DQRDFS
Towards a Semantic Web Enhanced with Data Quality

Ismael Caballero, Eugenio Verbo
Indra Software Labs, UCLM - Indra Research and Development Institute
Ronda de Toledo s/n – 13004 Ciudad Real, Spain

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

Keywords: Data Quality, Semantic Web, DQRDFS, DQMetadata, ISO 15939, Data Quality Measurement, Quantity of Data Quality.

Abstract: Nowadays, data is of critical importance as a resource. Using data of poor quality can be the source of several problems when developing a project. The World Wide Web is currently the main showcase for a vast amount of data. It would be desirable that machines can process the quality of the data contained on the Web Documents. This paper introduces a new view of the Semantic Web based on the concept of Quantity of Data Quality (QDQ), in which Data Quality issues will be used as a basis to enable machines to process the Semantic Web Documents for different activities like information retrieval or document filtering. This view can open new challenges in Semantic Webs oriented to improve users' satisfaction with the Internet.

1 INTRODUCTION

Advances in software technologies and networks have lead to the development of new Information Systems running on the Web, as a means for organizations to be as close as possible to all their customers, stakeholders and other organizations. This kind of system allows companies to publish, via the web, documents containing data related to the tasks which must be carried out by specific users. If data is not of high enough quality, then users cannot correctly complete their projects. So it is important to take care with data quality (hereafter DQ) in order to ensure that users achieve a better standard of project. Up to now, users have been responsible for assessing the quality of the data, since they are the only ones who are able to understand it and its meaning.

In their definition of Semantic Web, (Bernes-Lee et al., 2001) state that "*the Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation*". This definition opens the possibility for machines to understand data contained in web documents: if they can understand it, they can also manage the quality of it. Furthermore, machines could apply this managerial capability to doing tasks like discriminating documents attending to their data quality levels before deploying them to users.

But in order to turn these proposals into reality, some value must be added to the Semantic Web, and to the way in which Semantic Web Documents can be performed.

This paper has a twofold goal: (1) to provide readers with a brief background of DQ (section 2) and (2) to show how we have applied DQ fundamentals in order to enhance the quality of Web Documents for Semantic Web (section 3).

2 DATA QUALITY BACKGROUND

2.1 Data Quality Dimensions

Data is said to be of quality if it fits the purpose for which is used (Strong et al., 1997). One of the most

Caballero I., Verbo E., Calero C. and Piattini M. (2008).
DQRDFS - Towards a Semantic Web Enhanced with Data Quality.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 178-183
DOI: 10.5220/0001526701780183

interesting strategies for tackling the study of DQ for a context, is to break it down into "minor qualities" known as **DQ dimensions** (Lee et al., 2006). Each scenario requires some dimensions which best fit the use of the data. The sets of DQ Dimensions usable in a context are known as **DQ model**. Literature shows several examples of DQ models for specific scenarios: medical and healthcare (Al-Hakim, 2007), decision support system (Gendron and D'Onofrio, 2002), or web (Caro et al., 2007), to name a few. ISO, at this moment, is working on the draft of the ISO/IEC 25012 standard (ISO-25012, 2007), a part of the SQUARE family that will propose a DQ model for IS. In any case, the generic classification proposed by (Strong et al., 1997) has been widely used. These authors group DQ dimensions into four categories making reference to the point of view from which DQ can be observed: **Intrinsic DQ** (dimensions of accuracy, objectivity, credibility, reputation) refers to the quality of the data itself; **Accessibility DQ** category contains dimensions (accessibility, Access security) providing meaning about the extent to which data can be accessed; **Contextual DQ** (Relevancy, Value-Added, Timeliness, Completeness, Amount of data) refers to those DQ dimensions which deal with the use of specific data in a context; **Representational DQ** category (interpretability, ease of understanding, concise representation, consistent representation) is centred on those characteristics of the representation of data which make it usable. A more complete definition of the meaning of these dimensions can be found through DQ literature, the most interesting works being those proposed by (Batini and Scannapieco, 2006, Lee et al., 2006, English, 1999).

2.2 Measuring and Assessing DQ

Let us give the name *stakeholder* to any person or process involved in the use of the data or of resources which have data. Any stakeholder will need to assess how good a piece of data is for the task to be executed. We would like to highlight the difference between the concepts of "measuring DQ" and "assessing the DQ level" of a piece of data for a task. For each DQ Dimension belonging to the DQ model used for the assessment, some measurements must be taken.

