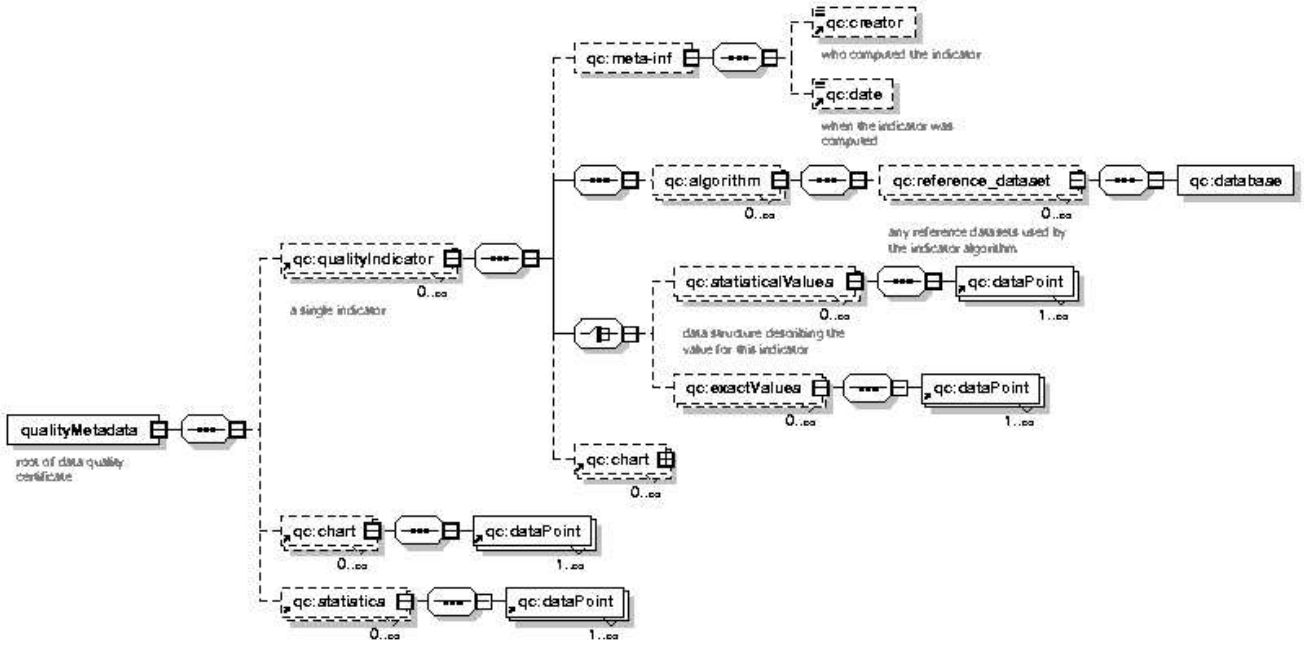

Figure 2: XML schema for quality certification

The model offers several chances for the provider to add collections of data points, using the chart element either as part of an indicator definition, or at a more global level, under the certificate root element metadata. When added to an indicator, the typical use for a chart is simply to suggest a presentation model for the existing exact or statistical data points. By providing all the necessary elements for a graphical display, this practical solution simplifies the processing of the certificate by the client, by removing its need to interpret the data values. For instance, a chart associated to the acc_sint_ind presented earlier may read:

```
<qc:chart title="address_syntactic_accuracy" type="pie">
   <qc:dataPoint label="Correct" value="979825" />
   <qc:dataPoint label="Incorrect" value="654104" />
   <qc:dataPoint label="Fixable" value="173387"/>
</qc:chart>
```

