

REPRESENTATION AND CERTIFICATION OF DATA QUALITY ON THE WEB

(Research Paper)

**Cinzia Cappiello
Chiara Francalanci
Barbara Pernici**

Politecnico di Milano, Milano, Italy
{cappiell, francala, pernici}@elet.polimi.it

Francesco Martini

C.A.D. Romana Lippini Srl, Prato, Italy
francesco@romanalippini.it

Abstract: The large majority of users accesses Web pages without considering the quality of their contents. In the Web environment, users are not provided tools that measure the quality and reliability of information, while Web information is often non reliable, as it is gathered from different sources that may not be integrated and consistent with each other. In distributed systems, such as the Internet, databases belong to different domains, are built according to different requirements and are updated at times and with procedures that depend on the specific context. Consequently, when information is integrated, problems concerning data consistency, accuracy, usability, and timeliness can arise. This paper addresses these issues and proposes a tool for the evaluation of data quality on the Web and for the management of the related certification process. For this purpose, the paper defines a data quality representation model and a language for the description and implementation of quality metadata by comparing different standards from the literature.

Key Words: Data quality, Web data, Quality Certification

1. INTRODUCTION

The fast development of information technology has provided companies with tools suitable to manage large amounts of interrelated data in a short time. Only a few years ago, it was possible to manage databases storing millions of records typical of complex organizations, such as public administrations, financial institutions, and universities, only with expensive systems. Nowadays, these databases can be accessed with a notebook. In particular, the increasing number of interconnections among information systems over the Internet and, in general, the number of resources available in a computer network allows organizations and individuals to share enormous quantities of data. This availability of information has augmented the flexibility of information systems, but it has also raised data quality problems. This could seem a paradox, since a fundamental goal of computer science, especially in the information systems field, has always been to augment the correctness, reliability, and accuracy of databases. In traditional information systems, databases were manipulated primarily manually and this interaction with users introduced a high error rate. The continuous advancement of computer science has limited data entry activities by users, but error probability is still high due to a number of open issues, such as the non linear

usage of the databases and the lack of database integration. The lack of database integration can cause many errors. In distributed systems, such as the Internet, databases belong to different domains, are built according to different requirements, and are updated at times and with procedures that depend on the specific context. Consequently, when information is integrated, problems concerning data consistency, accuracy, usability, and timeliness can arise [12].

Data quality has been successfully applied within database management processes to eliminate low quality data from organizational databases. In particular, in the latest years, different data quality programs have been created and introduced in organizations. For example, in 1995 MIT has created the Total Quality Data Management (TQDM). In 2001, the American Government has supported the Quality Act for the definition of quality criteria for the exchange and management of strategic information and ISO has reconsidered the ISO/IEC 9126 on the quality of software products to address data quality issues. Several authors define the quality of data as their “fitness for use”, i.e., the ability of a data collection to meet users’ requirements [15][24]. Data quality is a multi-dimensional concept and each data quality dimension is specifically related to a particular aspect of data such as data views, data values and data representation. The assessment phase is particularly relevant in a data quality assurance program. Quality assurance is faced by the need for objective measures of quality, since most often users cannot judge the quality of data or simply trust data sources.

This paper addresses this need by proposing a certification model for data quality over the World Wide Web which could help users to understand the reliability of the information contained in a Web page. A Web page will be certified only if quality requirements are satisfied. In general, organizations offer different types of services to satisfy different user requirements and each type of service is associated with different information and corresponding quality levels. Our goal is to be able to certify these different quality levels. For this purpose, we define a data quality representation model and we select a language for the description and implementation of quality metadata by comparing different standards from the literature. We also propose a structure and a formal representation of digital quality certificates for Web pages.

The paper is structured as follows. Section 2 discusses the issues related to the representation of the information over the Web. Section 3 presents the model for data quality certificate and Section 4 shows the architecture developed for the certification process. Section 5 presents a survey conducted to determine the most suitable language for the representation of the quality metadata. Finally, Section 6 contains an example of the implementation of the certificate.

2. THE REPRESENTATION OF INFORMATION OVER THE WEB

A Web page has a structure specifying the information objects composing the page and their position. Objects can have different types. For example, HTML is the standard language to create Web information objects, but popular document formats such as DOC, PDF and PS can also represent Web information objects. Web information objects can also be specified as queries on a specific database. Finally, multimedia files, such as images, sound and videos can be information objects of a Web page. Given the diverse nature of information objects, quality metadata cannot be associated with Web pages, but must be specifically designed for different types of information objects.

Data quality problems related to the validity of information of a Web page over time have been investigated in [16]. In this paper, the authors have considered the Web page as a unique object and quality metadata have been associated with the whole page. In our model, quality metadata are associated with each object contained in a Web page to derive the page’s quality metadata by aggregating corresponding objects’ metadata according to the page’s structure. In the following section the metadata model is described.

2.1 Design model for metadata over the Web

We can state that data quality analysis is based on the construction of a conceptual model that is called “data quality schema”. The data quality schema is a metadata model that can be formalized with classical data modeling techniques. Metadata are defined as data that contain information about other data in the same way as data contain information about the real world along a specific user view.

In Figure 1, it is possible to notice that each object contained in the Web page is associated with metadata that are distinguished in first-level metadata MD1 when they represent a measure of data quality dimension or second-level metadata MD2 (meta-metadata) when they allow the evaluation of data quality dimensions, that is they represent a metric. A hierarchy of MD1 and MD2 metadata constitutes a data quality schema [4][22][23].