Both measurement and assessment are going to depend on the intended use of the data and on the nature of the DQ dimension (which determine the measurement method (ISO/IEC, 2000)). For measuring, a base or a derived measure must be drawn. In this case, a measurement method or a measurement function is required. On the other hand, it could possibly be said that for the assessment and indicator might be enunciated.

According to literature, typical derived DQ measures have a measurement function based on the percentage of the Number of Data Units which do or do not satisfy a DQ criterion (Batini and Scannapieco, 2006, Lee et al., 2006). This fact confers to the measurement of a ratio scale (see Figure 1):

$$DQ_{Measure} = 1 - \frac{NumberOfDataUnitsNotSatisfyingADQCriterion}{TotalNumberOfDataUnits}$$

Figure 1: Typical DQ Measure.

In the formula of Figure 1, there are two base measures: *NumberOfDataUnitsNotSatisfyingACriterion* and *TotalNumberOfDataUnits*. The measurement method for both consists of counting a number of data units. For the second one, there is only one problem, which is counting all data units of the piece of data. In the first, the counting is limited to those data units affected by the criterion. A criterion is usually defined as a business rule to warranty the soundness of the data (English, 1999, Loshin, 2001, Wang, 1998). The result of deciding if the data unit satisfies the criterion can be "*True*" or "*False*". So, in order to obtain a value for the measure, a count of data units having obtained a "*true*" value must be done. But the intrinsic difficulty is addressed at defining how the data unit satisfies the criterion. Sometimes, some metadata is necessary for each piece of data to complete its meaning in order to be able to decide whether or not it satisfies the criteria. To make a decision, a rule based on this metadata is needed. This rule can consist of objectively or subjectively determining if the value of metadata belongs to a given domain. (Naumann and Rolker, 2000) identify the following as possible sources for values of metadata: a stakeholder, the information manufacturing process or even the same data store. Different authors in the DQ field agree that values for metadata coming from users are probably subjective, whereas the ones coming from the proper data stores are objective.

Having to add some metadata to the data, a new problem arises: how to attach the metadata to the data and how to store it conveniently. In (Wang et al., 1995), a possible solution for the relational model is proposed. It consists of tagging data: attach the DQ metadata as if it were another common attribute. It could be seen as a way of semantic annotation. (Caballero et al., 2007) propose another solution based on (Wang et al., 1995) for XML.

They propose an XML Schema named DQXSD that allows making such annotations for XML files (see figure 2): The **qualityData** is used as the root of the XML document; **Entity** is anything containing data (a relational schema, an XML documented). Each entity can have **attributes,** the DQ of which must be studied, like relational attributes or elements in XML files. Authors use the name **measurableConcept** for DQ Dimensions in order to align their model to ISO/IEC 15939 (ISO/IEC, 2000). Finally, for each *measurableConcept*, zero or more **DQMetadata-Attributes** are defined, and given a value which is used to assess the DQ level of each entity regarding the measurable concept.

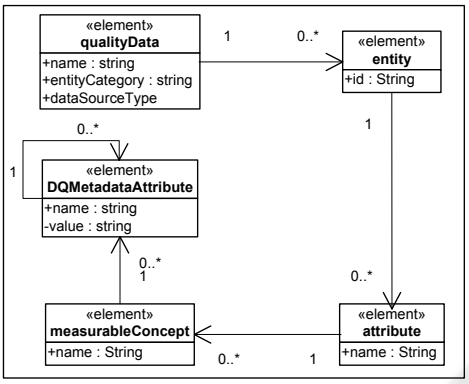

Figure 2: DQXSD (Caballero et al., 2007).

Next, we are going to explain how these fundamentals could be introduced into the Semantic Web.

3 DQRDFS: DQ AT SEMANTIC WEB

The main aim of this paper is to enable a new perspective in which machines can automatically process the quality of the data contained in the Semantic Web Documents in order to increase user's satisfaction with Semantic Web applications in tasks like information retrieval or semantic searches. This implies that machines need to measure and assess this data by making corresponding DQ semantic annotations in "*such way that can be used for more effective discovery, automation, integration and reuse across various applications* (Guha et al., 2003)"

