DQXSD: AN XML SCHEMA FOR DATA QUALITY
An XSD for Supporting Data Quality in XML

Eugenio Verbo, Ismael Caballero
Soluziona Consultancy and Technology
UCLM-Soluziona Research and Development Institute
Ronda de Toledo s/n – 13004 Ciudad Real, Spain

Mario Piattini
ALARCOS Research Group
Information Systems and Technology Departament
UCLM-Soluziona Research and Development Institute
Paseo de la Universidad 4 s/n – 13071 Ciudad Real, Spain

Keywords: Data Quality, Data Quality Dimensions, Data Quality measures, XML, quality attributes.

Abstract: Traditionally, data quality management has mainly focused on both data source and data target. Increasingly, data processing to get a data product need raw data typically distributed among different data sources. However, if data quality is not preserved when transmitted, resulting data product and consequent information will not be of much value. It is necessary to improve exchange methods and means to get a better information process. This paper focus on that issue, proposing a new approach for assuring and transmitting data quality in the interchange. Using XML and related technologies, a document structure that considers data quality as a main topic is defined. The resulting schema is verified using several measures and comparing it to the data source.

1 INTRODUCTION

Nowadays, organizations structure is usually spread on different locations, which requires to distribute organizational data storage in order to achieve better performance. On the other hand, Service Oriented Architectures are being consolidated. These services typically provide data in an XML format in order to be easily transmitted. Both scenarios suppose new challenges as data replication and data integration. One of our concerns is to study these challenges from the point of view of data quality.

In (Strong, 1997) the ten main problems for data quality are outlined and justified. Two of them are directly involved in distributed systems: a) multiple sources of the same data produce different values and b) distributed heterogeneous systems lead to inconsistent definitions, formats, and values.

Last years, XML has been intensively used up to become the main standard technology for data exchanging between distributed systems. Using XML, structured documents can be described, making their retrieval more efficient and effective. We could use it as a means/media for assessing and improving information quality, taking advantage of its related technologies as XSLT for easy processing of XML documents, and the restrictions model of XML Schema to define correct value patterns.

1.1 Data Quality Issues

A tipical example of a data quality problem consists of having different values for the same data stored in several sources. Suppose a decision-support system which access to those sources and analize them. With no additional information, the system only know that the data has different values but it cannot decide which is the correct one. If the system could assess the quality of each source and only accept those over a threshold, it would process the data with higher quality and, in consequence, produce better results.

Verbo E., Caballero I. and Piattini M. (2007).
DQXSD: AN XML SCHEMA FOR DATA QUALITY - An XSD for Supporting Data Quality in XML.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - DISI*, pages 359-364
DOI: 10.5220/0002381603590364

There are a lot of researching lines trying to explain what data quality is. Most of authors have drawn the conclusion that "fitness for use" is probably the best definition for the term. Fitness implies that a set of special and specific characteristics must be observed in the data as to say it can be used to get sound information. These characteristics have been named as Data Quality Dimensions. (Strong, 1997) propose a set of them which could be tailored for a wide range of contexts.

From these Data Quality Dimensions, several measures have to be defined to get a quantitative idea of how good a piece of data is. These measures (see nomenclature about SMO in (García, 2005)), also called metrics, must be defined to rightly manage data quality. (Lee, 2006) presents some of these measures. A data quality management team must use those measures to improve the data quality. The most used methodology for this goal is TDQM, mainly described in (Wang, 1998).

1.2 Addressing the Problems

Once brought to the context the main foundations of data quality, we want to address the problem we have posed. Supposing that data is currently stored in a database, and several data quality dimensions and measures have been defined. The problem is *How can data quality be assured when data flows from sources to targets?*

The answer comes with the technology on which data product mainly trips from sources to targets: XML. So the idea is to create an XML structure that can give the necessary support for transmitting data quality concerns used in the source to the targets in order to be used them to maintain the quality of the data products.

The remainder of the paper is structured as follows: section 2 present the main foundations of our work and design keys to elaborate our proposal. Section 3 shows several training examples of the proposal. Finally, section 4 outcomes several conclussions and future researching lines.