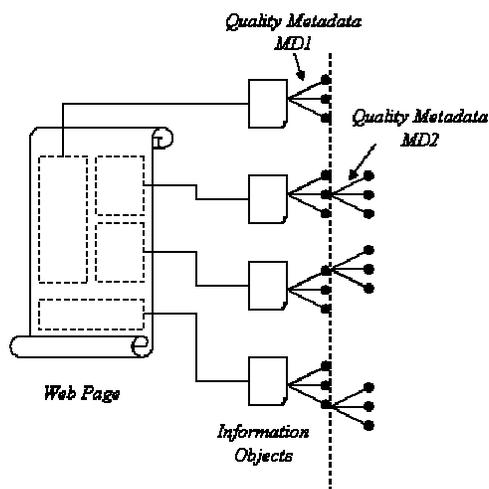


Figure 1- Structure of Web pages and related metadata.

Metadata have the following properties:

- Metadata are data that describe other data. It must be possible to manage them as data in terms of storage, input, and update and to link them to corresponding data.
- Metadata can describe metadata. The structure of the metadata conceptual schema can be composed of different hierarchical levels. Consequently, a DQ schema must have a structural granularity that can express the link among metadata at each hierarchical level and a relational granularity that supports the association between metadata and the reference data model.
- Metadata should be computable (machine understandable information). Metadata must be evaluated objectively by a machine according to predefined rules. Therefore, metadata must be defined according to a semantics and syntax shared among all users.
- Metadata should be platform independent, i.e. portable. The representation format of metadata should not be tied to a particular platform or development tool. Metadata should be compatible with any operating system and with any software tool.
- Metadata must be associated with other metadata or with a source. Metadata cannot be data themselves, but must belong to a data schema.

Table 1 is built by formalizing the design process of a database and focuses on data quality metadata definitions. Note that the choice of a modeling language also depends on the database logical model (e.g.

hierarchical, relational, object oriented, etc. [8]). The ER and UML methodologies cited in Table 1 represent widely used approaches ([2][6][8]).

Data and metadata design phase	Data Output	Data modeling language	Metadata output	Metadata modeling language
Collecting user requirements (isolated data views)	Partial views on data	Native, informal	-	-
Integrated Conceptual Design	Global conceptual data schema	ER, UML etc...	-	-
Collecting DQ user requirements (isolated metadata views)	-	-	Partial views on data quality information	Native, informal
Formalization of DQ requirements and relations between “data item” and “metadata item”	-	-	Partial views on MD1 and MD2	Descriptive language with MD1 and MD2 metadata
DQ Integrated Conceptual Design	-	-	DQ global conceptual schema	ER, UML etc...
Implementation: database logical design	Database logical schema (e.g. relational, object oriented etc.)	DDL has to be used XML		
Implementation: database physical design	File distribution on hard disks	DDL has to be used	-	-
DQ implementation: DQ logic design	-	-	DQ logical schema (e.g. relational, object oriented etc.) Data dictionary DTD,XSD	DDL has to be used XML Schema
DQ implementation: DQ physical design	-	-	File distribution on hard disks	DDL has to be used

Table 1 - Design phases for a database and related metadata.

At the end of the implementation phase, data and corresponding metadata are stored and available for query, input and update operations. Both are recorded and managed in files that can be managed as a unique file or in separate files. It is also necessary to provide descriptors of the associations between data and metadata. These descriptors can be stored in a metadata database or in a separate database. Details on metadata management are presented in Section 5.

2.2 Deriving Web information

In this paper, the analysis of logical database design has been extended to the Web by investigating the impact of publishing aspects on database usage and management. Since about 80% of Web pages are built with relational databases [6], the relational model can be considered a reference paradigm. An additional model to be considered is the object-oriented database technology, which provides inheritance abstraction mechanisms and functions that have data as domain [8]. Finally, the last model to be considered are *documental databases*, that is any document published on the Web and any potential data source that can

be classified as a semi-structured database [19]. In the following table, the characteristics of these reference technologies are presented.

Characteristic	Relational Database	Object-Oriented Database	Documental Database
Structure	Tables structured in fields (columns) and records (rows)	Graphs (fields) with pointers to instances (records)	Text with format properties
Relation modeling	Yes	Yes	No
Class modeling	Yes	Yes	No
Primary key	Yes	Yes	No
Attribute modeling	Yes	Yes	No
Method modeling	No	Yes	No
Rules for attribute retrieval	No	Yes	No
Rules for relation retrieval	No	Yes	No
Aggregation association modeling	No	Yes	No
Definition of attribute properties	No	Yes	No
Relation constraint typology	Cardinality	Cardinality, sorting, visibility, modifiability and navigability	-

Table 2 - Properties of reference database models.

The publishing phase associates a database with the logical concept of distributed source [2]. In the publishing phase, a relation between a part of a database (which can coincide with a data item or with a whole set of records) and a URI (Universal Resource Identifier) is created. In theory, there could be the same number of URIs and of records contained in the database or, alternatively, multiple URIs could be associated with the same record. The main problem is that different URIs corresponding to the same database may not be available at the same time and, consequently, a published database has a *dynamicity* property that distinguishes it from its off-line version.

Dynamicity does not depend on database management methods, but it depends on the inherent dynamicity of the Web and of the domains where the database is distributed. It is fundamental to define a Web-oriented description of the data model. This model will be similar to the off-line version, but will include the data type definition along with the URL type that will be associated with data instances. A universal Web-oriented representation language does not exist for a database. Consequently, it is necessary to analyze the representation languages defined in the literature and evaluate their suitability as database representation languages.

The characteristics of a suitable language to represent information quality on the Web and to support the representation of a data quality schema are:

- Representation of the objects of a typical object-oriented language.
- Representation of the relations among data.
- Availability of a URI descriptor that can be associated with data and metadata.
- Representation of the data of a documental database, by providing tools to identify text portions in a document.