For this reason, we are going to show how to integrate DQ issues into Semantic Web processing by following (Wang et al., 1995)'s ideas through the proposal of (Caballero et al., 2007), but adapted to RDF. In the "*traditional*" view, data on Semantic Web is modelled like a ***directed labelled graph***, wherein each node corresponds to a resource (subjects and objects) and each arc is labelled with a predicate. As a first approach to integrate DQ issues in Semantic Web, what we propose in this paper is to annotate RDF with values (metadata following DQ nomenclature or *DQMetadataAttribute* following (Caballero et al., 2007)) corresponding to DQ dimensions (*measurableConcepts*) which are of interest for the different *stakeholders*. This metadata might have an objective value used to compute a measure. Having measures for all best fitting DQ Dimensions, machines can process an assessment for the Semantic Web Document. This assessment represents the perception of the DQ for a Web Document of a stakeholder for a given application. We have named it as ***Quantity of DQ*** **(QDQ).**

The QDQ could be interpreted as a weight. It enables viewing the Semantic Web as a ***weighted directed graph*** for a specific task and stakeholder. This view will open new fields in machine-processing data having as basis DQ: for instance, Semantic Searchers can delimit the quantity of found results, showing to the users only those whose QDQ is within an acceptance threshold range (not only the "relevance"); or ordering the results according to a ranking model (Ding et al., 2005) based on DQ requirements. Another kind of application that can be improved is that oriented to automate a task, like the one described for (Bernes-Lee et al., 2001).

In order to achieve this goal, several challenges must be tackled: (1) identify which attributes must be studied from DQ point of view, (2) how to identify which are the proper *measurableConcepts* for those resources and how to identify the necessary metadata (when required) to make the measurements, (3) how to get values for that metadata and how to annotate them, and finally (4) how to compute the QDQ according to the perception of DQ of different groups of stakeholders through their selected dimensions.

3.1 Identify Attributes to be Annotated

The first step in order to enable DQ in Semantic Web consists of identifying from the Data Quality User Requirement Specification (DQ-URS) which elements need to be studied. The elements are related to the level of granularity at which the study is necessary. (Ding et al., 2005) identify the following levels of granularity according to the levels in which Semantic Web can be queried: RDF Database, Semantic Web Document (SWD), RDF subgraph or Semantic Web. According to DQXSD by (Caballero et al., 2007), our entities are going to

be the **RDF.** Their attributes are *Subject, Predicates, Object* and *Sentences.* In this approach we are going to focus only on **sentences,** our aim being to write DQ Sentences about sentences. This process is known as **reification** (Daconta et al., 2003).

For instance, let us consider the sentences "*Buddy owns business"* and "*business has-Website http://www.buddy.com"*-borrowed directly from (Daconta et al., 2003) and showed in figure 3-. Let's imagine that somebody could be interested in knowing, for instance, how reliable, reputable or timely the sentences are.

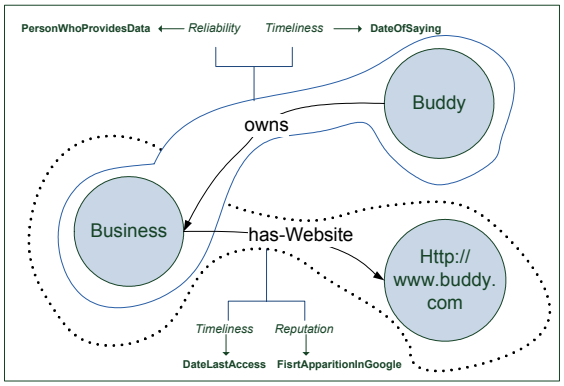

Figure 3: DQ Dimensions for different sentences.

3.2 Identify DQ Dimensions and Related Metadata

Once the attributes susceptible to study have been identified, the next step is to identify the DQ dimensions or measurable concepts that best fit the problem of assessing DQ for these elements. The easiest way to tackle this problem is to choose as a guide a suitable specific DQ Model (see section 2.1)

As previously mentioned, and according to the nature of each DQ Dimension, some metadata would be required. Sometimes, metadata can already be part of the RDF file or it may be necessary to add it. For instance, in Figure 2, a stakeholder could need two DQ dimensions (*Timeliness* and *Reliability*) to compute the QDQ of the sentence "*Buddy owns a Business*". Some metadata complementing the meaning of the *Reliability* dimension is required. Let us suppose that it has been decided that knowing the *PersonWhoProvidesData* can help to interpret and determine if the sentence is worthy or not. Please, note that on one hand we have the values corresponding to metadata for measuring a DQ Dimension by using a measurement method, and on the other hand we have the measures of the DQ Dimension used to calculate the QDQ for the predicate by aggregating those values through an indicator.