2 DQXSD: AN XML SCHEMA FOR DATA QUALITY

Data quality, as being quality, can be also studied from two points of view:

- **Expected Data Quality:** users expect that raw data and product data have a set of data quality dimensions like accuracy, free-of-error and so on. This kind of quality could almost always be evaluated without user interaction, for instance from metadata and quantity of stored data for each data set.
- **Required Data Quality:** users need and require that raw data and product data present specific and context-dependant characteristics which can be only evaluated by taking into account the judgement of the user, so user interaction is required, for instance, to provide a value which can be used as a basis or threshold to determine whether a data is good or not. The provided values must have been stored anywhere, and it could be necessary to be transmitted together with the data product. The way in which databases must be prepared to accept these values is given for the Data Quality Requirements in (Wang, 1995).

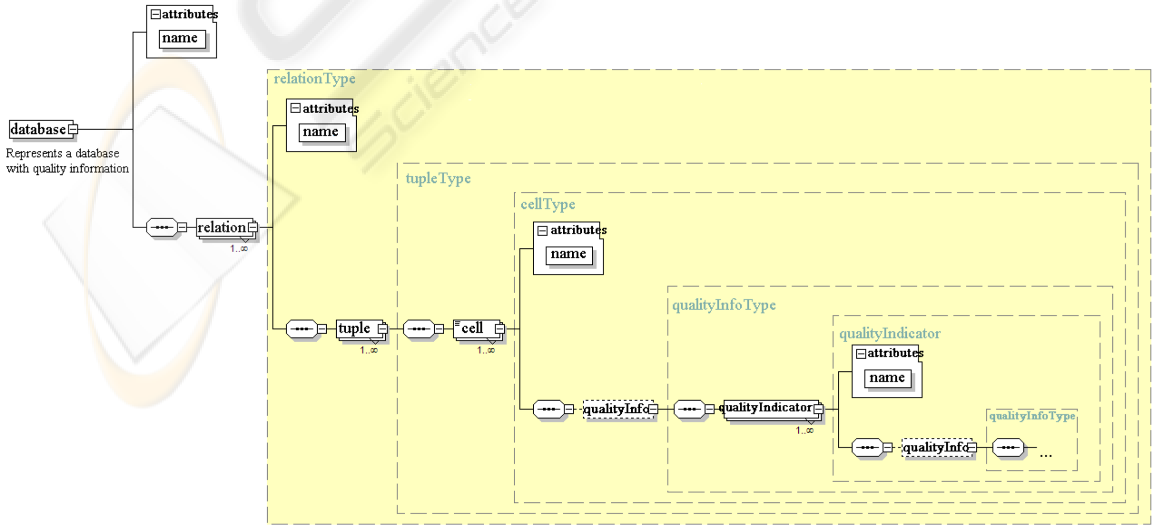


Figure 1: DQXSD Structure.

In order to grasp the special needs of XML documents quality, we are going to use more specific terms from now on:

- **External quality:** properties related to required data quality. It answers the question of what data is exchanged. Usually, an external agent must provide the information needed to address this issue.
- **Internal quality:** it deals with expected data quality and answers the question of how data is exchanged, i.e., the XML document structure. It is usually assessed applying measures on the document.

2.1 External Quality

In (Wang, 1995) the relational model is extended with an attribute-based data model to store data related to specific data quality dimensions which can let organizations achieve higher data quality. Since the attribute value of a cell is the basic unit of manipulation, it is necessary to tag quality information at the cell level.

Its principles are built over the notion of quality indicator. A quality indicator provides objective information about the characteristics of data and its manufacturing process.

It develops a mechanism to facilitate the linkage between an attribute and its immediate quality indicators. This mechanism is developed through the *quality key* concept. An attribute in a relation scheme is expanded into an ordered pair, called a *quality attribute*, consisting of the *attribute* and a *quality key*. The quality key is a reference to the underlying quality indicators. It also allows to detail a quality indicator linking it to a set of quality indicators. To achieve data integrity, an attribute value and its corresponding quality indicators must be treated as an atomic unit.

Suppose an organization has a database schema with associated quality information and they want to interchange its content with their partners. It would be much easier if there were a standard format to do so. In this point is when DQXML comes into play. Our work provides support for that model in XML.

Since XML is the preferred technology for data exchanging, it becomes the optimum choice. DQXML structure is defined with XML Schema and captures the requirements of the attribute-based approach. Figure 1 shows graphically the structure of the schema.

Table 1: Database and DQXML measures comparison.