3. DATA QUALITY DIGITAL CERTIFICATE

The term “digital certificate” has been used for the first time by Kohnfelder in 1978 [13]. A digital certificate is a file, built and distributed by an authority or a specific organization, containing all the information necessary to identify the user univocally within a specific communication process [7][13]. Essentially, the certificate is a file that contains four data types [17][20]:

- User identification data (name, optional attributes etc.).
- User public key, stored by the authority when the certificate is issued and associated with the user identifier.
- Certificate descriptive data (name of the Certification Authority (CA), issue date, expiry date, series number, certificate algorithm, etc.).
- Signed data field, ciphered with the private key of the CA that allows the identification of the user in the communication process.

A Quality Certificate (ISO 9002) is a document that guarantees the level of quality required on certain parameters crucial for a specific activity. The quality certification process includes different phases, such as the development and monitoring of suitable procedures to verify the level of different quality dimensions. According to [14] and considering [4][5][7][16][17], a digital data quality certificate (DQC) is a digital document that guarantees a specific quality level of a given data set. In the same way in which a digital certificate authenticates a specific public key and a quality certificate guarantees specific performance levels of a service, a data quality digital certificate authenticates the quality of data provided to a specific user by a specific source. If we apply the typical characteristics of a digital certificate to a DQC, we obtain four properties described in the following:

P1. *Content authentication*: a DQC has to guarantee the authenticity of exchanged data and related quality data. The authenticity of data and metadata can be obtained by attaching a digital signature. This property is sufficient to define the digital data quality certificate when the user has specified a minimum acceptable value for each data quality dimension and exchanged data satisfy user requirements completely.

P2. *Data confidentiality*: is ensured with data encryption.

P3. *Metadata confidentiality*: is ensured with data encryption.

P4. *Source authenticity*: a digital certificate is generated by a specific source which must be authentic.

A DQC must always satisfy property 1, but can satisfy any subset of properties 2, 3 and 4. Properties can be combined in order to obtain eight different types of certificates:

T1. Only property 1 is valid. In this case, the certificate guarantees the quality of exchanged data for a single transaction that is for a single data unit exchanged between source and destination. This type of certificate has the minimum security level and has to be generated for each transaction. The certificate is composed of different sections:

- a. Quality metadata, i.e. the values of data quality dimensions.
 - b. Certificate identifier: it contains data about the certificate and the Data Quality Certificate Authority. For example, the serial number of the certificate, the Authority Identifier, the digital signature algorithm and the public key of the certificate Authority can be included in this section.
 - c. Digital signature: it is composed of data, metadata and private key.
- T2. Properties 1 and 2 are valid. In this case, the quality metadata section includes also Data Confidentiality information that suggests to the user that he must decrypt data.
- T3. Properties 1 and 3 are valid. This is an unusual case in which a source does not wish to publish the quality level of its data, for example, for privacy reasons or special agreements.
- T4. Properties 1, 2 and 3 are valid. The DQC certifies both authenticity and confidentiality.
- T5. Properties 1 and 4 are valid. The DQC authenticates content and source. The certificate has an additional Digital Certificate section. This section stores the digital certificate owned by the source.

- T6. Properties 1, 2 and 4 are valid. DQC authenticates the content, encrypts data and authenticates the source.
- T7. Properties 1, 3 and 4 are valid. Similar to case *c*, it is a case adopted in particular reserved situation in which the organization wants to maintain the confidentiality on the results of data quality assessment procedure.
- T8. Properties 1, 2, 3 and 4 are valid. It represents the most secure DQC. It authenticates and encrypts the content and guarantees the authenticity of the source. The use of this certificate is suitable for important transactions or in critical situations requiring maximum security levels, such as financial transactions, commercial transactions or transactions that exchange highly sensitive data.

The results of our analyses are summarized in Table 3.

DQC Type	Content: data and metadata	Source	Encrypted sections	Access security
1	Authenticate	-	-	-
2	Authenticate	-	Data	Data confidentiality
3	Authenticate	-	Metadata	Metadata confidentiality
4	Authenticate	-	Data + metadata	Content confidentiality
5	Authenticate	Authenticate	-	-
6	Authenticate	Authenticate	Data	Data confidentiality
7	Authenticate	Authenticate	Metadata	Metadata confidentiality
8	Authenticate	Authenticate	Data + metadata	Content confidentiality

Table 3 - DQC types.

The certificate that we have used for our testing phase is the last and most secure certificate. The certificate is associated with data only when data satisfy user requirements. Consequently, in each transaction it is necessary to check whether required values are lower than the values of quality metadata. Indeed, the same data can be perceived in different ways and be judged as high-quality by a user and poor by a different user.

An analysis of the most important digital certificates has been performed in order to select a digital certificate that is suitable for quality certification on the Web. Our analysis has pointed out that the X.509v3 standard is the most widely used and complete standard.

4. CERTIFICATION PROCESS AND ARCHITECTURE

According to [16], an architecture to support the user of a DQC over the Web is represented in Figure 2. The architecture includes a Quality Manager and a Data Manager. The Data Manager is responsible for data management, storage and publication on a Web Page. The Quality Manager is responsible for data quality management and for the management of the Quality Knowledge Repository. A description of the software modules of the architecture is provided in the following:

- *Data Repository*: it contains data that are published on a Web page;
- *Quality Knowledge Repository*: it contains the quality metadata necessary for the calculation of quality values;
- *Web Data Design*: it is responsible for the definition of the representation schema of data that are published on a Web page;
- *Web Data Publishing*: it publishes data on a Web Page;
- *Quality Designer*: it provides the data quality schema on the basis of the information received by the Web Data Design module;

- *Quality Calculator*: on the basis of the data quality schema received by the Quality Designer, the Quality Calculator assesses data quality dimensions by using the information contained in the Quality Knowledge Repository;
- *Monitor*: it verifies the Data Quality levels provided by the Quality Calculator. If the data quality levels satisfy user requirements, data are considered publishable. Further, the Monitor module verifies the values of data quality dimensions periodically or upon events;
- *Quality Publisher*: it prepares quality data for publication and sends them to the Web Data Publishing module.

