FEDERATING INFORMATION SYSTEM QUALITY FRAMEWORKS USING A COMMON ONTOLOGY

Samira Si-saïd Cherfi
CEDRIC-CNAM, 292 Rue Saint Martin, F-75141 Paris Cedex 03
samira.cherfi@cnam.fr

Jacky Akoka
CEDRIC-CNAM and Institut Telecom/TEM Research
akoka@cnam.fr

Isabelle Comyn-Wattiau
CEDRIC-CNAM and ESSEC Business School
wattiau@cnam.fr

Abstract: Information system (IS) quality can be characterized as a multidimensional system. It encompasses software quality as well as data quality. It also comprises model quality, service quality, process quality, and more generally IS quality. Modeling several aspects of IS quality leads to specific ontologies. To the best of our knowledge, there is no global ontology dedicated to all the dimensions of an IS. A single ontology federating all the aspects of quality is not available. The aim of this paper is to propose and discuss the main constituents of an ontology of quality federating all the aspects of IS components quality (software, data, models, etc.). In order to operationalize the proposed ontology, we describe an approach allowing us to use the ontology in order to achieve specific quality goals.

Key Words: Quality, ontology, quality ontology, quality attribute, quality metrics, quality improvement, quality measurement.

1. INTRODUCTION

Huge investments have been made by companies in information systems (IS) in order to improve IS quality. For many companies, IS quality is also a matter of image and a mean to achieve or maintain a good position in markets competitions. Quality approaches may be generic, such as ISO9000 implementations, or specific, e.g. data quality based on specific software. Stylianou defined total IS quality as a multidimensional concept including infrastructure, software, data, information, administration, and service quality [28]. Main researches on IS quality are concentrated on one of these aspects, historically software quality and information quality. [1] describe a step forward to a common vision.

In this paper, we propose to go further in this direction by capitalizing knowledge on IS quality in a common ontology. We argue that the ontology is a rich concept enabling this capitalization based on the advantages listed in [20]:

– It permits knowledge reuse.

- It allows users to understand rather easily the knowledge of a domain they want to discover.
- It separates fundamental knowledge and implementation considerations.
- It offers a mean to analyze knowledge structure.
- It helps in finding a consensus on domain knowledge.
- It facilitates knowledge sharing and dissemination.
- It enables the dynamic use of this knowledge by programs and tools.

The structure of this paper is as follows. Section 2 provides the reader with a state of the art on IS quality ontologies. The third section describes the main components of our quality ontology, named QualOnto. Section 4 proposes a guidance approach based on this ontology and sketches several potential applications. Finally, the last section concludes and describes future research directions.

2. RELATED WORKS

The domain of information system quality was subject to numerous modeling efforts. The first approaches took place in the field of software engineering, leading to hierarchical definitions of quality factors composed of characteristics, which could lead to evaluations based on metrics [18, 4,5]. These quality models were largely adopted by software engineers and later recognized as standards, such as ISO9126 [8] or, later, SQuaRE [10].

Researchers defined service quality on a larger perimeter, for example by the SERVQUAL conceptual framework, where customer satisfaction was defined and measured [22]. SERVQUAL was mainly based on five aspects: reliability, assurance, tangibles, empathy, and responsiveness. It was largely used and referenced in many fields, such as marketing and tourism management [21]. [7] proposed an interesting framework for database quality composed of four main dimensions: process, data, model, and behavior. It is one of the first attempts to aggregate intrinsic and perceived quality. Model quality is a more recent research topic. It reflects the growing maturity of engineering methods. Model quality is of crucial importance in model driven approaches (MDA). [19] presents an interesting state of the art on model quality, leading to the following quality goals: correctness, completeness, consistency, comprehensibility, confinement, and changeability. These quality goals may be reached thanks to best practices described in modeling methodologies and formal approaches. Literature mainly concentrates efforts on UML models and describes experiments conducted with classes of IS or computer science students. The underlying principle is that a model aims to a given objective, on which quality goals may be defined and best practices may be associated.

Thanks to their underlying ontology theories, new quality models appear. The first paper dedicated to this topic is [13]. It proposes a quality ontology composed of three domains: measure, identification, and traceability. Quality is defined as conformity to requirements. Measure allows the evaluation of the gap between the product or the service and the requirements. Identification focuses on the definition of the quality problem. Finally, traceability permits to record all information required to understand quality problems. The TOVE-Quality Ontology is populated thanks to a method based on a business scenario. Domain hypotheses are added. The definition of the quality problem consists of a question deduced from the scenario. It is a basis for the definition of competency questions. Users provide these questions in order to extract knowledge from the ontology. TOVE-Quality Ontology addresses both product and service quality. It can be used, for instance, to build quality control applications. The latter allow users to identify and track product or service defects. The measure ontology can also be used in different contexts, for example as a foundation on which quality management web services may be defined [14].

