A SOA-BASED DATA QUALITY ASSESSMENT FRAMEWORK IN A MEDICAL SCIENCE CENTER

(Practice-Oriented)

Yao Zhou
Charité Universitätsmedizin Berlin, Germany
yao.zhou@charite.de

Sabine Hanß
Charité Universitätsmedizin Berlin, Germany
sabine.hanss@charite.de

Malte Cornils
Charité Universitätsmedizin Berlin, Germany
malte.cornils@charite.de

Sonja Niepage
Charité Universitätsmedizin Berlin, Germany
sonja.niepage@charite.de

Claudia Hahn
Charité Universitätsmedizin Berlin, Germany
claudia.hahn@charite.de

Thomas Schrader
Fachbereich Informatik und Medien,
Fachhochschule Brandenburg, Germany
thomas.schrader@computer.org

Abstract: The Open European Nephrology Science Center is a metadata repository of clinical and research related data on nephrology diseases. Clinical data is source data for medical research. Medical research has specific and changing quality requirements for source data, so the quality of clinical data has to be assessed before it can be used for medical research. Quality assessment requirements change from one medical research to another, hence, a flexible data quality assessment solution is needed. In this paper, we present a data quality assessment framework which supports flexible data quality assessment by allowing users to define their own data quality assessment requirements, and automatically searching for proper data quality assessment tools which fulfill the requirements. Development of the framework is based on an ontology and SOA. Components of this framework are the *Formal Quality Requirement,* the *Service Repository* and the *Service Selection Process*. Those components are discussed in detail and an example is given to show how the framework works.

Key Words: data quality assessment, medical science center, patient record, ontology, SOA

INTRODUCTION

In the field of medicine, a huge amount of data is generated from clinics every day. This data is very important for medical research. However, due to the factors such as physical distance and data security, sharing this valuable medical data with others is difficult and challenging. Nowadays, quite a few projects [18, 23] have been initiated to build systems for sharing medical data across clinics and related medical organizations, in order to support medical research. Open European Nephrology Science Center (OpEN.SC) is one of them. OpEN.SC is a metadata repository of primary clinical and research related data in the domain of nephrology diseases, particularly kidney transplantation [15, 16]. OpEN.SC imports anonymous clinical patient records from partner sources. The amount of data in OpEN.SC increases constantly with scheduled data import updates. Scientists are also provided with tools to carry out research at OpEN.SC.

Data quality (DQ) is one of OpEN.SC's main concerns. As a medical science center, OpEN.SC is aware that different medical research has different DQ requirements for its research source data, due to the specific research approaches or perspectives. Therefore, OpEN.SC is trying to provide specialized DQ assurance functions to medical scientists to support their research work. DQ assurance consists of a series of activities, from DQ assessment to DQ improvement. OpEN.SC's DQ assurance focuses on DQ assessment activities.

BACKGROUND

OpEN.SC's data is mainly clinical data, for example patient records. Data in OpEN.SC is divided into seven domains: basic data (of a patient), examination results, diagnoses, treatment data, external data (e.g. virtual slides, discharge letters), administrative data (e.g. case numbers in a hospital, SAP data) and project data (internal) [17].

Patient records are generated during the process of treatment. The aim of the information gathering from patient histories is to provide an adequate basis for action. For a doctor actively involved in treating a patient, there is only one real problem: what best to do next [2]. Hence, patient records are documented for the purpose of finding and recording a good way of clinical treatment, so they have good quality for this treatment. However, they are not made with medical research requirements in mind.

Nowadays, medical research is highly specialized, such as nephrology-pathology or orthopedics-radiology. Medical research has domain-specific quality requirements for its source data. Although clinical data has good quality for clinical treatment, this does not automatically extend to its usefulness in medical research. Furthermore, in medical research clinical data which is considered to have good quality for one medical research might not be regarded as sufficient for another. For example, there are two teams doing research about broadening the acceptability of both donors and recipients for transplantation. One team is a group of transplant physicians, and the other is a group of pathologists. For the transplant physician group, good source data for them would be patient records including non-surgical patient care monitoring records of donor and recipient pairs in transplantation. The monitoring record should at least include a list of certain examinations and check result values. For the pathologist group, good source data would be patient records from donor and recipient pairs with electronic stained slides of different zoom levels. Therefore, for different medical research, DQ assessment has different requirements.