3.3 Getting and Annotating Values for Metadata

This is the great challenge since it implies three key aspects in measuring DQ:

(1) *Who must provide these values?*
(2) *How and where to store these values?* and
(3) *How to get a representative value for different values of the same* ***DQMetadataAttribute*** *for all stakeholders in order to calculate the QDQ?*

The main response to question (1) might be found in social annotations like those in Web 2.0 (e.g. del.icio.us or flickr) (Bao et al., 2007). This situation enables on one hand, the possibility of easily getting values for the same metadata, with all the connotations and backgrounds of each user. And on the other hand, it is possible to determine through users' experiences the relationships amongst the most important DQ Dimensions when assessing the DQ from their corresponding backgrounds in order to create specific DQ Models for each context.

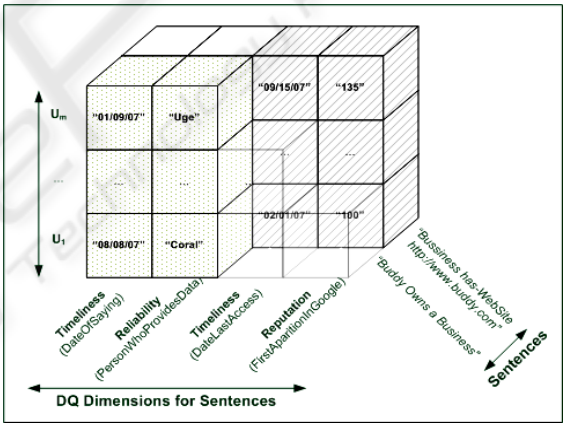

Figure 4: Conceptual Representation of DQ metadata.

To answer question (2), in Figure 4, a cube is shown with the following information: On the X axis, we have the DQ dimensions; the Y axis shows all users having annotated a value; the Z axis gathers all possible sentences of an RDF. Each individual block stores the value V_{ijk} for a metadata given for a DQ dimension D_i by a user U_j for the sentence S_k. So, column *i* contains values of metadata for the DQ dimension D_i, whereas the row *j* gathers the values given for a user u_j for all DQ dimensions involved in the evaluation of a sentence S_k. Since not all dimensions are implied at the same time in the calculus of QDQ for each sentence, not all values are required. Each block of figure 4 has been particularized with values for the example proposed in Figure 3. All these values must be stored together with the RDF in a RDF Server or in a XML

Database. The DQXSD proposed by (Caballero et al., 2007) is used for describing how to attach and store the value for metadata to their corresponding sentence (an attribute for DQXSD).

In order to make this DQXSD operative in this context, we have developed a counterpart RDF Schema, which is shown in Figure 5. This Schema is what we have named **DQRDFS.** As an example of its use, figure 6 shows how the sentence "*Buddy owns a Business"* and its corresponding *measurableConcepts* (see figure 3) can be represented by using the proposed DQRDFS.

```
<?xml version='1.0' encoding='UTF-8'?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
 xmlns:dq="'http://alarcos.inf-cr.uclm.es/dqrdf/1.0'"
 xmlns:rdfs="'http://www.w3.org/2000/01/rdf-schema ">
<rdfs:Class rdf:about="&dq;DQMetadataAttribute"
 rdfs:label="DQMetadataAttribute">
<rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdf:Property rdf:about="&dq;DQMetadataAttributes"
 rdfs:label="DQMetadataAttributes">
<rdfs:domain rdf:resource="&dq;measurableConcept"/>
<rdfs:range rdf:resource="&rdfs;Class"/>
</rdf:Property>
<rdfs:Class rdf:about="&dq;attribute"
 rdfs:label="attribute">
<rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdf:Property rdf:about="&dq;attributes"
 rdfs:label="attributes">
<rdfs:domain rdf:resource="&dq;entity"/>
<rdfs:range rdf:resource="&rdfs;Class"/>
</rdf:Property>
<rdfs:Class rdf:about="&dq;entity"
 rdfs:label="entity">
<rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdf:Property rdf:about="&dq;id"
 rdfs:label="id">
<rdfs:domain rdf:resource="&dq;entity"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdfs:Class rdf:about="&dq;measurableConcept"
 rdfs:label="measurableConcept">
<rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdf:Property rdf:about="&dq;measurableConcepts"
 rdfs:label="measurableConcepts">
<rdfs:domain rdf:resource="&dq;attribute"/>
<rdfs:range rdf:resource="&rdfs;Class"/>
</rdf:Property>
<rdf:Property rdf:about="&dq;name"
 rdfs:label="name">
<rdfs:domain rdf:resource="&dq;DQMetadataAttribute"/>
<rdfs:domain rdf:resource="&dq;attribute"/>
<rdfs:domain rdf:resource="&dq;measurableConcept"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&dq;nestedEntity"
 rdfs:label="nestedEntity">
<rdfs:domain rdf:resource="&dq;DQMetadataAttribute"/>
<rdfs:range rdf:resource="&rdfs;Class"/>
</rdf:Property>
<rdf:Property rdf:about="&dq;value"
 rdfs:label="value">
<rdfs:domain rdf:resource="&dq;DQMetadataAttribute"/>
<rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
</rdf:RDF>
```