Shekhovtsov published a very detailed comparative study on quality conceptualization techniques, either models or meta-models, or ontologies [27]. The author proposes fourteen dimensions on which the different approaches may be compared and characterized. The large time span addressed enables also a description of the evolution of quality conceptualization during several decades. The dimensions include abstraction level, structural complexity, ability to measure, modeling of dependencies between quality attributes, etc. The scope of the study is limited to software quality. Quality concept in [27] is close to the definition proposed by the CORE ontology (Core Ontology for Requirements Engineering) [12]: quality embraces a set of particular perceivable and measurable entities that characterize individuals, e.g. the components of software. These qualities may be linked together either hierarchically or by means of interdependencies. Quality is associated to a conceptual space in which an individual may be located thanks to quality measures. Thus a map of the approaches may be obtained.

Based on our literature review, we argue that current or past quality ontologies are not centered on IS quality. Moreover, to the best of our knowledge, there is no quality ontology allowing IS experts to federate different IS quality approaches. Finally, existing ontologies are not necessarily consistent with well-known standards, such as ISO. In the following, we propose a step toward a quality ontology dedicated to IS encompassing its different components (software, models, data, business processes, etc.). It is based on existing standards aiming at the unification of various sets of knowledge.

3. INTRODUCTION TO QUALONTO

Quality is a hot research topic. It is of great interest to several Information Systems engineering communities. However, there exists neither integrated, nor agreed view of the concepts and methods for quality analysis, evaluation and improvement and even no common vocabulary. The existing literature describes plenty of proposals resulting in a variety of terms, definitions, vocabulary etc. with different objectives, different degrees of formality, maturity and rigor. For example, in software engineering, quality concepts are well defined and structured and ISO proposed several standards, such as such ISO 9126 and ISO 25030, etc. In the domain of model quality, even if there are several rich contributions, the lack of maturity and validation effort did not yet permit the emergence of standards. We believe that, in order to achieve a certain level of maturity, it is necessary to share knowledge in a common representation. It is an intermediate step toward standardization. Moreover, there is a need to bridge the gap between the different engineering areas or domains and to exhibit similarities but also differences in quality problems and solutions.

As a step forward towards this objective, we propose to construct a quality ontology addressing quality problems and solutions through several domains. In computer science, the concept of ontology emerged during mid-80s as a mean to structure and use knowledge. According to [29], an ontology has the following requirements:

- It enables identification of key concepts and their relationships within the suitable context of use (scope).
- It provides textual, precise and non-ambiguous definitions for these concepts and relationships.
- It provides precise terms and vocabulary to characterize these concepts.
- It represents an agreement on all the precedent subjects.

According to our knowledge in the areas of software quality, data quality and conceptual model quality, we believe that IS quality expertise should benefit from a general quality ontology. We propose to define QualOnto, in order to capitalize and share knowledge about all IS quality practices.

In this paper, we define the main concepts of a general quality ontology. The latter has two main objectives. The first one aims to capitalize and unify structuring and abstraction effort conducted by several research communities in the definition of quality concepts. The second objective is to help reinvestment of this effort for new quality definition areas for which the current contributions are rudimentary or insufficient.

3.1. The main concepts of QualOnto

A quality ontology should be able to provide answers related to the characterization and the measurement of quality. It should also be able to guide the achievement of a given quality goal by the means of suitable and precise quality concepts and measures. The difficulty lies in various factors that we quote below:

– the variety of engineering areas (software engineering, database, information system modeling, etc.) and the related quality proposals that have been developed independently of other areas,

– the variety of quality frameworks within the same area,

– the specific quality requirements for each project increasing the need for assistance in the implementation of quality approaches.

Although [6] suggests that it is undesirable to build such different viewpoints in a common ontology, we argue that this can be done by separating knowledge into three semantic levels of the same ontology, thus coping with the complexity derived from these factors. These three levels can be defined as follows:

- Domain Independent Quality Ontology – DIQO,
- Domain Specific Quality Ontology– DSQO,
- Operational Quality Ontology– OQO.

We describe below the main concepts of the meta-model underlying QualOnto and gathering the three semantic levels.

3.1.1. The DIQO level

The DIQO level is the more generic one. It covers all engineering areas. Its purpose is to define a reduced but sufficiently generic set of concepts and relationships between concepts, enabling the description of all quality concepts. To do so, we have studied the concepts of quality proposed in various fields. For example, in software engineering, we argue that the ISO 9126 and its evolution SquaRE are a good starting point [10]. In the field of data quality, we have taken into account the standards ISO 19113 [9] but also several data quality research frameworks [23; 31; 30]. We also studied some specific work dedicated to application domains such as medical [2], banking [26] or geographic area [11], etc. This simple model allows the definition of quality at a very high level. Quality concepts defined according to this structure can further be extracted according to their names, their key words or according to the nature of the link, which maps them with other concepts. In the various engineering areas of our scope, we have observed that this high level quality view is a useful and a relevant step before obtaining more detailed and more operational concepts.