Research about DQ assessment can be divided into three categories: *problem classification*, *dimensions of data quality*, and *assessment methodology* [8]. *Problem classification* research tries to identify and classify DQ problems [7, 10, 14]. A general classification model, which is able to encompass all known *DQ problem classifications*, is described by Ge and Helfert [8]. One important classification element is the perspective. Problems with data quality can be seen from the user's point of view, in which the semantics of the data are not sufficient for the user's requirements (*user perspective*). The other possible perspective is the *data perspective*, in which technical problems on the data occur, for example syntax violations. The second classification element, independent of the perspective, is the context-sensitivity. Some problems only exist in the specific domain context the data is to be used – they are *context-dependant*. Other problems occur for any possible use of the data, like insecure data. These are *context-independent*. According to this classification model, most of OpEN.SC's DQ problems are defined from the data user's perspective, in our case the medical research scientist, and most are context dependent to the specific medical domains.

DQ dimensions have been researched a lot during the last two decades. There are mainly two *DQ dimension* identification perspectives: database and management. A management-identified dimension would be believability of data, while a database-identified dimension would be input correctness. OpEN.SC DQ research considers the management perspective for identifying *DQ dimensions*.

DQ assessment methodologies are divided into two types by Pipino et al.: objective and subjective DQ assessment [19]. Objective DQ assessment is to measure the extent to which information conforms to quality specifications and references [20, 24]. Subjective DQ assessment is to measure the extent to which information is fit for use by data consumers [11, 13]. Both objective and subjective *DQ assessment methodologies* are needed by OpEN.SC DQ assessment. Each *DQ problem* can be linked to different *DQ dimensions*, and each *DQ dimension* can be linked to different *DQ assessment methodologies* [8]. This gives us the inspiration to automate the linking among *DQ problems, DQ dimension* and *DQ assessment methodologies*.

RATIONALE & PURPOSE

As analyzed above in background, we know that different medical research has different DQ requirements for its source data. As a medical science center where patient records are used as a source for medical research, OpEN.SC is responsible for providing scientists with tools which can assess the quality of data they want to use according to particular research requirements. With any change of research requirements, the requirements for DQ assessment change. How to make DQ assessment functions more flexible in order to meet all medical research DQ assessment demands, but not only a few of them?

Previous approaches emphasize mostly on general DQ assessment problems and methodologies [19, 20]. Users are limited by only using the available DQ assessment tools, even though the tools do not exactly meet their DQ assessment requirements. In contrast, OpEN.SC is trying to take an approach which allows users to define their own quality requirements, and then we search automatically a proper service from our service repository to match their quality requirements.

METHODS

Ontology

The term ontology originates from philosophy, but it has been used in many other sciences. According to Tom Gruber, who is credited with giving the term ontology a technical definition for computer science, "In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes, attributes, and relationships." Therefore, an ontology is usually used to describe a formal agreement of an entity within a particular domain [9]. An ontology provides a semantic representation of an entity and its relationships to other entities.

Ontology languages are formal computer languages to encode an ontology. Ontology languages are usually XML based, as an ontology can be considered a conceptual level on top of XML data [12]. Hence, an entity which is presented with an ontology is manipulable in an information system.

Service Oriented Architecture

A flexible software architecture is necessary in order to meet DQ assessment requirements. Service Oriented Architecture (SOA) is this kind of architecture. SOA, defined from the perspective of architecture style, is a set of patterns and guidelines for creating loosely coupled, business-aligned services that, because of the separation of concerns between description, implementation, and binding, provide unprecedented flexibility in responsiveness to new business threats and opportunities. A service is a software resource with an externalized service description. Service consumers search, bind and invoke a service by the service description. The service provider realizes the service description implementation and delivers the quality of service requirements to the service consumer [1]. Separation of concerns is not new, so what makes service-oriented separation different? Service-oriented architecture encourages individual units of logic to exist autonomously, yet not isolated from each other. Units of logic are still required to conform to a set of principles that allow them to evolve independently, while still maintaining a sufficient amount of commonality and standardization. These units of logic are actually services. A service has the following characteristics [6]:

- *Reusable*: Services are designed to support potential reuse.
- *Loosely coupled*: Services must be designed to interact without the need for tight, cross-service dependencies.
- *Share a formal contract*: For services to interact, they need a formal contract that describes each service and defines the terms of information exchange.
- *Abstract underlying logic*: Underlying logic, beyond what is expressed in the descriptions that comprise the service contract, is invisible and irrelevant to service requesters.
- *Composable*: Services may compose other services, which allows logic to be represented at different levels of granularity and promotes reusability and the creation of abstraction layers.
- *Autonomous*: Service has control in an explicit boundary within which the logic resides.
- *Stateless*: Services should be designed to maximize statelessness, in order to not impede their ability to remain loosely coupled.
- *Discoverable*: Services should allow their descriptions to be discovered and understood by humans and service requesters.