Global charts, on the other hand, may convey additional analytical information that the provider deems important to clarify data quality, and whose scope is broader than the indicator. For instance:

```
<qc:chart xAxis="Condition" yAxis="Data_count"
   title="Data_distribution_by_condition"
   type="histogram">
   <qc:dataPoint
     label="Formally_Correct_taxID" value="580054" />
   <qc:dataPoint
     label="Name_present" value="580186" />
   <qc:dataPoint
     label="HQ_name_complete" value="576290" />
   <qc:dataPoint
     label="Beginning_date_present" value="281262" />
   <qc:dataPoint
     label="HQ_name_uniqueness" value="577899" />
   <qc:dataPoint
     label="Address_semantically_correct_>_55%"
     value="1807316" />
   <qc:dataPoint
     label="Consistent_beginning_dates" value="579875" />
   <qc:dataPoint
     label="Consistent_termination_dates" value="575260" />
</qc:chart>
```

Finally, a more general statistics element may be used to add arbitrary (name, value) pairs that do not necessarily fit in the chart format, often to provide summary metadata. The following fragment states that the data is the result of a named canned query on AAEP, and that the user specified that no quality ratings be computed, but instead requested that all available metadata be returned.

```
<qc:statistics name="Model_information">
   <qc:dataPoint label="Data_source" value="AAEP" />
   <qc:dataPoint label="Query"
     value="Active_businesses_data_from_all_sources" />
   <qc:dataPoint label="Model" value="Metadata_only" />
</qc:statistics>
<qc:statistics name="General_informations">
   <qc:dataPoint label="Total_Number" value="1807316" />
   <qc:dataPoint label="Data_satisfying_all_conditions"
     value="153592" percent="8.5" />
</qc:statistics>
```

The chart elements are also used to report on the quality ratings optionally computed on the data using the indicators, as described next.

4. QUALITY RATING FUNCTIONS

While the "raw" indicators do convey some information regarding quality, as we have seen they are often defined at a low level, and their understanding may require substantial insight into the details of data quality analysis. Even when the meaning of the individual indicators is clear, it is often the case that different indicators "point in different directions"; for example, there may be a trade-off between data accuracy and timeliness, when accuracy depends on expensive and time consuming manual validation of the data. Although the user may decide that, for a particular usage of data, timeliness is more important than accuracy, it is difficult to formulate this preference as a decision procedure on indicator values.

Quality ratings are simpler, pre-defined higher-level metrics computed from the indicators, that hopefully have a more intuitive meaning to the end user. Rating models implement the decision logic corresponding to some of the expected user preferences, for example by assigning different weights to different indicators. Data providers may decide to offer a collection of such ratings models, which may differ in the relative importance they assign to the indicators. Data consumers may also define their own models; note that, in our architecture, they are computed on the provider side (because only summary metadata is delivered to the user), so that such custom models would have to be hosted by the provider.

Two main options are available to design ratings models, namely by manually encoding data experts' knowledge into functions of the indicators, or by automatically inducing the model using machine learning techniques. Our current experiments focus exclusively on the first option. Specifically, two rating models have been designed, with the help of the data domain experts, which take into account two different sets of indicators. Only preliminary validation has been performed on these models, using the experts' judgement as a subjective metric to assess their effectiveness on sample data sets. Among several possible choices of decision models, we have selected decision trees, since they appeared to accurately capture the experts' rules. The trees are defined as follows.

- The rating r is an integer in the conventional range $[-5, 5]$, 5 being the best rating; a neutral rating of 0 gives no indication as to the expected quality of the data;
- At each node n_i in the tree, the value of a single indicator is tested. The outcome of the test determines which branch down the tree is taken, and it also determines a partial rating r_i;
- when a leaf is reached, a function of the partial ratings r_i is computed to yield the final rating r. In the simplest case, this is just the average of the r_i. However, in the current implementation the rating function may include user-configurable weights; while naive users are expected to select one of the available models as a black box, more advanced users have the option to set these parameters. Since each r_i corresponds exactly to one indicator, users may assign weights w_i to the indicators in an intuitive way. In this case, the final rating is