Measure meaning	**Database measure name**	**DQXML measure name**
Effort necessary to retrieve all the information related to a component	DRT	DDQT
Information fragmentation degree	RD	RD
Quantity of information about a component directly accessible.	NA	NA
Cohesion of the system	COS	COS
Wasted communication bandwidth	-	NEE, NEA
Size of the system	-	NN
Number of references between the components of the system	-	NArc
Structural complexity of the system	-	SC_{XML}

The main element is *database*, which represents the whole database we want to exchange. It is composed by a sequence of *relations* elements. Each *relation* models a table of the DB scheme and is, in turn, formed by a set of *tuples*, which are divided in *cells*. In addition to its respective values, inside a *cell* there can be a *qualityInfo* element that specifies quality information and is detailed with a set of *qualityIndicator* subelements that model a single quality indicator value. Note that a quality indicator can have nested quality information as well.

This schema defines what we have called **XSD for Data Quality (DQXSD)**. It preserves external quality as data associated with quality indicators is carefully treated in the document scheme. Moreover, it is especially useful in the sense that there are already several database management systems with XML support in their queries. They also allows to specify an XSD that defines the structure that the results must follow so the translation of the data stored in the database to DQXSD structure would be quite immediate.

We would like to highlight the fact that quality indicators are intended to contain useful data for assessing data quality on a certain quality dimension. The purpose of this model is acquisition and transmission of data. It should not include any measurement result because measurement methods may vary depending on the role of the system user and its context. We will treat measurement in the next section.

From now on, we will write DQXML to refer to an XML document validated against DQXSD.

2.2 Internal Quality

After extracting database records according to the DQXSD structure, we obtain an XML document that preserves external quality from the original data source but, *what happens to the efficiency and accuracy of the data representation, i.e., internal quality?* In order to assess this issue objectively we have applied a measurement approach.

In (Ivan, 1998) a general definition for data measures is given along with examples of use. (Piattini, 2001) proposes some internal measures to measure relational databases which influence its complexity. Centered in XML documents, in (Díaz, 2003) a set of measures are proposed and implemented in a measurement tool. However, there is not much research work in data quality measurement oriented to XML documents.

Our idea is to adapt validated database measures to XML documents and compare them in order to demonstrate that the results are similar. In (Piattini, 2001) and (Calero, 2001) the following measures are defined:

- *Depth of the referential tree*: the DRT of a table A (DRT(A)), is the length of the longest referential path from the table A, counted as the number of arcs on the path and considering cycles only once.
- *Referential degree*: the RD of a table A (RD(A)), is the number of foreign keys in the table A.
- *Number of attributes*: the NA of a table A (NA(A)), is the number of attributes of the table A.
- *Cohesion of the Schema*: the COS of a schema S, is the sum of the square of the number of tables in each not connected component in the schema graph.

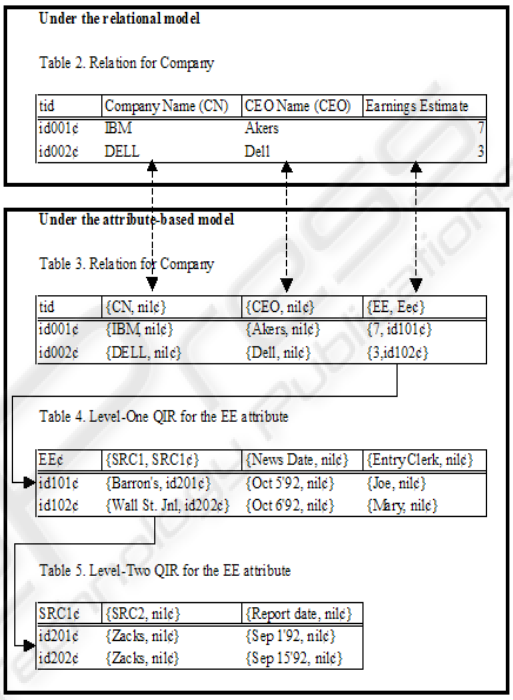
We have adapted those measures to XML documents with DQXSD structure:

a. *Depth of the DQXML tree*: the DDQT of a DQXML (DDQT(D)), is the number of nested *qualityInfo* elements inside a *cell* plus one. It is equivalent to DRT for relational databases.

b. *Referential degree*: the RD of a DQXML (RD(D)), is the number of *qualityInfo* subelements that *cell* elements of a *relation* contains.

c. *Number of attributes*: the NA of a *relation* element R in a DQXML (NA(R)), is the maximum number of *cell* elements that each *tuple* element contains.

d. *Cohesion of the Schema*: the COS of a DQXML (COS(D)), is the square of the number of *relation* elements that are not connected in the scheme graph.