Service modeling

A service is the building block of a SOA system. Service modeling is the key issue of developing a SOA system. Many SOA books, discuss service modeling [3, 4, 6]. All of the referenced books agree on one result of service modeling – service categories or a service hierarchy. Service categories or hierarchy groups services into logic units. The logic behind the categorization or hierarchy can be business-based or technology-based. The main aim is to facilitate service search, service composition and decomposition. Building of a service hierarchy can be the first step of service modeling, or the last one, depending on the type of approach, top-down or bottom-up. A step-by-step sample service modeling process is proposed by Erl [6]. When building a service hierarchy for a SOA system, the first step – Decompose Business Process – should be followed.

Service discovery

Normal word matching search is not adequate for our service search, as word matching search generates so many irrelevant results. Semantic Web services technologies provide us with possible solutions. Semantic Web services are Web services in which semantic Web ontologies ascribe meanings to published service descriptions so that software systems representing prospective service clients can interpret and invoke them. [5] In contrast to non-semantic Web services discovery which must be carried out manually by developers with query registries such as UDDI, semantic Web service discovery is made through an automatic *matchmaker* which process queries to find appropriate services from among those advertised using Semantic Web language descriptions. [22] The matching mechanism of a *matchmaker* bases on semantic descriptions of services.

RESULTS

The outcome of our research is a DQ assessment framework (see figure 1). The framework consists of the following four components:

- *Formal Quality Requirement*
- *Service Selection Process*
- *Service Repository*
- *Data Quality Ontology*

The *Formal Quality Requirement* is a set of keywords of data quality requirements extracted from the *User Quality Requirement* with the *Requirement Formalization Process*. The *Formal Quality Requirement* also serves as input to the *Service Selection Process*. The *Service Selection Process* communicates with the *Service Repository*, which stores DQ assessment services in two possible ways: either the *Service Selection Process* retrieves services from the *Service Repository* for the DQ requirements if there are services conforming to the requirements in the *Service Repository*, or if not, new services are added into the *Service Repository*. The *Data Quality Ontology* is used by the three components above as the semantic backbone; meanwhile the *Data Quality Ontology* is dynamically complemented by interaction with those three components.

We are going to take a close look at each component of the framework one by one.