$$r = \frac{\sum_i r_i \times w_i}{\sum_i w_i} \tag{1}$$

As an example, a (partial view of) one of the available decision trees is shown in Figure 3. The first two levels test the presence of the full name of the business, for which two versions exist (the mutual consistency of these two identifiers can be tested, the difference in their meaning being quite subtle). These nodes provide different ratings depending on the presence of at least one of the two names, along with a complete address. Of particular interest are the bottom nodes in the figure, which account for the semantic accuracy of addresses. Since this is a statistical indicator, the expression acc_sem_ind $> X$ indicates the probability that any single record has a semantic accuracy greater than X (rather than referring to a specific record). When this condition is true, then the rating is a function of this probability, computed as

$$r_4 = 1 + 20 \times (x - 0.8) \tag{2}$$

For $0.8 \leq x \leq 1$, this gives $1 \leq r_4 \leq 5$. When the condition is false, we compute some $-5 \leq r_4 < 1$ as

$$r_4 = 1 + \frac{15}{2} \times (x - 0.8) \tag{3}$$

For completeness, we include the corresponding functions for the right-hand side version of the same node, which computes similar ratings:

$$r_4 = 1 + \frac{80}{3} \times (x - 0.85) \tag{4}$$

$$r_4 = 1 + \frac{120}{17} \times (x - 0.85) \tag{5}$$

Quality ratings are computed as requested by the users, for each individual data item in a query result (ratings can for the most part be pre-computed, however, as pointed out in the next section). Similarly to indicator values, the individual ratings are then aggregated and only the distribution of their values is presented in the quality certificate, using the global chart element for the purpose. Figure 4 shows the actual chart for the ratings obtained by applying the tree sketched above, to the second of our test queries. The results show that most of the AAEP records that originate from the most authorative original source (this is what the query returns) get good ratings, according to the model described.

5. QUALITY CERTIFICATION DELIVERY ARCHITECTURE

The prototype has been implemented as a new component of the existing SAS-based query infrastructure for AAEP, as shown in Figure 5. The component includes the following elements:

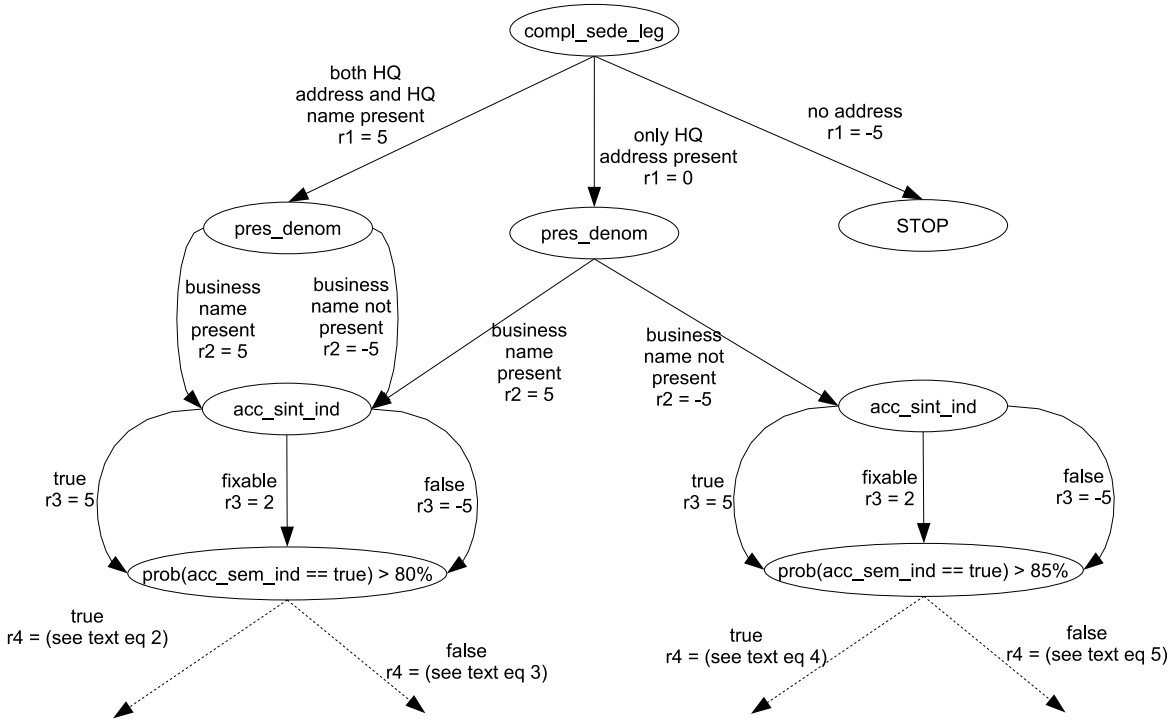

Figure 3: Example decision tree for some of the indicators in Table 1

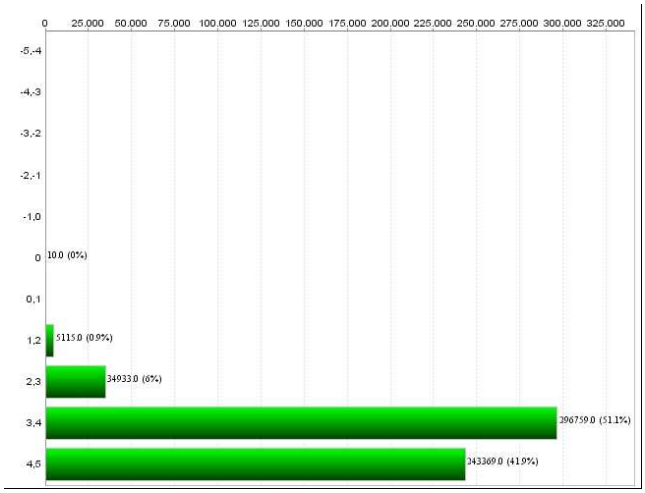

Figure 4: Ratings chart for one of the test queries

- a metadata repository, where quality indicators are indexed by data identifier. A separate, smaller structure is used to hold statistical and dataset-oriented indicators;
- a metadata query interface for accessing the repository;
- an interpreter for XML-formatted query requests, and encoder of quality certificates, which also manages the entire quality certification process;
- the quality ratings evaluator.

This component is accessed through an XML-based interface, which can easily be wrapped as a Web service. Quality-unaware queries will continue to flow over the current interface.

Most of the metadata required by the certificate is computed before query time and stored persistently in the metadata repository. This information includes all values for the available indicators, as well as the partial r_i quality ratings for all of the decision trees. This way, the indicator weights w_i are the only dynamic features of metadata; users wishing to make use of configurable rating models, should be aware that the final ratings must be computed at query time, according to equation (1). In summary, user queries are answered as follows:

- the query is encoded as an XML document by the client component (JSP in the prototype). The request includes the query (its logical name in the case of canned queries) as well as the choice of rating models to apply, if any;
- the query request is interpreted and forwarded to the underlying query processor for AAEP;
- for each record in the query result, the metadata repository is accessed to retrieve the corresponding metadata;
- once all indicator values are available, the chosen quality ratings are retrieved (or computed, if parametric),