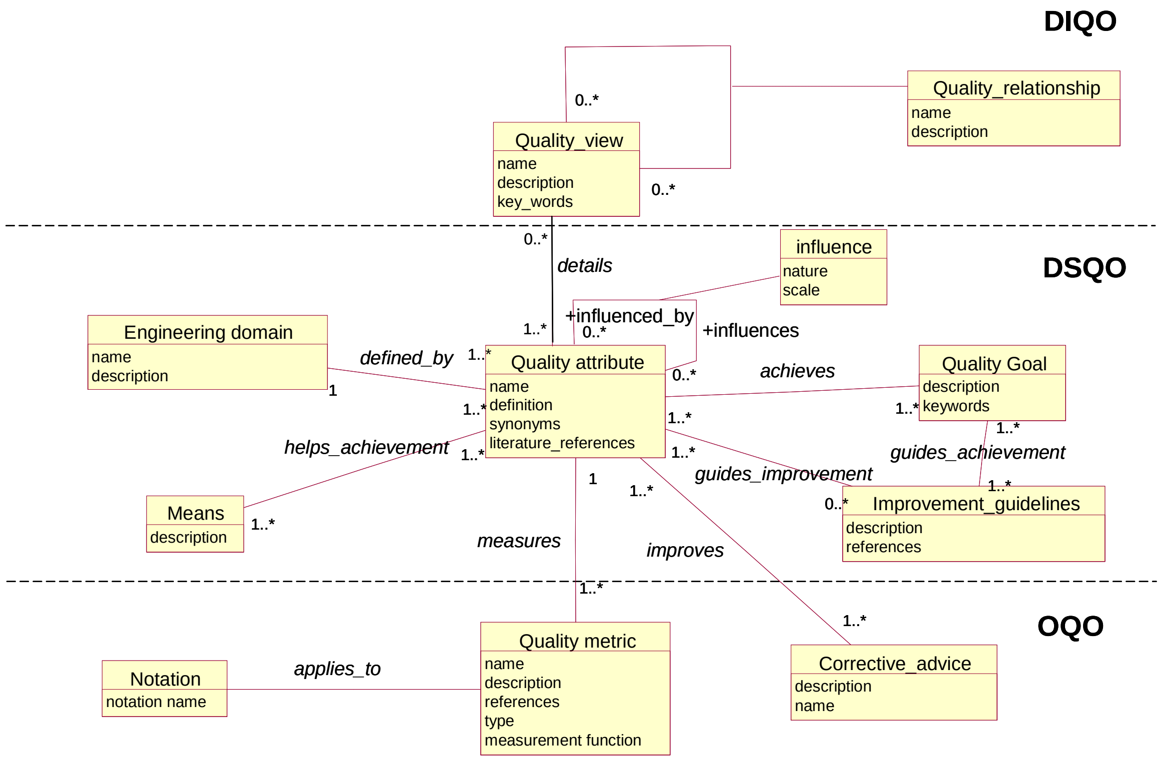

Fig. 1. The ontology meta-model

In software engineering for example, the McCall model [18] defines quality according to three perspectives: product revision (ability to undergo changes), product transition (adaptability to new environments), and product operations (its operation characteristics). According to ISO/IEC 9126, software quality is measured according to its *internal* characteristics (based on structural aspects), *external* characteristics (measuring conformity to requirements) and quality *in use* characteristics (measuring the user's satisfaction during the fulfillment of their goals).
From a conceptual modeling point of view, the Lindland's quality framework [17] and the Krogstie's extension of this framework [16] characterize models according to their *syntactic* quality (conformance to the used notation), *semantic* quality (conformance to the domain knowledge) and *pragmatic* quality (conformance of the model interpretation by actors).

From the requirements engineering area, Pohl [24] has defined three dimensions that are: *specification* (measuring the completeness according to the domain knowledge), *representation* (measuring the degree of formality of the specifications) and *agreement* (measuring the agreement on the produced specifications). Krogstie has added this notion of agreement as *social quality* in the extension of the Lindland's framework.

The relationship concept *Quality_relationship* captures semantics-based relationships among quality concepts. These relationships are from two *types:* explicit or inferred. Their semantics is described by the *name* and *description* attributes. Synonymy, homonymy, refinement and generalization are examples of *Quality_relationship* concept.

3.1.2. The DSQO level

We represent at this level the quality models dedicated to specific engineering areas (software engineering, data quality, etc.). It enables the refinement of the DIQO level by adding quality concepts taken from specific engineering domains.

The ontology at this level relies on the four following concepts:
1) The quality goal defines the fulfilled quality objective. This goal is proposed by a stakeholder. For example, given a conceptual model, the analyst could be interested in improving the clarity and the semantic completeness of the specification. However, a designer in charge of detailed specifications could be more interested in improving the syntactic correctness and completeness. He/she could be also be focused on the formality of the models to be able to build easily the implementation of the conceptual artefacts and minimize the errors.