Figure 5: DQRDFS: a RDF Schema supporting DQ.

For the last question, it is important to realise that having several values for each metadata, it is necessary to give a global representative value for all provided values. If given values are numbers, an example of this representing global value could be an arithmetic mean; should these values for metadata be subjective linguistic labels, then an aggregation method is required, like the one proposed by (Herrera-Viedma et al., 2006).

```
<?xml version="1.0"?>
<rdf:RDF [...]
 xmlns:dq="http://alarcos.inf-cr.uclm.es/ontologies/dqmo#" >
<rdf:Description>
  <earl:asserts rdf:parteType='Resource'>
    <rdf:subject>
      <rdf:Description rdf:about="#Buddy"></rdf:subject>
    <rdf:predicate>
      <RDFNSId:owns><RDFNSId='#business'>
    </rdf:predicate>
  </earl:asserts>
  <dq:entity rdf:about="&dq;entity_Instance"
      dq:id="entity1"
      rdfs:label="entity_Instance">
      <dq:attributes rdf:resource="&dq;sentence_Instance"/>
  </dq:entity>
  <dq:attribute rdf:about="&dq;sentence_Instance"
    <dq:name="sentence1" rdfs:label="sentence_Instance">
      <dq:measurableConcepts
        df:resource="&dq;measurableConcept_Instance_1"/>
      <dq:measurableConcepts
        rdf:resource="&dq;measurableConcept_Instance_2"/>
  </dq:attribute>
  <dq:measurableConcept
rdf:about="&dq;measurableConcept_Instance_1"
      dq:name="Reliability"
      rdfs:label="measurableConcept_Instance_1">
      <dq:DQMetadataAttributes
          rdf:resource="&dq;metadataAtt_Instance_1"/>
  </dq:measurableConcept>
  <dq:measurableConcept
rdf:about="&dq;measurableConcept_Instance_2"
      dq:name="Timeliness"
      rdfs:label="measurableConcept_Instance_2">
      <dq:DQMetadataAttributes
          rdf:resource="&dq;metadataAtt_Instance_2"/>
  </dq:measurableConcept>
    <dq:DQMetadataAttribute
rdf:about="&dq;metadataAtt_Instance_1"
      dq:name="PersonWhoProvidesData"
      rdfs:label="metadataAtt_Instance_1">
      <dq:value> "Uge"</dq:value>
      <dq:value> "Coral"</dq:value>
  </dq:DQMetadataAttribute>
  <dq:DQMetadataAttribute
rdf:about="&dq;metadataAtt_Instance_2"
      dq:name="DateOfSaying"
      rdfs:label="metadataAtt_Instance_2">
      <dq:value> "01/09/07"</dq:value>
      <dq:value> "08-08-2007"</dq:value>
  </dq:DQMetadataAttribute>
 </rdf:Description>
</rdf:RDF>
```

Figure 6: An example of DQRDF.

3.4 Calculating QDQ

Once representative values for each measurable concept have been obtained, the next step is to calculate the QDQ for each sentence. As can be seen

in figures 3 and 4, for each QDQ a set of DQ Dimensions is involved. The value of QDQ, as previously said, must be calculated by aggregating the corresponding measures for the required *measurable Concepts* as an Indicator (ISO/IEC, 2000), taking into account the relationships between the different DQ Dimensions. An interesting proposal for calculating the QDQ, which takes into account the possible relationship between DQ Dimensions, is the one by (Caro et al., 2007), in which a Bayesian Network (BN) is implemented for their own DQ model for calculating the level of Representational DQ of Educational Web Portals.