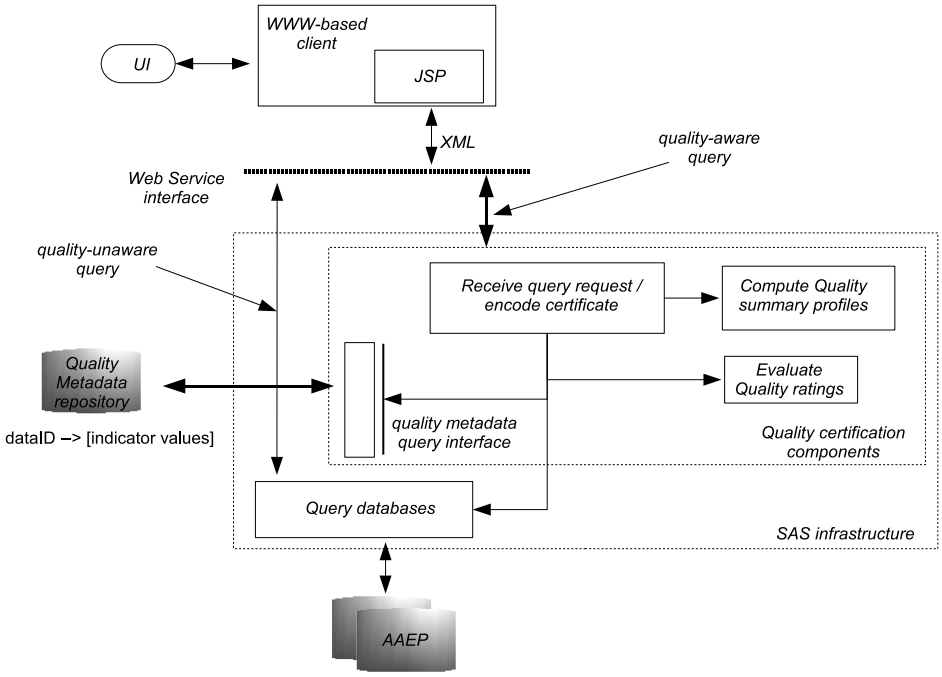

Figure 5: Quality certification architecture

and the summary profiles are computed;

- the certificate is encoded in XML and sent across the interface, separately from the query result.

In this scenario, the client can be very thin, its minimal capability being to offer the user a choice amongst pre-defined items, and to render simple charts.

Minimizing the dynamic aspects of certification results in a scalable architecture. An obvious, but intrusive, further optimization would consist in integrating the quality metadata into the AEEP database, so that no additional queries are needed at all.

As mentioned in the introduction, the drawback of a concise certificate information is that there is currently no way for the user to operate a selection of records based on the summary quality profiles. In the continuation of this project, follow-up user queries based on ratings and indicators thresholds will be offered, resulting in a complete two-step interaction.

6. DISCUSSION

In this paper we have presented a general and practical data model and architecture for providing succinct quality metadata information in response to a user query; this metadata can be assembled efficiently at query time and it can be interpreted easily by non-expert users.

A number of issues have emerged from our experience with the initial implementation. Firstly, we have been assuming that certificates hold metadata that is true to the best of the provider's knowledge. In other words, we assume that the provider is trusted, and we ignore the possibility that the provider may deliberately mislead its users by crafting untruthful certificates. We believe that this falls into a general issue of trust management, which is not specific to research on architectures for data quality.

Secondly, indicator values may be incomplete: there is no guarantee that all values will indeed be available on the data and at the time when they are needed. When indicators are missing, less significant quality summaries can still be offered; however, some rating models, including decision trees, are not robust to missing input values. We are currently investigating solutions in the area of decision models in the presence of uncertain information, for which well-understood techniques exist [12].

The issues of reusability of the model and architecture and scalability of the approach are perhaps the most pressing from the business perspective. In the CSI architecture, indicators for AAEP data emerge quite naturally as part of a separate effort devoted to quality management, which is facilitated by the adoption of integrated SAS data quality solutions. They are computed incrementally, according to existing business plans, and reused for the purpose of certification. We are aware, however, that computing indicators anew is an expensive proposition that needs to be justified

by a cost/benefit analysis; and that there is hardly any practical evidence that quality awareness actually brings added value to data, either in the context of cooperative systems, or for data marketing.

On the other hand, it is clear that the model and architecture presented in this paper are sufficiently generic to be applicable to a number of information systems within the company. The more specific elements of the architecture are the rating models, which encode specific domain knowledge that may not be reusable beyond the boundaries of a single system. Based on these considerations, the continuation of the project will address the issue of scalability, with priority given to strategic data that is central to multiple cooperative processes.