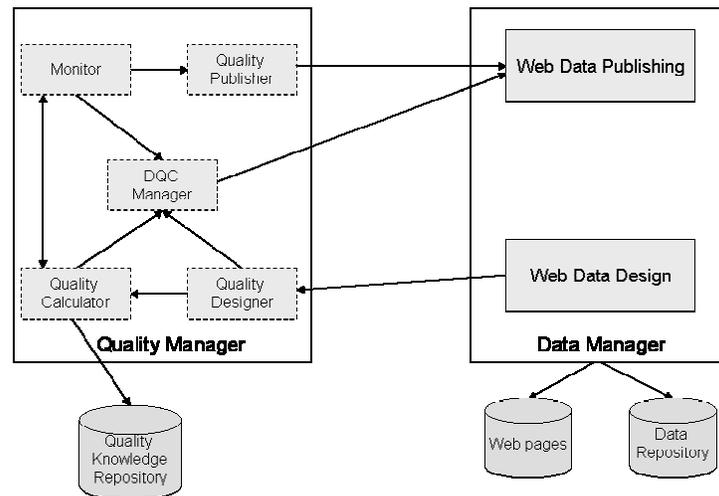


Figure 2 - Architecture to support the user of a DQC over the Web.

The fundamental module in the architecture is the *DQC Manager*. It is responsible for issuing the DQC. It interfaces with the Monitor module to verify that data quality values satisfy user requirements. If user requirements are satisfied, the DQC Manager builds and sends the certificate for a single transaction. The certificate will be manageable as a simple certificate and will be accessible through a common browser after the publication performed by the Web Data Publishing module. In particular, when the user asks for the certification of the Web page using an ad hoc button on the tool bar, a graphic symbol will indicate him the suitability of the Web page. In our example, as described in Section 6, we have used a traffic-light symbolism in which the content of the Web page turns into red if the content is not certified and into green otherwise. The user has also the possibility to see the details about the quality values related to each object contained in the Web page in a new Web page.

5. METADATA MANAGEMENT ON THE WEB

The literature describes three different approaches to metadata management with in Web pages [2][6]. The first approach recommends the encapsulation of metadata inside related data. For example, metadata can be hidden inside the header of an HTML document or in the structure information of a Word document. The second approach suggests that metadata are exchanged separately. Metadata can be managed together with the document or in a separate file. Finally, the third approach suggests that metadata are available in a document different from the one used to store data. As an example, an XML document and its DTD can be stored in different locations and can be retrieved with different URLs.

The first metadata management approach involves size constraints, since it is not possible to store large amounts of information in the header of an HTML document or in the structure information of a document. Further, when data and metadata are stored inside the same document, the client has to

download data in order to obtain metadata and metadata are subject to the same corruption risk as data. With the second approach, an accidental corruption of data does not involve the corruption of metadata and the size of the data quality schema can be increased arbitrarily. Finally, the third solution does not raise issues with the size of metadata and the quality of data can be evaluated before downloading data. Data corruption does not affect quality data, but the time required to obtain information is higher since each time data are retrieved, metadata must be retrieved from a different URL.

We have compared the fundamental descriptive languages for Web information by focusing on XML, XML-S, RDF, RDFS, DAM+OIL and OWL. These languages are markup languages which have been recognized as standards by the World Wide Web Consortium [26]. Our goal was to select the most suitable language for the representation of quality data over the Web. Results show that:

- HTML is not suitable for the description of Data Quality over the Web since it is not appropriate for the definition of new elements or new properties, since it supports the definition of new names for metadata but they are not structured and manageable inside the document.
- XML DTD and XML Schema (XMLS) allow the definition of new elements and new properties, but it does not allow a precise definition of their type, structure and format. Moreover, XMLS does not have the predefined structure to represent the relations among elements, although it allows the representation of object-oriented schemas with a hierarchical model.
- RDF inherits all the proprieties of XML and adds a basic structure for the “class” concept and allows the definition of the relationship between class and Web resource. RDF Schema, in particular, defines the concept of “class type”, implements the inheritance mechanism and specifies the formalization of class properties.
- DAML+OIL extends the representation capabilities of RDFS classes. In particular:
 - it allows recursive inheritance between classes;
 - it provides primitives for numeral sets;
 - it formalizes the equivalence among classes in terms of instances;
 - it allows designers to import new classes from external documents;
 - it implements XML schema Datatypes;
 - it offers greater control on class proprieties, for example, by specifying cardinality, transitivity and equivalence;
 - it allows the definition of classes as a result of combinations among elements belonging to pre-defined classes through mathematical set operations, such as intersection, union, etc.
- OWL is the most complete language for the representation of Web data. It has all the features of DAML+OIL, but it provides a grater variety of user-defined constructs. For example, it supports the definition of “complex classes”. Furthermore, similar to DAML+OIL, it offers all the data types that are specified in XML schemas by using the name-space technique.

Overall, XML Schema, RDF Schema and OWL seem the most suitable languages for the representation of data quality information over the Web. DAML+OIL is not considered, since OWL represents its evolution.

The comparison among XML Schema, RDF Schema and OWL has been performed along the following critical parameters derived from [9], [10], [11], [21]:

- *Context*: capability of a language to tie constructs to different semantical domains and, consequently, to different contexts.
- *Subclasses and properties*: capability of a language to express generalization hierarchies with related attributes.
- *Data Type*: capability of a language to offer multiple data types, such as integer, string, classes, class attributes or complex types.