2) The quality means was introduced by [17]. The authors distinguish two sub-concepts, namely the *property* of the product and its *activity*. For example, to achieve the syntactic correctness, one should verify the formality of the notation (property) using error detection techniques (activity).

3) The quality attribute represents desired properties of the product. These quality attributes, also referred to as quality factors, properties or characteristics, bridge the gap between the quality goal and the way to measure it. The main objective of quality evaluation is to find the best compromise between the different quality values according to a given set of quality attributes, maximizing the global quality. Completeness and integrity are examples of quality attributes. Quality attributes are not independent. The influence relationship materializes possible dependencies between attributes. As an example, conceptual model simplicity is generally opposed to model understandability.

4) The improvement guidelines represent a set of advices and recommendations originating from best practices and/or development methods that aim to improve the quality of delivered products. The originality of our proposal is to relate these good practices and advices to quality measurement results. Most of the previous quality approaches focus mainly on quality measurement. Literature contains many modeling methods and recommendations. It lacks links between poor quality and guidance methods to improve it. This concept of quality guideline is then refined at the operational level into corrective actions.

In order to illustrate our ontology, let us consider internal quality as a *quality view* from the DIQO level. It is refined, at the DSQO level, in conceptual modeling (*engineering domain*), by syntactic correctness (*quality attribute*) to a*chieve* the model correctness improvement (*quality goal*). To measure this correctness, we could apply error detection techniques (*means*). The improvement of correctness could use refactoring guidelines (*improvement guidelines*).

3.1.3. The OQO level

This level defines required concepts needed to operationalize quality measurement and improvement. It introduces the concept of quality metrics which is the mean used to evaluate quality and compute values for a given IS project according to targeted quality attributes. Several metrics can be associated to the same quality attribute. Metrics can be automatable. In this case, they have an associated function, based on the characteristics of the object being measured, which could be implemented using an algorithm. However, some metrics rely on the judgment of human agents and are therefore non-automatable.
For example, to measure the clarity of a model, an automatable metrics could be the ratio between the number of crossing segments and the total number of segments in a model. We could also have a non automatable metrics consisting in assigning, by a human agent, a value between 0 (very bad clarity) and

5 (very good clarity).

Metric definitions are based on notations used to describe the object being measured. As an example, for a programming activity, the *notation* can be a programming language. For a modeling activity, the notation is the modeling notation used. The notation could also be a structured natural language for requirements specification. Some metrics are consequently notation-specific. If we consider the automatable clarity metrics presented before, this metrics can only be applied to a graphical notation.

The *corrective actions* are precise guides that can be structured either as a sequence of transformation actions or as a textual description of the detailed improvements to be applied. Some advices may also be specific to the notation used. An example of corrective action is the merging of entities participating to a 1-to-1 relationship in order to increase the simplicity of an entity-relationship model.

3.2. Ontology construction

This section is dedicated to the process of QualOnto construction. For each of the three definition levels, Figure 2 presents the sources of knowledge explored, the actors involved, and the resulting concepts of the process at each level. We have prototyped the ontology using the editor Protégé [25].