A final point concerns the vocabulary used in the indicator descriptors, which currently lacks any formal specification. Following current trends in semantic web techniques for data and service annotation (see for instance [7]), one may attempt to define a formal model, i.e., an ontology of quality descriptors, for the purpose of sharing the meaning of the terminology used, and, eventually, for automated processing (generation, interpretation) of these annotations. To the best of our knowledge, this has not yet been done, and it is part of our plans for further work.

7. ACKNOWLEDGMENTS

We are grateful to Giuliana Bonello at CSI for promoting this project, and to Dr. Suzanne Embury at the University of Manchester for her valuable comments on this paper.

8. REFERENCES

[1] A.Motro, P.Anokhin, and A.C. Acar. Utility-based resolution of data inconsistencies. In Felix Naumann and Monica Scannapieco, editors, *International Workshop on Information Quality in Information Systems 2004 (IQIS'04)*, Paris, France, June 2004. ACM.

[2] Laure Berti-Equille. Quality-adaptive query processing over distributed sources. In *Procs. 9th International Conference on Information Quality, ICIQ 2004, Cambridge, Ma*, 2004.

[3] H. K. Bhargava and S. Sundaresan. Managing quality uncertainty through contingency pricing. In *36th Annual Hawaii International Conference on System Sciences (HICSS'03)*, 2003.

[4] C. Cappiello, C.Francalanci, B.Pernici, P.Plebani, and M.Scannapieco. Data quality assurance in cooperative information systems: a multi-dimension quality certificate. In T.Catarci, editor, *DQCIS 2003: International Workshop on Data Quality in Cooperative Information Systems*, Siena, Italy, January 2003.

[5] M.G. Elfeky, A.K. Elmagarmid, and V.S. Verykios. Tailor: a record linkage tool box. In *Proceedings of the 18th International Conference on Data Engineering (ICDE 2002)*, San Jose, CA, Feb. 2002. IEEE Computer Society.

[6] S. Frlung and J. Koistinen. Qml: A language for quality of service specification. Technical Report HPL98-10, HP Labs, HP Software Technologies Laboratory, 1998.

[7] Y. Gil, E. Motta, V.R. Benjamins, and M. Musen, editors. *Proceedings 4th International Semantic Web Conference, ISWC 2005*, volume 3729 of *Lecture Notes in Computer Science*, Galway, Ireland, November 2005.

[8] M.Buechi, A.Borthwick, A.Winkel, and A.Goldberg. ClueMaker: A language for approximate record matching. In *Procs. 8th International Conference on Information Quality, ICIQ 2003, Cambridge, Ma*, 2003.

[9] M. Mecella, M. Scannapieco, A. Virgillito, R. Baldoni, T. Catarci, and C. Batini. Managing data quality in cooperative information systems. In Robert Meersman and Zahir Tari, editors, *DOA/CoopIS/ODBASE 2002*, volume 2519 of *Lecture Notes in Computer Science*, pages 486–502, Irvine, California, USA, 2002. Springer.

[10] M.Scannapieco, A.Virgillito, C.Marchetti, M.Mecella, and R.Baldoni. The daquincis architecture: a platform for exchanging and improving data quality in cooperative information systems. *Inf. Syst.*, 29(7):551–582, 2004.

[11] F. Naumann, U.Leser, and J.C.Freytag. Quality-driven integration of heterogenous information systems. In *VLDB'99, Proceedings of 25th International Conference on Very Large Data Bases*, pages 447–458, Edinburgh, Scotland, UK, September 1999. Morgan Kaufmann.

[12] J. Pearl. *Probabilistic Reasoning in intelligent systems: Networks of plausible inference*. Morgan Kauffman, 1988.

[13] M. Scannapieco, P. Missier, and C. Batini. Data quality at a glance. *Databanken-Spektrum*, 14:6–14, 2005.

[14] W.E.Winkler. Exact matching lists of businesses: Blocking, subfield identification, information theory. In Alvey and Kills, editors, *Record Linkage Techniques*. US Internal Revenue Service, 1985.

[15] W. E. Winkler. Methods for record linkage and bayesian networks. Technical report, U.S. Census Bureau, Statistical Research Division, 2002.