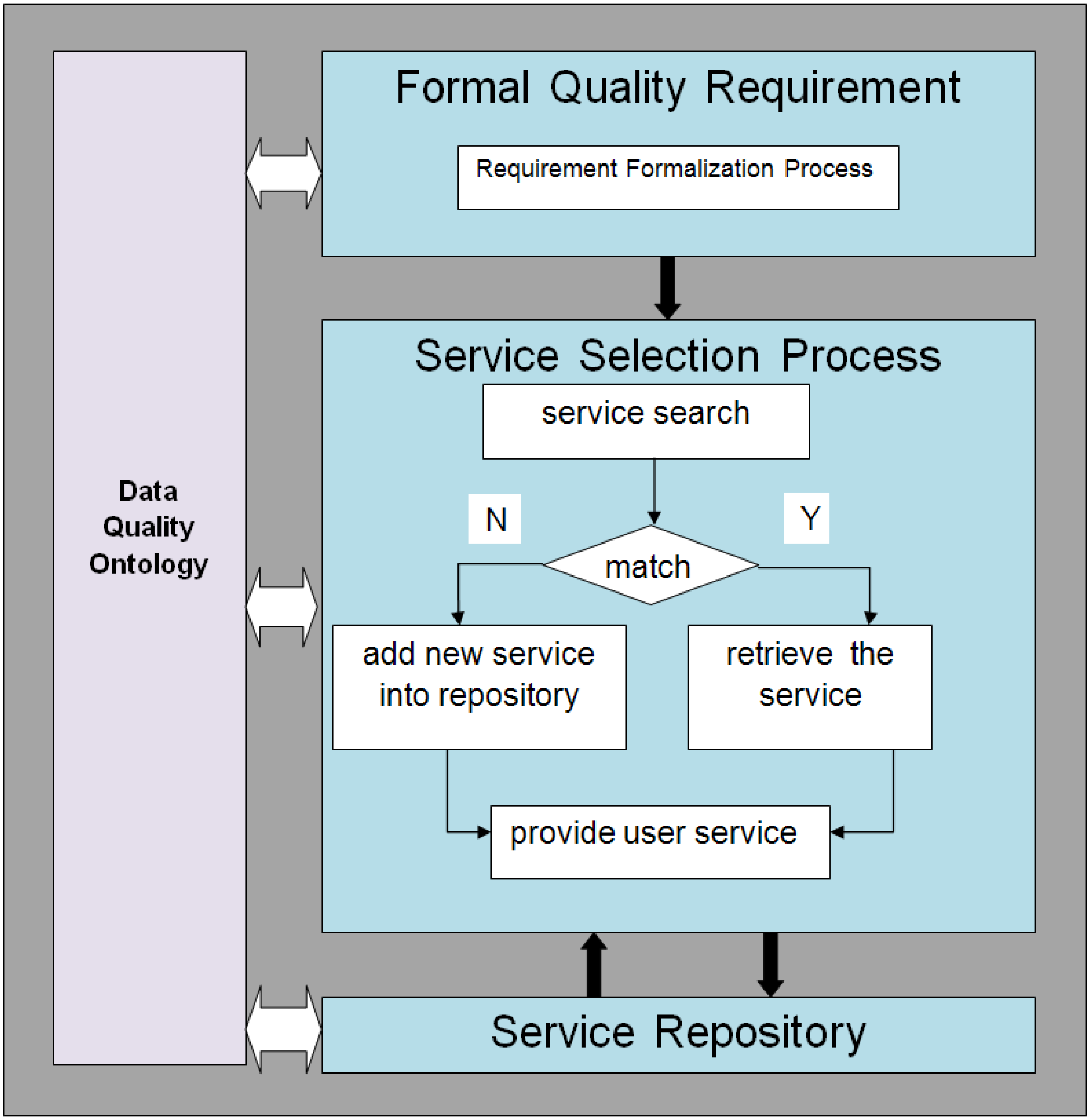

Figure 1 OpEN.SC DQ assessment framework

Data Quality Ontology

DQ is an entity which cannot be described from only one perspective. The DQ dimension [11] is one of the possible perspectives to define DQ. For example, a DQ assessment requirement can be "how believable is the data". However, in different use cases, believability has different implications. In the use case of clinical diagnosis, if the diagnoses of a patient record have been reviewed by a few doctors, or by a well-known expert, the patient record is considered to have high believability. However, the same patient record could be considered to have low believability in the use case of a laboratory, as the laboratory test result values are not accurate enough. This example shows that DQ should be described at least from two perspectives: DQ dimension and use case context. Schrader tries to organize all relevant perspectives for describing DQ into a DQ ontology [21].

Figure 2 is an excerpt from Schrader's DQ ontology [21]. The current DQ ontology version consists of five top-level classes: Quality Dimension, Metrics, Linguistic Property, Use Case Context, and Roles. Under each top-level class, there can be one or more levels of subclasses. In figure 2, two example subclasses of Quality Dimension, and one example subclass of Schema Completeness are shown. Each bottom-level class has more than one instance, such as Schema Completeness, which has three instances.