- *Instances*: capability of a language to specify class instances.
- *Property constraints*: capability of a language to specify property constraints. It is a complex parameter composed of:
 - *Domain*: classes that can be associated with the property.
 - *Range*: objects usable as values.
 - *Cardinality*: it specifies how many values are associated with the property.
 - *Values*: ad hoc-constructs to constrain property values. It is in turn a complex parameter composed of:
 - *Default*: specification of a default value for the property.
 - *Enumeration*: specification of a set of possible values for a property.
 - *Sorting*: specification of a value set sorted implicitly or explicitly.
- *Instruction Type*: instructions that are defined by the language (e.g. negation, conjunction, disjunction).
- *Inheritance*: capability of a language to specify inheritance among classes, either single or multiple.
- *Definitions*: capability of a language to verify the link between a class and a subclass or to verify whether an instance is a member of a class.
- *Predefined data sets*: a language has constructs to specify predefined sets that can be:
 - *Closed Lists*: the elements included in the set are completely specified.
 - *Sorted sets*: sets in which the input order is considered.

The analysis points out that OWL is the most complete language from a semantic perspective. In particular, OWL provides all the constructs that are necessary to represent a data conceptual model, such as classes, inheritance among classes, predefined data sets, class properties, quality constraints etc. As concerns metadata modeling, OWL is also more efficient than XMLS and RDFS at modeling data quality schemas. Further, in OWL, it is possible to represent data and metadata inside the same document, since metadata can be modeled as a property of a specific source (data). Finally, OWL also allows designers to import predefined ontologies. Details on the conducted analysis are contained in Appendix A.

6. LEVEL 8 DATA QUALITY CERTIFICATE DEFINITION IN OWL

A test Web page has been implemented to verify of the certification process discussed in the previous sections. Verifications address the completeness and timeliness data quality dimensions. In the literature, *completeness* is associated with data values and is defined as the degree to which a specific database includes all the values corresponding to a complete representation of a given set of real word events as database entities [18]. According to this definition, it is possible to obtain an objective measure of the completeness of a data source by considering the amount of significant data values and comparing them with the amount of values that should be included in the Web page.

In Web systems, completeness can be measured as the degree to which a Web page includes all relevant information. The evaluation procedure is different for documents and elementary data items. If a section contains a document, the corresponding value of completeness can be obtained from the document's metadata, that is the data certifying the quality of the document. If a section contains a list of elementary data items, completeness can be measured by comparing the list with a data source that is certified as complete.

The timeliness dimension is defined as the property of information to arrive early or at the right time and is usually measured as a function of two elementary variables, currency and volatility [1][3]. A measure of timeliness is defined in [1] as:

$$Timeliness = \max \left[\left(1 - \frac{Currency}{Volatility} \right); 0 \right]^s,$$

where the exponent s is a parameter necessary to control the sensitivity of timeliness to the currency-volatility ratio. With this definition, the value of timeliness ranges between 0 and 1.

Currency is usually defined as the time interval between the time instant in which data are updated and the time instant in which data are used [3]. Volatility is instead a static dimension that expresses the validity of data in a specific context [1].

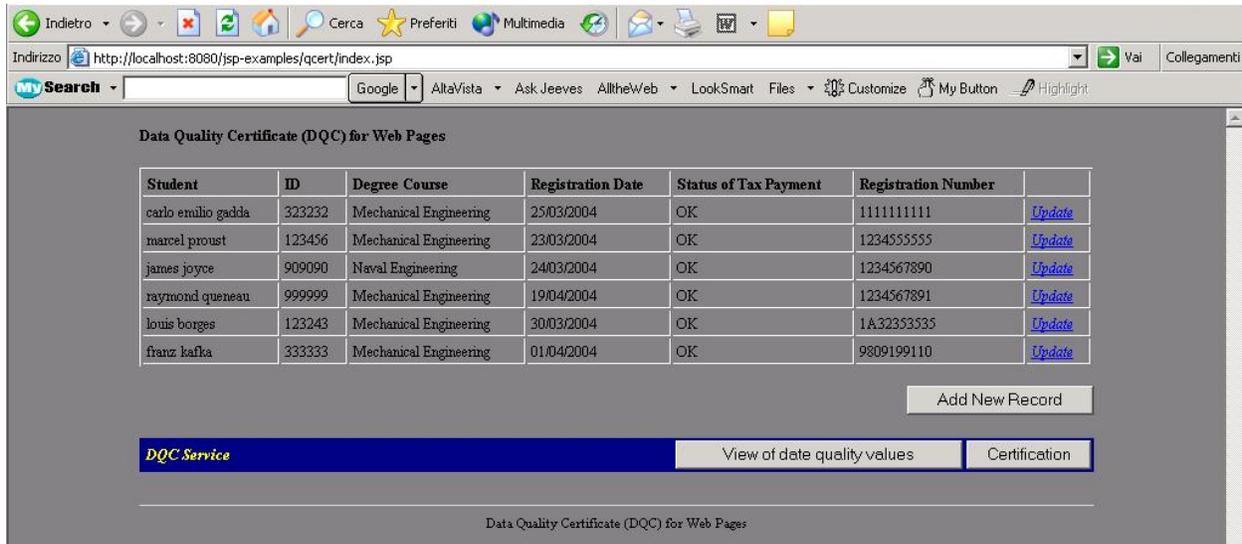


Figure 3 - Web page used as example

In our example, the Web page contains a list of students that are registered for the final discussion of their graduation thesis (Figure 3). We have defined a name space dqs which contains the general syntax to use for a data quality schema. Beside timeliness, currency, volatility, completeness, we have defined the following items:

- $dqs:data$;
- $dqs:metadata$;
- $dqs:index$ representing the primary key associated with the data instance.