4 CONCLUSIONS

This paper has introduced some fundamentals of DQ and has highlighted the importance of annotating DQ issues of Semantic Web in order to have an improved web through QDQ Concept. This QDQ enables a view of the Web as a weighted directed graph which would open new challenges in machine-processing Semantic Web Documents in order to optimize users' satisfaction. In the future we will deal with refining and validating the DQRDFS. A study of how to extend the proposal to the remainder of the elements of RDF is also planned.

ACKNOWLEDGEMENTS

This research is part of the projects ESFINGE (TIN2006-15175-C05-05) supported by the Spanish Ministerio of Educación y Ciencia and MECENAS (PBI06-0024) supported by the Consejería de Educación y Ciencia of Junta de Comunidades de Castilla – La Mancha.

REFERENCES

Al-Hakim, L. (2007). Procedure for Mapping Information Flow: A case of Surgery Management Process. In *Information Quality Management: Theory and Applications* (Ed, Al-Hakim, L.) Idea Group Publishing, Hershey, PA, USA, pp. 168-188.

Bao, S., Wu, X., Fei, B., Xue, G., Su, Z. and Yu, Y. (2007) Optimizing Web Search Using Social Annotations. In *WWW'05 Proceedings*. ACM Press, Banff, Alberta, Canada, pp. 501-510.

Batini, C. and Scannapieco, M. (2006) *Data Quality: Concepts, Methodologies and Techniques*, Springer-Verlag Berlin Heidelberg, Berlin.

Bernes-Lee, T., Hendler, J. and Lassila, O. (2001) The Semantic Web. *Scientific American.*

Caballero, I., Verbo, E. M., Calero, C. and Piattini, M. (2007) A Data Quality Measurement Information Model based on ISO/IEC 15939 *In 12th ICIQ Proceedings*. MIT, Cambridge, MA. Pp. 393-408.

Caro, A., Calero, C. and Piattini, M. (2007) Development process of the operational version of PDQM *In The 8th WISE Proceedings*. Springer-Berlag, Nancy, France.

Daconta, M., Obsrt, L. and Smith, K. (2003) *The Semantic Web: A guide to the future of XML. Web Services and Knowledge Managemen*t, Willey Inc, Indianápolis, Indiana.

Ding, L., Finin, T., Joshi, A., Peng, Y., Pan, R. and Reddivari, P. (2005) Search on the Semantic Web. *IEEE Computer*, 38, 62-69.

English, L. (1999) *Improving Data Warehouse and Business Information Quality: Methods for reducing costs and increasing Profits*, Willey & Sons, New York, NY, USA.

Gendron, M. and D'Onofrio, M. J. (2002) Formulation of a Decision Support Model Using Quality Attributes In *7th ICIQ. MIT*, Cambridge, MA, USA, pp. 305-316.

Guha, R., McCool, R. and Miller, E. (2003) Semantic search. *In Proceedings of the 12th WWW*. ACM Press, Budapest, Hungary, pp. 700-709.

Herrera-Viedma, E., Pasi, G. and Lopez-Herrera, A. (2006) Evaluating the Information Quality of Web Sites: A Quality Methodology Based on Fuzzy Computing with Words. *JASIST*, 54, 538-549.

ISO-25012 (2007) ISO/IEC 25012: Software Engineering - Software Quality Requirements and Evaluation (SQuaRE) - Data Quality Model (Draft). Vol. 2007 ISO/IEC.

ISO/IEC (2000) ISO/IEC 15939. *Information Technology - Software Measurement Process*.

Lee, Y. W., Pipino, L. L., Funk, J. D. and Wang, R. Y. (2006) *Journey to Data Quality*, Massachussets Institute of Technology, Cambridge, MA, USA.

Loshin, D. (2001) Data Quality and Business Rules. In Information and Database Quality (Eds, Piattini, M., Calero, C. and Genero, M.) Kluwer Academic Publishers.

Naumann, F. and Rolker, C. (2000) Assessment Methods for Information Quality Criteria *In Fifth ICIQ Proceedings*. MIT, Cambridge, MA, USA, pp. 148-162.

Strong, D. M., Lee, Y. W. and Wang, R. Y. (1997) Data Quality in Context. *Commun. ACM*,, 40, 103-110.

Wang, R. Y. (1998) A Product Perspective on Total Data Quality Management. *Commun. ACM*, 41, 58-65.

Wang, R. Y., Reddy, M. and Kon, H. (1995) Towards quality data: An attribute-based approach. *Decis. Supp. Syst.*, 13, 349-372.