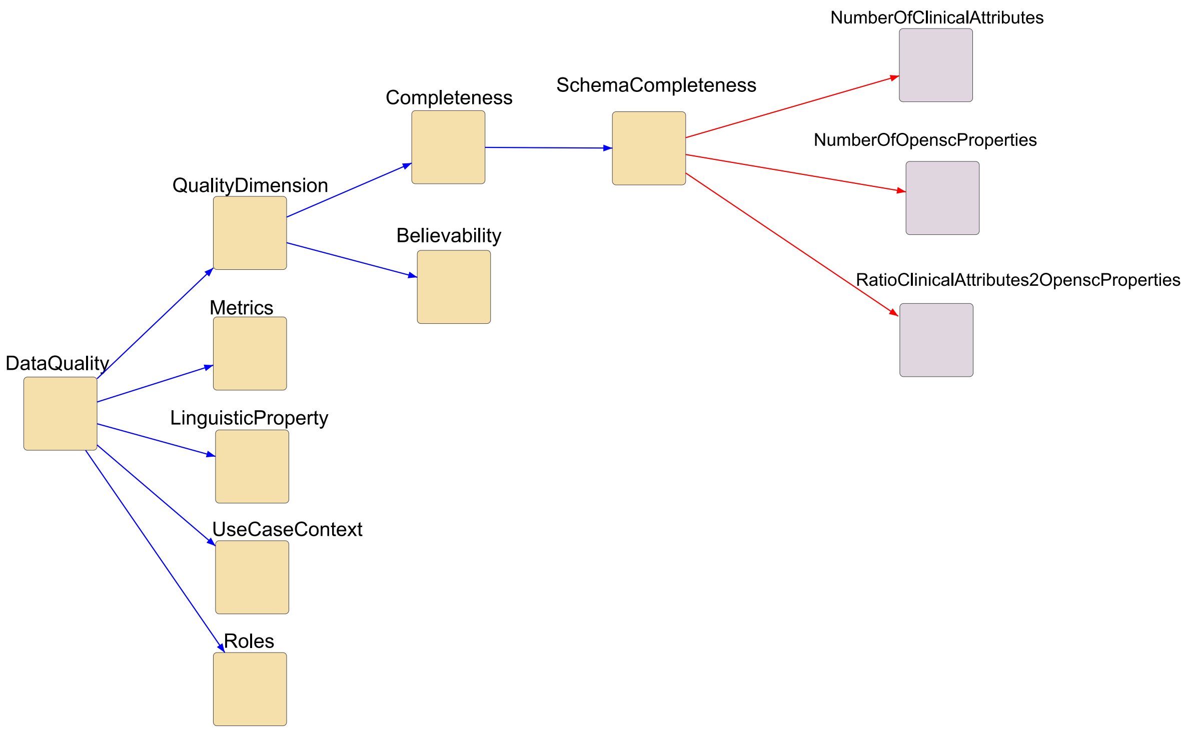

Figure 2 Excerpt of Schrader's DQ ontology

The *Data Quality Ontology* interacts with the other three components of the DQ assessment framework, and gets expanded by the growth of the other components. The other three components need the *Data Quality Ontology* as an ontology reference. In the following part, the interaction of each component with the ontology will be explained in detail.

Formal Quality Requirement

The *Formal Quality Requirement*, in contrast to the *User Quality Requirement,* is a set of keywords extracted from the *User Quality Requirement*. The *User Quality Requirement* is the source of the *Formal Quality Requirement*. With the help of the *Data Quality Ontology* and the *Requirement Formalization Process*, keywords of *User Quality Requirement* are extracted. The *Requirement Formalization Process* is still under research. A possible solution is to let the user define his or her *User Quality Requirement* by filling in a template which is designed based on the *Data Quality Ontology* but is expressed in a more understandable way for the user. That means there is a mapping between *Data Quality Ontology* classes to the *User Quality Requirement* template. The following example shows a user's quality requirement expressed in unstructured text in the ellipse, describing his or her background and ideas. The *User Quality Requirement* displayed in the rectangle frame shows how the requirement is expressed in the template. Table 1 shows the resulting basic contents of the *Formal Quality Requirement* in the form of keywords.

A medical doctor has collected a list of patient records related to his research topic by searching in the OpEN.SC database. He finds out that the diagnoses of some patient records are questionable. It is common in the medical field that for any given patient case, a different doctor has different opinions on the diagnosis. The doctor wants to know how reliable the diagnoses are. The other doctors or experts may be asked to do this evaluation.

1. What is important to you regarding the quality of the patient data?
    - Believability
    - Timeliness
    - Free of error
2. What can be done to improve on the believability?
    - Get more opinions
    - Ask experts
3. How should the data quality be presented?
    - Text
    - Graph
    - Chart

Ontology class	Keywords
Quality dimension	Believability
Use Case Context	Different opinions
Roles	Data customer
Metrics	String
Linguistic Property	Semantic

Table 1 Example: basic contents of a *Formal Quality Requirement*

Service repository

The service modeling methods mentioned in the Method section have shown us that building a service hierarchy is the key issue of service modeling. As mentioned before, building a service hierarchy can be either the first step or last step depending on the approach taken. OpEN.SC DQ assessment takes a top-down development approach. Therefore, we first build up a service hierarchy in order to store future services. From the discussion about service modeling in the Method section, we know that the *Decompose Business Process* is a way to build a service hierarchy. The business goal of OpEN.SC DQ assessment is to show the DQ assessment results in the required way for certain data by going through the DQ assessment process. This business goal can be reached in two steps. The first step is to get the DQ assessment results. The second step is to display data with these results. Hence, we divide the first level of our DQ assessment hierarchy into two branches: *Quality Information Gathering* and *Data Quality Labeling* (see figure 3). The former gathers quality information, which is actually the assessment result of the data, and the latter labels data with the gathered information.