We have defined the certification values that represent the minimum values set by the provider as a result of an agreement on the quality level that has to be guaranteed to a specific user. Certification values should be stored by an official Certification Authority. For example, the certification values of timeliness and completeness can be formally specified as follows:

$dqs:timelinessDQCValue = "0.4"$
 $dqs:timelinessDQCTrueTest = ">"$
 $dqs:completenessDQCValue = "1"$
 $dqs:completenessDQCTrueTest = "="$

This specification indicates that the value of timeliness has to be greater than 0.4 and the value of completeness has to be equal to 1 in order for data to be certified.

The data contained in the page, that is the list of students that are registered for their thesis discussion, are described as an OWL data schema:

```

<rdf:Description rdf:about="http:Page_Demo.jsp">
  <demo:registration rdf:resource="#Registration_degree">
    <owl:Class>
      <owl:ObjectProperty rdf:ID="student">
        <rdf:type rdf:resource="&dqs;data"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:ID="ID_student">
        <rdf:type rdf:resource="&dqs;data"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:ID="Degree_course">
        <rdf:type rdf:resource="&dqs;data"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:ID="Registration_date">
        <rdf:type rdf:resource="&dqs;data"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:ID="Paid_tax">
        <rdf:type rdf:resource="&dqs;data"/>
      </owl:ObjectProperty>
      <owl:ObjectProperty rdf:ID="protocol">
        <rdf:type rdf:resource="&dqs;index"/>
      </owl:ObjectProperty>
    </owl:Class>
  </demo:registration>
</rdf:Description>

```

The Data Quality Schema contains the values of the data quality dimensions. In this case, the page is composed of a list of different instances. Each instance can be seen as an object. If the quality values associated with each object contained in the Web page satisfy certification constraints, then the Web page can be certified. If there are one or multiple objects that do not satisfy constraints, the Web page cannot be certified. In the example, the user can check data quality values through a Web page (Figure 4).

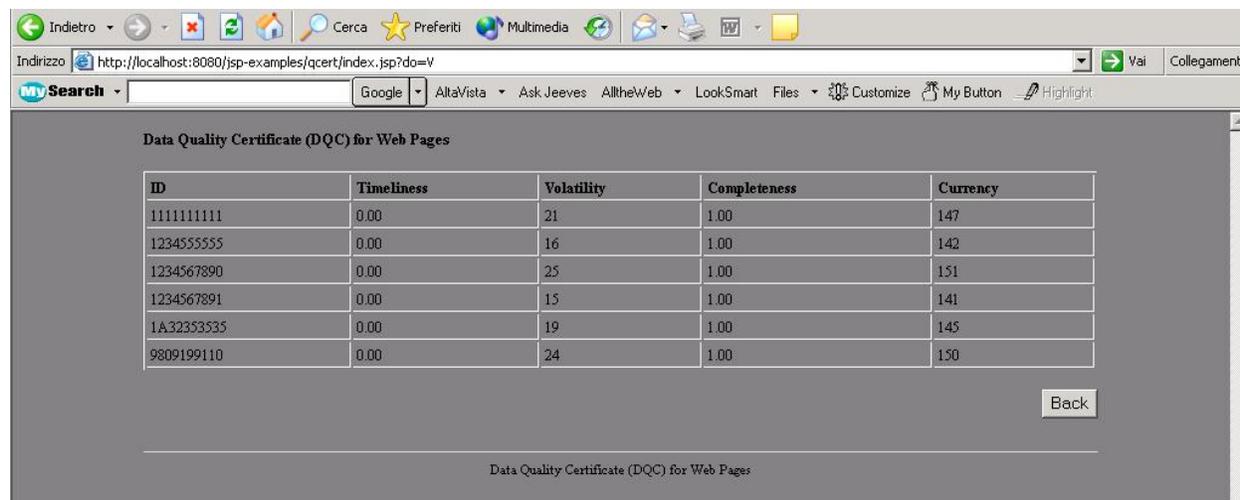


Figure 4 - Data Quality values associated with the objects contained in a Web page

When users request the certification of a page, the results of the certification process are shown with traffic light colours: a green page with aggregate data quality values if the certification process is successful (Figure 5), a red page otherwise (Figure 6). In both cases, the properties of the page and the indication of the number of the certifiable instances are provided.