Under the relational model

Table 2. Relation for Company

tid	Company Name (CN)	CEO Name (CEO)	Earnings Estimate
id001¢	IBM	Akers	7
id002¢	DELL	Dell	3

Under the attribute-based model

Table 3. Relation for Company

tid	{CN, nil¢}	{CEO, nil¢}	{EE, Ee¢}
id001¢	{IBM, nil¢}	{Akers, nil¢}	{7, id101¢}
id002¢	{DELL, nil¢}	{Dell, nil¢}	{3,id102¢}

Table 4. Level-One QIR for the EE attribute

EE¢	{SRC1, SRC1¢}	{News Date, nil¢}	{EntryClerk, nil¢}
id101¢	{Barron's, id201¢}	{Oct 5'92, nil¢}	{Joe, nil¢}
id102¢	{Wall St. Jnl, id202¢}	{Oct 6'92, nil¢}	{Mary, nil¢}

Table 5. Level-Two QIR for the EE attribute

SRC1¢	{SRC2, nil¢}	{Report date, nil¢}
id201¢	{Zacks, nil¢}	{Sep 1'92, nil¢}
id202¢	{Zacks, nil¢}	{Sep 15'92, nil¢}

Figure 2: Database with quality information - taken from (Wang, 1995).

In addition to the previous measures, some specific ones for XML from (Díaz, 2003) has been included:

e. *Number of empty elements*: the NEE of an element A (NEE(A)) is the number of empty elements that are children of the element A.

f. *Number of empty attributes*: the NEA of an element A (NEA(A)) is the number of empty attributes of the element A.

g. *Number of nodes*: the NN of a DQXML (NN(D)) is the number of nodes needed to represent the document graph considering as a node any element, attribute or element value.

h. *Number of arcs*: the NArc of a DQXML (Narc(D)) is the number of arcs needed to represent the document graph. An arc is a relation between parent and children elements, element attributes and element values.

i. *Structural complexity*: the SC_{XML} of a DQXML is:

$$SC_{XML} = NArc - NN + 1 \qquad (1)$$

Taking advantage of XML related technologies, XSLT can be applied to calculate many of this measures, simplifying the construction of a management tool. Consequently, a second XML document with the measurement results would be generated for later viewing or processing. The most remarkable benefits of this approach are portability, interoperability and programming language independence.

A brief summary of the measures applied to databases and to DQXMLs is shown in Table 1.

3 EXAMPLES

3.1 External Quality

To illustrate DQXSD usage, we have borrowed the theoretical training example shown in Figure 2 from (Wang, 1995) with the aim of adapting it to our model.

Suppose an organization has a database schema that contains a table like Table 2 (Tables 2-5 are embedded in Figure 2) with attributes like company name, CEO name and earnings estimate. Data may be collected over a period of time and come from a variety of sources. If the organization wanted to assess the believability of the data, the previous database schema should be adapted to the new quality requirements.

As a result, the original Table 2 is expanded into Table 3, which consists of the ordered pairs ({CN, nil¢}, {CEO, nil¢}, {EE, EE¢}). The "nil¢" indicates that no quality indicators are associated with attributes CN and CEO; whereas EE¢ indicates that EE has quality indicators associated.

Table 4 is a quality indicator relation for the attribute *Earnings Estimate* in Table 3 and Table 5 is a quality indicator relation for *SRC1* in Table 4.

First of all, we translate Table 3 to the DQXSD format without including quality indicators (text in normal font in Figure 3). In the resulting DQXML, there is only one *relation* element, which contains the two tuples that specifies companies data.

Later, we include the first level of quality indicators inside the *cell* element named "EE" (text in bold and italic in Figure 3). And, finally, we include the second level of quality indicators into the first level quality indicator *SRC1* (text in italic in Figure 3). For shortening, only one tuple and one cell have been included in Figure 3.

3.2 Internal Quality

The results of applying the measures explained in Section 2.2 can be consulted in Table 6.

First of all, we can see that equivalent measures DRT and DDQT have a similar value, 3.

```
<database>
  <relation>
    <tuple>
      <cell name="EE">
        7
        <qualityInfo>
          <qualityIndicator
              name="SRC1">
            Barron's
            <qualityInfo>
              <qualityIndicator
                  name="SRC2">
                Zacks
              </qualityIndicator>
              <qualityIndicator
                  name="Report
Date">
                1992/09/1
              </qualityIndicator>
            </qualityInfo>
          </qualityIndicator>
          <qualityIndicator
              name="News Date">
            1992/10/5
          </qualityIndicator>
          <qualityIndicator
              name="entryClerk">
            Joe
          </qualityIndicator>
        </qualityInfo>
      </cell>
      ...More cell elements...
    </tuple>
    ...More tuple elements...
  </relation>
</database>
```

Figure 3: DQXML with complete quality data.