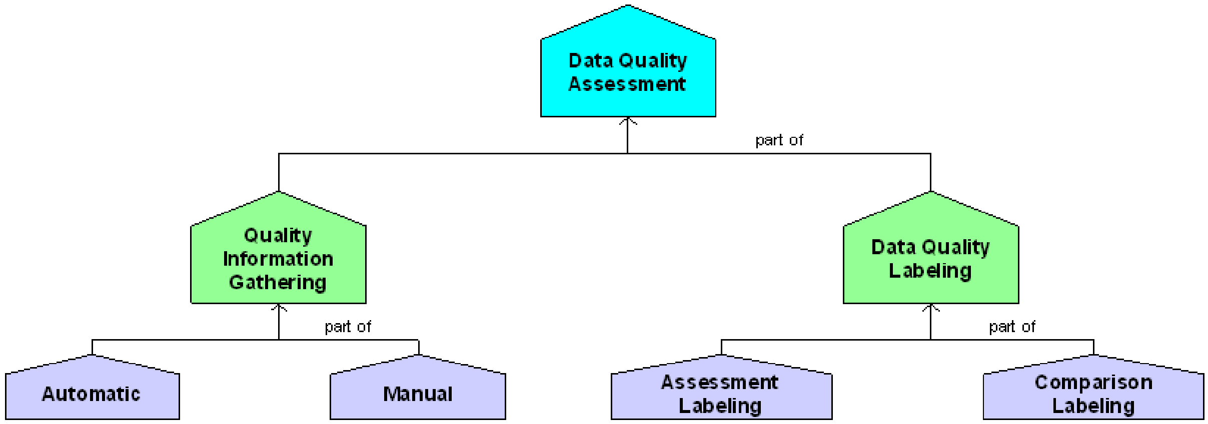

Figure 3 OpEN.SC DQ assessment service repository hierarchy

Further decomposition of the business process leads to the branches of the second level of our service hierarchy. *Quality Information Gathering* services are divided into two branches: *Automatic* and *Manual*. The *Automatic* branch consists of services which gather DQ information without human intervention. *Manual* services gather DQ information by involving human activities. *Data Quality Labeling* services are also divided into two branches: *Assessment Labeling* and *Comparison Labeling*. *Assessment Labeling* services are those which display DQ assessment results, such as graphic measurement result. *Comparison Labeling* services are those which present DQ comparison results, such as a deviation comparison.

In service modeling, the design of the *Service Repository* hierarchy can make a service search more efficient. We are going to introduce the *Service Class* and *Service Instance* structures into our *Service Repository*. Hence, under the *Automatic* branch, there will be a *Service Class* first. Under the *Service Class* will be a set of *Service instances*.

As mentioned in the Method section, service discovery needs an ontology to ascribe meanings to service descriptions. This is how the *Service Repository* uses the *Data Quality Ontology*. Meanwhile, new services may contain new ontology entries, so the *Service Repository* also inputs new ontology entries into the *Data Quality Ontology*.

Service Selection Process

The initial *Service Repository* can be an empty hierarchy without any service. When a DQ assessment requirement arrives, the *Service Selection Process* refers to the *Data Quality Ontology* to find relationships between classes and instances to carry out a service search by matching the *Formal Quality Requirements* with service descriptions. If a suitable *Service Class* is found in the *Service Repository*, we further search which *Service Instance* is most suitable for the requirement. Then, the scientists who need it can use it immediately. Otherwise, we analyze the requirement, and develop or integrate a corresponding service or services to meet this requirement. Then, we add the new service or services into the *Service Repository* under the proper *Service Class*. In addition, the *Service Selection Process* also produces new service development requirements if there are no matching services in the *Service Repository*. Therefore, the *Service Selection Process* will input the relationships between newly-added ontology classes and instances into the *Data Quality Ontology*.