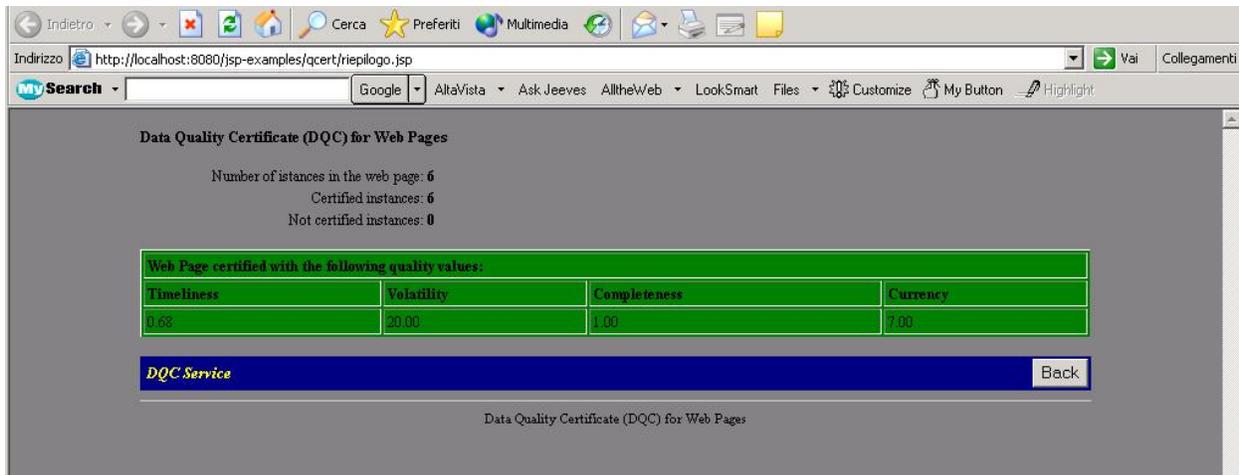


Figure 5 - Certified Data Quality values

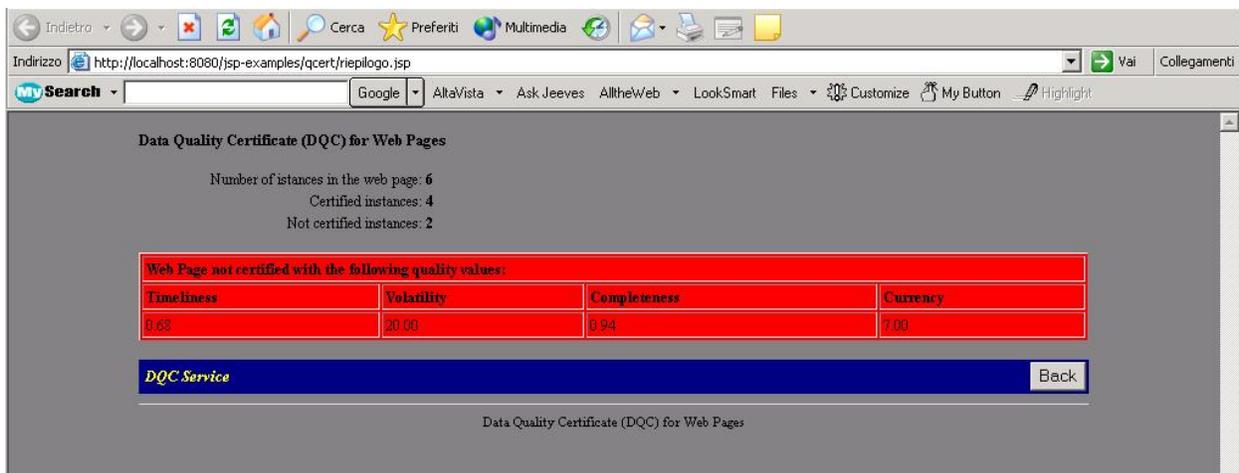


Figure 6 - Not certified Data Quality values

All data and data quality values are included in the certificate that is the described by the following OWL schema:

```
<owl:Class rdf:ID="CertificateX509v3">
  <owl:subClassOf rdf:resource="#Data_Quality_Certificate_Owl_Schema" />
</owl:Class>
```

Data quality dimensions are formally defined as attributes of the Digital Certificate:

```
<owl:ObjectProperty rdf:ID="timeliness">
  <rdfs:subProperty rdf:resource="#TBSCertificateExtensionsSubjectDirectoryAttributes"/>
</owl:ObjectProperty>
```

The values associated with data quality dimensions are defined by using corresponding tags, for example:

```
<timeliness>....</timeliness>
```

Finally, flags for *data security*, *metadata security* and *digital certificate* must be provided in the certificate.

The data packet to be signed for the type 8 DQC is a text file, such as:

```
-----BEGIN DATA UNIT
      Sequence of (Key, Data, Value)
-----BEGIN CERTIFICATE
      (it includes all definitions and properties described above)
-----END CERTIFICATE
```

In addition, the certificate has associated all X509v3 specific fields, such as the ones indicating the validity properties of the certificate, its digital signature, and so on, that are also represented using the OWL language.

7. CONCLUSIONS

In this paper we have presented a certification model for data quality over the Web which could help users to understand the reliability of the information contained in a Web page. According to this model, a Web page is certified only if quality requirements are satisfied. Quality requirements can be tailored to different types of services and users by means of a user-oriented data quality representation model and a corresponding formal representation of digital quality certificates for Web pages. The approach has been verified on a test Web page for the completeness and timeliness data quality dimensions.

Future work will complete the implementation of data quality dimensions. The certification tool will be verified with a broader variety of test cases for an overall assessment of the approach. In particular, the certification of more complex Web pages including data from multiple data sources as well as less structured information such as text or multi-media files will be considered. Other approaches to quality assessment along multiple dimensions will be experimented in order to evaluate the sensitivity of certification results.

ACKNOWLEDGMENTS

This work has been partially supported by the Italian FIRB Project MAIS.

REFERENCES

- [1] Ballou, D. P., Wang, R., Pazer, H.L., Tayi, G.K. Modelling Information Manufacturing Systems to Determine Information Product Quality. *Management Science*, vol. 44, no. 4 (April 1998).
- [2] Berners-Lee, T. *Relational Databases on the Semantic Web*. Technical Report 2002. Available on <http://www.w3.org/DesignIssues/RDB-RDF.html>.
- [3] Bovee, M., Srivastava, R.P., Mak, B. A Conceptual Framework and Belief- Function Approach to Assessing Overall Information Quality. *Proceedings of the Sixth International Conference on Information Quality*, Boston, MA 2001.
- [4] Cappelletto, C., Francalanci, C., Missier, P., Pernici, B., Plebani, P., Scannapieco, M., Virgillito, A. DL2: Presentation of Metadata and of the Quality Certificate. *DaQuinCIS Project Report*.
- [5] Cappelletto, C., Francalanci, C., Pernici, B., Plebani, P., Scannapieco, M. Data Quality Assurance in Cooperative Information Systems: a Multi-dimension Quality Certificate. *Proceedings of the International Workshop on Data Quality in Cooperative Information Systems (DQCIS '03)*, Siena, Italy, 2003.
- [6] Ceri, S., Fraternali, P., Bangio, A., Brambilla, M., Comai, S., Matera, M. *Designing Data Intensive Web Applications*. Morgan Kaufmann Publisher, 2003.
- [7] Dwaine, E.C. *SPKI/SDSI HTTP Server / Certificate chain discovery in SPKI/SDSI*. Master Thesis of