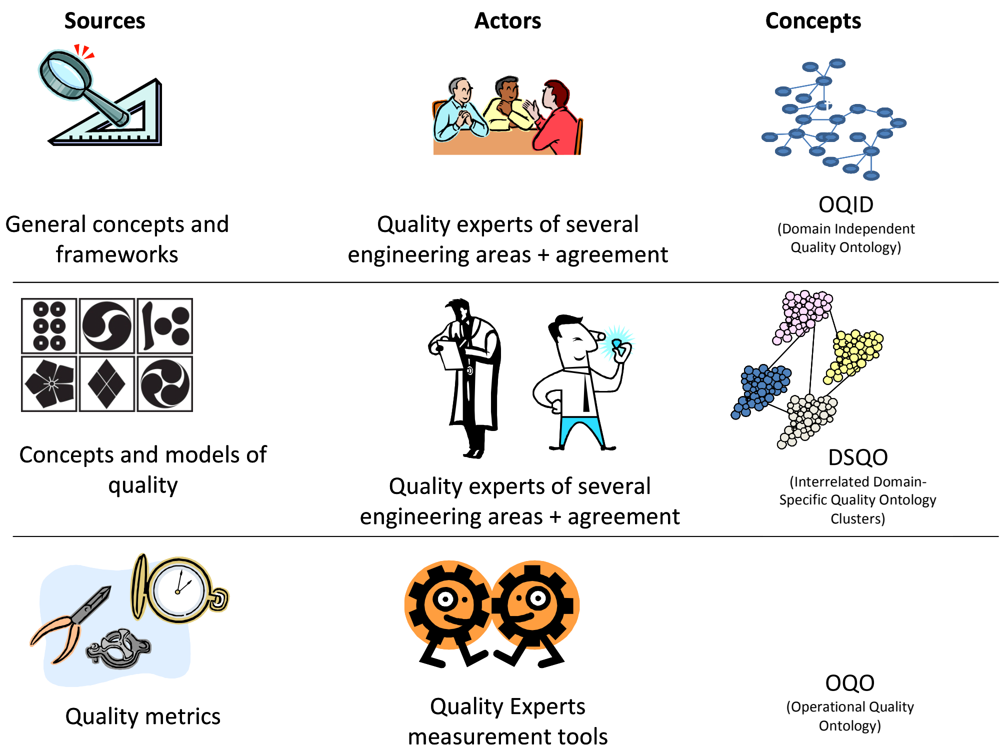

Fig. 2. The three definition levels of QualOnto

3.2.1. QualOnto – DIQO Construction

The quality concepts from DIQO result from a fine-grain analysis of existing quality frameworks and standards from various engineering domains. According to the ISO/IEC 9126 standard, quality characteristics are categorized as external quality, internal quality, and quality in use. The syntactic and semantic quality aspects defined within the framework of Lindland logically are related to internal and external quality characteristics of a software. The pragmatic quality, measuring the correspondence between the model and people's interpretation can be defined within either the external quality or the quality in use.

Consequently, we suggest to adopt the characterization of quality as internal, external and/or in use, at the DIQO level. Furthermore, the fact that we borrowed this vocabulary from a well-recognized standard is a step towards an agreement, which is an important requirement for ontology construction. The description presented in Figure 3 is the implementation of this level using Protégé.

```
Class: Qualonto:Quality_concept
    SubClassOf:
        owl:Thing,
        Qualonto:QV_key_words min 1 xsd:string,
        Qualonto:QV_description exactly 1 xsd:string
Individual: Qualonto:InternalQuality
    Types:         Qualonto:Quality_concept
    Facts:           Qualonto:QV_description  "Quality of
the  product  based  on  its  intrinsinc  characteristics
(structure)"
Individual: Qualonto:QualityInUse
    Types:
        Qualonto:Quality_concept
        Facts:
        Qualonto:QV_description   "Quality of the product
as perceived by its end users in the achievement of their
goals"
Individual: Qualonto:ExternalQuality
    Types:         Qualonto:Quality_concept
    Facts:    Qualonto:QV_description    "quality  of  the
final product as assessed by its external behavior"
```

Fig. 3. QualOnto-DIQO description

3.2.2. QualOnto – DSQO Construction

This level of ontology is supposed to cover all areas of engineering, addressing the problem of measuring and improving quality. At this level, the concepts are derived from the literature but are validated and enriched by quality experts from various fields. This level is subject to a continuous enrichment, especially for areas where quality is highly subjective and poorly structured. Figure 4 presents an excerpt of that level. The arcs have been labelled to help readability.

It illustrates the fact that within the SoftEng *Engineering domain* we have identified an InternalQuality *Quality view detailed by* Analysability *Quality Attribute* that *achieves* the Maintainability *Quality goal*

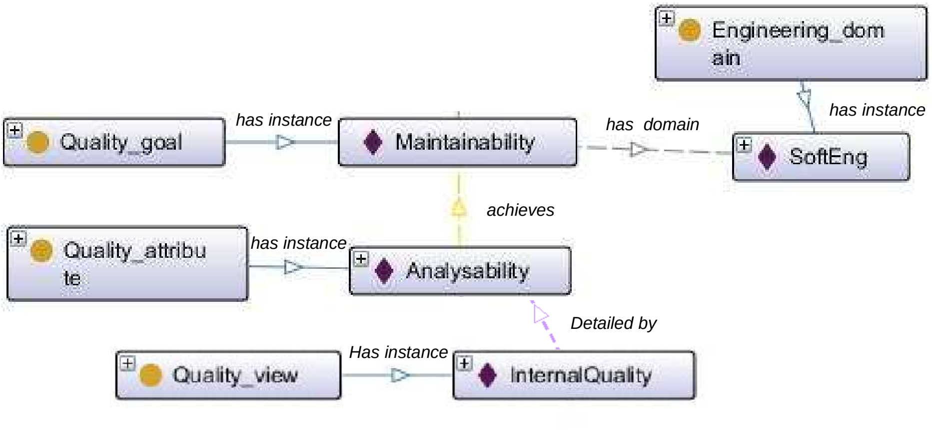

Fig. 4. Extract from DSQO level

The result can be seen as a set of interrelated domain-specific ontology clusters. Consequently, one usage of this ontology is to infer inter-domain semantic relationships, leading to inter-domain quality concepts elicitation, and enrichment of poor-structured quality domains. Indeed, many efforts have been done in several engineering domain for quality definition and an interesting issue is the one of capitalizing inter-domain knowledge about quality not to unify proposals but for a mutual enrichment and learning from other's experiences. This is the essential objective of this level as it is more detailed than DIQO and less specific than the next level OQO.

3.2.3. Qualonto – OQO

OQO level supports the operationalization of the evaluation and the improvement of quality, based respectively on quality metrics and corrective actions. Once the quality attribute for achieving the quality goal is identified, it can be measured using metrics. According to the measurement results, possible improvements can be obtained by applying corrective actions.