Following is an example of how the *Service Selection Process* works. In this example, a process of adding new services to the *Service Repository* to meet the example DQ assessment requirements is shown (see table 1 from the section *Formal Quality Requirement*). Firstly, with the input of the keywords in table 1 and reference to the relationship between the keywords and service descriptions in *Data Quality Ontology*, the service search engine checks the *Quality Information Gathering* branch of the Service Repository to see whether there is any service - either automatic or manual- matching the keywords. We cannot find such a service, as the repository may still be in its initial empty status. Then, the developer has to analyze the *Formal Quality Requirement* to design a *Service Class* to meet this requirement. Using the technology of service modeling, we analyze and design a service class called Teleconsultation (see figure 4). A Teleconsultation *Service Class* contains services to increase the *believability* of OpEN.SC's medical data by routing certain medical cases to a group of experts to ask them to give *different opinions* about the case assigner's questions on the diagnosis. Secondly, we design and construct a Teleconsultation *Service Instance* Case Routing Teleconsultation which works by assigning tasks to other experts to ask them for opinions on a certain medical case. The other *Service Instance* Net Conference Teleconsultation implements the Teleconsultation *Service Class* by organizing an online conference of experts to discuss a certain medical case. Thirdly, we add the Teleconsultation *Service Class* and its two instances to the *Service Repository* under the branch of *Quality Information Gathering*, under the node *Manual*.

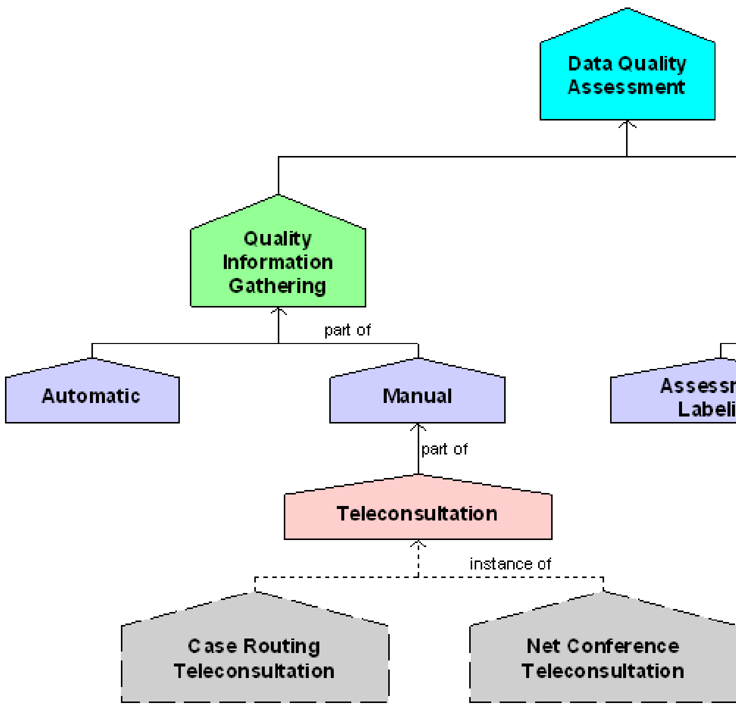

Figure 4 OpEN.SC data quality assessment service repository

DISCUSSION

DQ assessment research has resulted in many academic and practical solutions for *DQ problem classification*, *DQ dimension identification* and *DQ assessment methodologies*. They are the theoretical foundation for the formation of our DQ assessment framework. Most of OpEN.SC's DQ problems originate from evolving medical research DQ assessment requirements for the research source data. The ontology makes it possible to express DQ assessment requirements and describe services semantically, so that a service search can be done more accurately. SOA and Web service technologies allow us to build a *Service Repository* and a *Service Selection Process* to provide proper DQ assessment tools for particular DQ assessment requirements. All these help realize our innovative DQ assessment approach: giving users the freedom to define their own DQ assessment requirements, and providing them with proper DQ assessment tools.

Our DQ assessment framework is an attempt to enable DQ assessment on demand. There is a need for it, especially in a medical research center like OpEN.SC, but former approaches mainly focus on one part of the DQ assessment field, either the DQ requirement definition, or the DQ assessment methodology. Our framework is based on previous research on DQ assessment, and tries to combine the research contributions in order to provide a comprehensive solution covering more parts of the DQ assessment process.