- Engineering in Computer Science and Engineering. Report. Massachusetts Institute of Technology, 2001.
- [8] Elmasri, R., Navathe, S.B. *Fundamentals of Database Systems*. Addison Wesley, Redwood City, CA (USA), 1994.
- [9] Gil, Y., Ratnakar, V. *A comparison of (Semantic) Markup Languages*. Technical Report. USC Information Sciences Institute and Computer Science Department. CA, USA. 2001.
- [10] Heflin, J.D. *Towards the Semantic Web: knowledge representation in a dynamic, distributed system*. Dissertation for the degree of Doctor of Philosophy. Graduate School of the University of Maryland, College Park. 2001.
- [11] Horrocks, I., Patel-Schneider, P.F. Three theses of representation in the Semantic Web. *Proceedings of the International World Wide Web Conference (WWW 2003)*, Budapest, Bulgaria.
- [12] Jarke, M., Lenzerini, M., Vassiliou, Y., and Vassiliadis, P. *Fundamentals of Data Warehouse*. Berlin: Springer-Verlag 2000.
- [13] Kohnfelder, L.M. *Towards a practical Public-key Cryptosystem*. Bachelor's Thesis, EECS Dept., Massachusetts Institute of Technology, 1978.
- [14] Meyen, D.M., Willshire, M.J. A Data Quality Engineering Framework. *Proceedings of the Conference on Information Quality*. Cambridge, MA. 1997, pp. 95-116.
- [15] Orr, K. Data Quality and Systems Theory. *Communications of the ACM*, 41, 2 (February 1998).
- [16] Pernici, B., Scannapieco, M. Data Quality in Web Information Systems. *Journal on Data Semantics I*. Springer-Verlag, Berlin, 2003, pp.48-69.
- [17] PGP Team. *PGP freeware*. <http://web.mit.edu/netwok/pgp.html>.
- [18] Redman, T.C. *Data Quality for the Information Age*. Artech House, 1996.
- [19] Spertus, E., Stein, L.A. Just in Time Databases and the World Wide Web. *Proceedings of the seventh international conference on Information and knowledge management*, Bethesda, Maryland, United States, 1998, pp. 30 – 37.
- [20] The Internet Society. *SPKI Certificate Theory*. Available on line <http://www.faqs.org/ftp/rfc/pdf/rfc2693.txt.pdf>.
- [21] W3C. *Web architecture: extensible languages*. W3C Note. 1998. <http://www.w3.org/DesignIssues/Extensible.html>.
- [22] Wang, R.Y., Reddy, M.P., Kon, H.B. Toward quality data: An attribute-based approach. *Decision Support Systems, Special Issue on information technologies and systems*, vol. 13, no.3-4 (March 1995), pp. 349-372.
- [23] Wand, Y., Wang, R.Y. Anchoring Data Quality Dimensions in Ontological Foundations. *Communication of the ACM*, vol. 39, no. 11 (1996).
- [24] Wang, R.Y. A Product Perspective on Total Data Quality Management. *Communications of the ACM*, vol. 41, no.2 (February 1998).
- [25] Wang, R.Y., Strong, D.M. Beyond Accuracy: What Data Quality Means to Data Consumers, *Journal of Management Information Systems*, vol.12, no.4 (Spring 1996), pp.5-34.
- [26] World Wide Web Consortium, W3C. Web site: www.w3.org.

APPENDIX A – COMPARISON AMONG XMLS, RDFS AND OWL

Representation parameters	Sub-parameters	XML Schema	RDF Schema	OWL
Context		Yes	Yes	Yes
Classes	Classes and Properties	It does not have an appropriate semantics. However the is-a hierarchy could be defined through the XML hierarchical structure.	rdfs:Class and rdfs:Property	owl:Class, and owl:ObjectProperty owl:DatatypeProperty
	Inheritance	No	rdfs:subClassOf rdfs:subPropertyOf	rdfs:subClassOf owl:equivalentClass owl:disjointWith rdfs:subPropertyOf
Property constraints	Domain	Implicit: defined with the attribute	rdfs:domain (global, it is valid for all the instances that have a specific property, irrespective of their class)	rdfs:domain (global)
	Range	Implicit: attribute value. It is local for the properties of parents but global for all siblings.	rdfs:range (global)	rdfs:range (global) owl:Restriction owl: onProperty (local)
	Cardinality	minOccurs, maxOccurs (local)	No	owl:minCardinality owl:maxCardinality owl:Cardinality (local) owl:FunctionalProperty owl:InverseFunctionalProperty (global)
	Values: Default	Yes	Yes	Yes
	Values: Enumeration	enumeration	No	owl:oneOf
	Values: sorting	It is implicit in XML	It derives from XML	It derives from RDFS
	Basic data type		Yes	Yes (all types of XML Schema Datatypes)
Instances		Yes	rdf:ID	rdf:ID
Instructions type	Negation	No	No	owl:ComplementOf
	Conjunction	No	rdfs:subClassOf	owl:IntersectionOf
	Disjunction	No (only union)	No	owl:DisjointWith with owl:unionOf
Property type	Inverse	No	No	owl:inverseOf
	Transitive	No	No	owl:TransitiveProperty
	Equivalency	No	No	owl:EquivalentProperty
	Symmetry	No	No	Owl:SymmetricProperty
Definitions	Subclass belongs to a class	No	rdfs:subClassOf	rdfs:subClassOf owl:sameAs owl:equivalentClass
	An instance belongs to a class	No	rdf:type,rdfs:member, rdf:first,rdf:rest	See RDFS and owl:oneOf
Predefined data sets	Closed lists	No	rdf:List	rdf:List, owl:oneOf
	Sorted sets	No	rdf:Seq	rdf:Seq