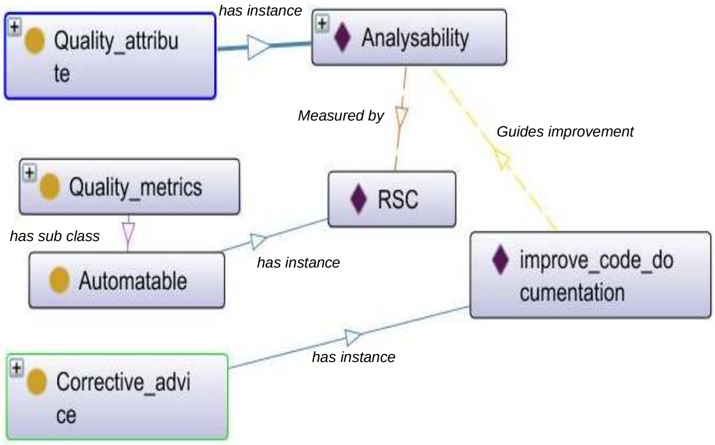

Fig. 5. Extract from QualOnto-OQO

For example, the ontology excerpt presented in Figure 5 describes RSC (Readability of Source Code), which is an example of *automatable metrics*, for the measurement of analyzability *quality attribute*. This analyzability can be improved through the *corrective advice* "improve code documentation".

4. USING QUALONTO

QualOnto is mainly dedicated to federating other quality ontologies and frameworks. In this section, we sketch several types of usage of QualOnto. Let us remind that a quality ontology includes knowledge representation and a consensual semantics [3]. It is composed of a set of concepts (quality attributes, quality metrics, etc.), and of a set of relations between these concepts (ISA, instance of, etc.). One of the objectives of Qualonto is knowledge sharing and reuse. As pointed by [29], reuse of existing ontologies requires considerable effort. Our objective is not to integrate existing ontologies but to discover, thanks to QualOnto, semantically close knowledge contained in different ontologies and to infer on this semantic closeness. However, the process to fullfill is merely as follows:

1. to find the parts of ontologies where they overlap;
2. to link concepts that are semantically close using equivalence and subsumption relations;
3. to check the consistency and non-redundancy of the result.

We are interested in discovering such relationships between ontologies related to different engineering domains. In order to achieve this objective, we define semantic similarities as a mean to detect semantic closeness.

4.1. A Proposal for Semantic Similarity Measurement

This process relies on categorization of mismatches between the ontologies. In the literature, a difference is made between i) language level mismatches occurring when ontologies are described using different languages, and ii) ontology level mismatches [15]. Since we have unified the vocabulary thanks to QualOnto model, the first mismatch type is not to be considered. As for the second type of mismatch, we have identified a set of rules enabling semantic closeness discovery, based on semantic similarity measures. We have defined two semantics for concept closeness, namely *Concept explanation closeness* and *Concept conceptualization closeness*. For each of them we have associated semantic distances.

4.1.1. Concept explanation closeness

Concept explanation closeness applies when two quality concepts seem to represent the same concept or are very close but are not described in the ontologies with the same terms. Five distance metrics have been defined:

- Synonymy closeness metrics: Two terms refer to the same concept if they are synonyms. This metrics relies on the fact that, during the implementation of the ontology within the Protégé environment, we have used the connection with Wordnet to facilitate the definition of synonyms. The synonymy closeness distance is defined as the number of common synonyms for two instances of the same QualOnto concepts.

- Homonymy closeness metrics: The homonymy closeness metrics applies when two terms of two ontologies that are instances of the same QualOnto concepts have the same names. The distance equals 1 if there is at least a common synonym and 0 otherwise.

- Paradigm closeness metrics: This metrics helps in detecting terms referring to similar concepts but that have not been associated to the same QualOnto concepts. For example, let us consider a quality_metrics and a quality_attribute. This metrics applies for terms that are equal (the same name) and that are instances of different QualOnto concepts. It sums the number of common synonyms and the number of common key words.

Other metrics have been defined to detect redundancies and conflicts within the same ontology (related to same engineering domain) that are *Description closeness* and *Encoding closeness* metrics.

- The first one detects different instances of the same QualOnto concepts that have merely the same descriptions (synonyms, keywords,etc).
- The second helps detecting instances of different QualOnto concepts having merely the same descriptions (synonyms, key words,etc).

4.1.2. Concept conceptualization closeness

Concept conceptualization closeness applies when two concepts from two different ontologies have the same names. We have defined one distance metrics, namely coverage closeness metrics. The latter helps detecting when a term from an ontology corresponds to a set of terms from another ontology. The metrics detects the existence of a one-to-many relationship between terms of different ontologies based on the detection of common synonyms and key words.

In the next section, we illustrate these metrics.