For RD, the difference is in the measure method. For the quality database, we have three values: the result is 2 for the table "Company"; the result is also 2 for the table with the first level of quality indicators for attribute EE; and the result is 0 for the table with the second level of quality indicators for attribute EE. However, for DQXML we only have one value, 4, for the entire document. This occurs because the quality information embedded in cell elements of the DQXML is divided in three tables in the database. If we put together the values of every table of the schema, we get the same result, 4.

For the measure *number of attributes* the result is different as well. The explanation is that in the database, the storage medium does not differentiate between quality and raw data while DQXSD treats quality data adding semantic value that it did not have when stored in a database.

Table 6. Measurements results

Measure	Relational database	DQXML
DRT	3	
DDQT		3
RD	RD(Comp)=2 RD(EE_1)=2 RD(EE_2)=0	RD(R_1)=4
NA	NA(Comp)=4 NA(EE_1)=4 NA(EE_2)=3	NA(D)=4
COS	0	0
NEE	-	0
NEA	-	0
NN	-	73
NArc	-	72
SC_{XML}	-	0

The results of the measures *NEE* and *NEA* shows that the DQXML has high quality because it has no empty elements or attributes that waste bandwidth. Lastly, *NN* and *NArc* with their low values indicates that the DQXML has no excessive complexity, statement confirmed by SC_{XML}.

4 CONCLUSIONS AND FUTURE WORKS

Traditionally, data quality has been only applied to data stored in databases as being raw data for manufacturing data products. This approach is clearly out of date because data exchanging is continuously getting more important in parallel to the consolidation of Service Oriented Architectures.

Static data quality issues must also be propagated when transmitted. To give the necessary support to this goal, we define a new document structure, DQXSD based on the most important technology for information exchanging, XML. To define it, XML Schema is used.

DQXSD helps to capture quality data stored in a database schema and translate it to a proper format ready to be transmitted.

To prove the data quality preservation through that process, several measures for DQXML documents have been developed and compared to database equivalents getting satisfactory results.

Although the results presented in this paper are oriented to capture quality data stored in relational databases, DQXSD could be easily adapted to other storage models due to the flexibility of the technologies used for its definition.

ACKNOWLEDGMENTS

This research is part of the FAMOSO and ESFINGE projects supported by the Dirección General de Investigación of the Spanish Ministerio de Ciencia y Tecnología (Ministry of Science and Technology)(TIC2003-07804-C05-03).

REFERENCES

Calero, C., Piattini, M. & Genero, M., 2001. *Metrics for controlling Databases Complexity*, Becker, S.

Díaz, E., 2003. *Herramienta para la gestión de métricas en documentos XML*. Departamento de Tecnologías y Sistemas de Información, Escuela Superior de Informática de Ciudad Real, Universidad de Castilla-La Mancha.

Fran, W. & Simeon, J., 2003. Integrity constraints for XML. *Journal of Computer and System Sciences*.

García, F., Bertoa, M. F., Calero, C., Vallecillo, A., Ruiz, F., Piattini, M. & Genero, M., 2005. Toward a consistent terminology for software measurement. *Information and Software Technology,* 48**,** 631-644.

Ivan, I., Parlog, O., Oprea, P., Nosca, G. & Ivan, A.-A., 1998. Data Metrics. In *IQ 1998, Conference on Information Quality*.

Klettke, Sheneider, M. L. & Heuer, A., 2002. *Metrics for XML Document Collections*. Database Research Group, University of Rostock, Germany.

Lee, Y. W., Pipino, L. L., Funk, J. D. & Wang, R. Y., 2006. *Journey to Data Quality*, The MIT Press.

Piattini, M., Calero, C. & Genero, M., 2001. Table Oriented Metrics for Relational Databases. *Software Quality Journal*.

Strong, D. M., Lee, Y. W. & Wang, R. Y., 1997. Data Quality in Context. *Communications of the ACM*.

Strong, D. M., Lee, Y. W. & Wang, R. Y., 1997. 10 Potholes in the Road to Information Quality. *IEEE Computer*.

Wang, R. Y., 1998. A Product Perspective on Total Data Quality Management. *Communications of the ACM*.

Wang, R. Y., Reddy, M. P. & Kon, H. B., 1995. Toward quality data: An attribute-based approach. *Decision Support Systems*.