4.2. Examples of usage contexts

The current implementation of QualOnto gathers quality concepts from domains such as Conceptual Modeling, Software Engineering, Data Management and Ontology engineering. In this section, we describe in detail three contexts of usage of QualOnto. The latter acts as a huge knowledge-base for quality definition and measurement by browsing the content of the three layers of QualOnto. Moreover, Qualonto can also be used to help defining, enriching, and validating its content. Some of these facilities are detailed in this section.

4.2.1. Ontology Definition Enrichment

Ontology construction is a hard task, which is time consuming. However, an important part of the work is performed thanks to inference rules. As an example, one inference rule is as follows:

If a Quality Attribute *QA1* achieves a Quality Goal *QG1*
And *QA1* is close to a Quality Attribute *QA2* according to the synonymy closeness metrics
Then *QA2* achieves *QG1*

Such an inference rule enables to enrich the semantic relationships within the ontology. For example, we first defined a *achieves* relationship between Comprehensibility *quality attribute* and Maintainability *quality goal* from Conceptual modeling *engineering domain*. Moreover, the synonymy closeness metrics detected nearness between Comprehensibility and Readability *quality attributes.* The inference rule infers the fact that Readability *achieves* Maintainability quality goal.

4.2.2. Ontology Definition Completion

QualOnto aims to help in completing the ontology definition by reusing knowledge from other domains. Indeed, as mentioned in previous sections, quality evaluation and improvement research contributions are heterogeneous. Some engineering domains, such as software engineering, are at an advanced stage reaching standardization, whereas others are rather recent.

By applying the coverage closeness metrics, we identified that accuracy *quality attribute* from Data e*ngineering domain* is covered by syntactic correctness and semantic correctness *quality attribute*s from conceptual modeling *engineering domain*. Moreover, the definitions of syntactic and semantic correctness are detailed and contain precise metrics expressions.

Thus, thanks to QualOnto, we are able to complete the definition of accuracy *quality attribute* for data quality by introducing concepts of semantic accuracy. The latter may be defined as the correspondence between data and real values, whereas semantic correctness is defined as the correspondence between data and user requirements. Similarly, we define syntactic accuracy as correctness according to the coding rules in the schema, by analogy with syntactic correctness defined as the degree with which a model respects the notation rules.

4.2.3. Ontology definition validation

Finally, QualOnto aims to help validating the ontology content by detecting redundancies or overlapping in the definition of concepts. As an example, if two quality attributes are defined within the same engineering domain and share the same set of synonyms, then we suspect a redundancy in the definition of the same quality attribute.

As illustrated above, many usages can be implemented, based on QualOnto. The federation of different knowledge sources enables a rich cross-fertilization between different ontologies and frameworks.

5. CONCLUSION AND FUTURE RESEARCH

Information system quality is a crucial topic for companies as well as for researchers. There is a large set of quality frameworks. However, the lack of a consensus and standards allowing managers to evaluate globally the quality of their IS is a strong limitation for the quality experts. In this paper, we proposed an approach enabling the federation of existing ontologies. QualOnto is characterized by three main strengths: 1) it puts together in a structured manner different quality frameworks, as a step forward towards a federated view of quality. 2) It allows a rich sharing of quality knowledge between different levels of expertise and between different fields of qualities. The three abstraction levels of QualOnto lead to different usage scenarios (from documentation to operationalization). 3) Finally, the definition of a guidance process, based on this ontology, is proposed to go further in the usage of the quality concepts in the domains where quality is not yet mature enough.

The next step of our research effort lies in the formalization of the rules and the metrics that are currently defined in Protégé using SQWRL queries. We are also, in parallel, working on a querying facility using several criteria (quality concepts names, keys words, engineering domain, authors etc.).

Future research will deal with more validation effort, namely the ontology structure and its content. The usage of ontologies helps knowledge sharing and we plan to publish the ontology to collect feedbacks.

Finally, we aim at using QualOnto as a framework linking together the IS engineering process and the IS product, which could serve as a basis for statistical studies on the correlation and causality between both process and product qualities. Indeed, during previous work on both data quality and model quality we observed correlations and mutual influences between data quality dimensions and conceptual models quality criteria. QualOnto federates the several proposals within the same ontology and we hope this will help working on these inter-domain influences.

6. References

[1] J. Akoka, L. Berti-Équille, O. Boucelma, M. Bouzeghoub, I. Comyn-Wattiau, M. Cosquer, V. Goasdoué, Z. Kedad, S. Nugier, V. Peralta, et S. Sisaïd-Cherfi, A Framework for Quality Evaluation in Data Integration Systems. In Proc. of the 9th Intl. Conf. on Enterprise Information Systems (ICEIS 2007), Madeira, Portugal.

[2] D.G.T. Arts, N. F. de Keizer, G.J. Scheffer, Defining and Improving Data Quality in Medical Registries: A Literature Review, Case Study, and Generic Framework, In JAMIA 2002.

[3] J. Bouaud, B. Bachimont, J. Charlet, P. Zweigenbaum, Methodological principles for structuring an ontology. In ACM Press, editor, Proceedings of IJCAI'95 Workshop: Basic Ontological Issues in Knowledge Sharing, 1995.

[4] Boehm, B.; Brown, J.R.; Kaspar, H.; MacLeod, G.J.; Merritt, M.J. (1978), Characteristics of Software Quality, North-Holland.

[5] A. Davis, S. Overmyer, K. Jordan, J. Caruso, F. Dandashi, A. Dinh, G. Kincaid, G. Ledeboer, P. Reynolds, P. Sitaram, A. Ta, M. Theofanos: Identifying and measuring quality in a software requirements specification, In Proceedings First International In Software Metrics Symposium (1993), pp. 141-152.

[6] Gonçalves B., Guizzardi G., Pereira Filho J.G., Using an ECG Reference Ontology for Semantic Interoperability of ECG Data, Journal of Biomedical Informatics, 44(2011), pp 126-136.

[7] Hoxmeier J.A., "Typology of database quality factors", Software Quality Journal 7, 1998, pp 179-193.

[8] ISO, International Organization for Standardization: ISO9126-1:2001, Software Engineering – Product Quality, Part 1: Quality model (2001).

[9] ISO 19113:2002, Information géographique – Principes qualité.

[10] ISO/IEC 25000:2005, Ingénierie du logiciel -- Exigences de qualité du produit logiciel et évaluation (SQuaRE) -- Guide de SQuaRE.

[11] Jansone A., Borzovs J. An approach to geographical data quality evaluation, In 7th International Baltic on Databases and Information Systems 2006.

[12] I. Jureta, J. Mylopoulos, S. Faulkner. A core ontology for requirements. Applied Ontology 4, 169-244; 2009.

[13] Kim H.M., Fox M.S., Grüninger M., An ontology for quality management – enabling quality problem identification and tracing, BT Technol J, Vol 17 N°4, 1999.

[14] Kim H.M., Fox M.S, Towards a Data Model for Quality Management Web Services: An Ontology of Measurement for Enterprise Modeling, CAISE 2002, LNCS 2348, pp. 230-244, 2002.

[15] M. Klein. Combining and relating ontologies: an analysis of problems and solutions. In IJCAI-2001 Workshop on Ontologies and Information Sharing, pages 53--62, Seattle, WA, 2001.

[16], Krogstie J., Lindland O.I., Sindre G., "Defining quality aspects for conceptual Models", In Proceedings of the IFIP8.1. Working Conference on Information Systems Concepts (ISCO3),

Chapman & Hall, London, 1995.

[17] Lindland O.I., Sindre G., Solvberg A., “Understanding Quality in Conceptual Modeling”, IEEE Software, Vol. 11(2), 1994, pp 42-49.

[18] McCall, J. A., Richards, P. K., and Walters, G. F., "Factors in Software Quality", Nat'l Tech. Information Service, no. Vol. 1, 2 and 3, 1977.

[19] Mohagheghi P., Dehlen V., Neple T., “Definitions and approaches to model quality in model-based software development – A review of literature”, Information and Software Technology, 2009, Vol 51, pp 1646-1669.

[20] Noy, F. N. McGuinness, D.L. 2001. Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report SMI-2001-0880, Stanford Medical Informatics,Stanford University, Stanford, CA, USA.

[21] Obenour W., Patterson M., Pedersen P., Pearson L., Conceptualization of a meaning-based research approach for tourism service experiences, In: Tourism Management, 27 (2006), pp. 34-41

[22] Parasuraman A., Zeithaml V., Berry L., “A conceptual model of service quality and its implications for future research,” Journal of Marketing, 1985, Vol 49, pp 41-50.

[23] Pipino L., Lee Y.W., Wang R.Y. “Data Quality Assessment.” Communications of the ACM, 45(4). 2002, pp. 211-218.

[24] Pohl K., The three dimensions of requirements engineering, CAiSE 1993.

[25] http://protege.stanford.edu/

[26] Luís Francisco Ramos Lima, Antonio Carlos Gastaud Maçada, Xenophon Koufteros: A Model for Information Quality in the Banking Industry - The Case of the Public Banks In Brazil. ICIQ 2007: 549-562

[27] V.A. Shekhovtsov, On the Evolution of QualityConceptualization Techniques. In: The Evolution of Conceptual Modeling, LNCS 6520, pp. 117-136, 2011.

[28] Stylianou A.C., Kumar R.L., “An integrative framework for IS quality management”, Communications of the ACM, Vol. 43(9), pp. 99-104, 2000.

[29] Uschold M., King M. “Towards a Methodology for Building Ontologies, Workshop on Basic Ontological Issues in Knowledge Sharing, in conjunction with IJCAI-95, Montreal, 1995.

[30] Wang, R., Kon, H. & Madnick, S. Data Quality Requirements Analysis and Modelling, Ninth International Conference of Data Engineering, Vienna, Austria, 1993.

[31] Wang, Y. A Product Perspective on Total Data Quality Management. Communications of the ACM. 41, 2, 1998, pp. 58-65.