LIMITATIONS

A special DQ assessment tool for a particular DQ assessment requirement can help users find out about the data quality. However, when there are few services in the repository, it may take a relatively long waiting time for users until a DQ assessment tool becomes available to them. Nevertheless, with the increasing number of requirements comes an increasing number of services in the *Service Repository*. In addition, services are composable and decomposable, which will speed up the delivery of services for new DQ assessment requirements. Moreover, services are added to the repository not only as newly developed or composed services driven by DQ assessment requirements. Instead, existing services which are accessible through the network can also be added. Therefore, the waiting time for a suitable DQ assessment tool will decrease over time with the increasing number of DQ assessment services in the *Service Repository*.

CONCLUSION & FUTURE WORK

Data in OpEN.SC is clinical data, especially patient records. As medical research has its own specific criteria for source data, only certain patient records which meet these criteria can be used for medical research. Moreover, different kinds of medical research have different criteria for source data. In order to help medical scientists find satisfactory research source data in OpEN.SC, in this paper we present a DQ assessment framework, based on an ontology and SOA, to allow users to define their own DQ assessment requirements and to be provided with proper DQ assessment tools. Another contribution this framework makes to DQ assessment is that it covers more parts of the DQ assessment process than previous attempts, from the requirement definition to the methodology decision. The development of this framework has only taken its first step. There is still a large amount of work to do to complete it, and there are many interesting research topics waiting for us ahead. For example, the use of ontology will not be limited only to the *Data Quality Ontology*, but may also include other ontologies such as a Web Service description ontology in order to make the service search more accurate. Moreover, the service search may not only be limited to a keyword-based search, but may also offer a more complicated semantic annotation-based search. In the end, the framework will be evaluated by medical doctors.

REFERENCES

[1] Arsanjani, A. "Service-oriented modeling and architecture." *IBM DeveloperWorks*, 2004, https://www.ibm.com/developerworks/library/ws-soa-design1/

[2] Berg, M. "Medical work and the computer-based patient record: a sociological perspective." *Methods of Information in Medicine,* 37(3). 1998. pp.294-301

[3] Bell, M. *Service Oriented Modeling*. John Wiley & Sons. Hoboken: N.J., 2008

[4] Bieberstein, N., Laird, R.G., Jones, K., and Mitra, T. *Executing SOA.* IBM Press. Boston:M.A.,2008

[5] Burstein, M., Bussler, C., Zaremba, M., Finin, T., Huhns, M., Paolucci, M., Sheth, A., and Williams, S. "A Semantic Web Service Architecture." *IEEE Internet Computing,* 9(5). 2005. pp. 72-81

[6] Erl, T. *Service-oriented Architecture Concepts, Technology and Design*. Prentice Hall. Upper Saddle River: N.J., 2005

[7] Garvin, D. A. *Managing Quality 3rd Edition.* Blackwell Publishing. Cowley Road: Oxford, 1999

[8] Ge, M., and Helfert, M. "A Review of Information Quality Research." *Proceedings of the 12th International Conference on Information Quality*, 2007

[9] Gruber, T. "Ontology." *Entry in the Encyclopedia of Database Systems, Liu, L. and Özsu M.T. (Eds.)*, Springer-Verlag, 2008

[10] Huang, K., Lee Y.W. and Wang R.Y. *Quality Information and Knowledge*. Prentice Hall. Upper Saddle River: N.J., 1999

[11] Kahn, B., Strong, D., and Wang, R.Y. "Information Quality Benchmarks: Product and Service Performance." *Communications of the ACM*, 45(4). 2002. pp. 184-192

[12] Klein, M., Fensel, D., Harmelen, F., and Horrocks, I. "The Relation between Ontologies and Schema-languages: Translating OIL-specifications in XML-Schema." *Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods, 14th European Conference on Artificial Intelligence*, 2000

[13] Lee, Y., Strong, D., Kahn, B., and Wang, R. Y. "AIMQ: A Methodology for Information Quality Assessment." *Information & Management*, 40 (2). 2002. pp. 133-146

[14] Lesca H. and Lesca E. *Gestion de l'information: qualité de l'information et performances de l'entreprise.* LITEC, Les essentiels de la gestion, 1995

[15] Lindemann, G., Schrader, T., and Schmidt, D. "The Aim of the Open European Nephrology Science Center - OpEN.SC." *Proceedings of 15th Workshop on Concurrency, Specification & Programming*, 2006. pp. 480-487

[16] Lindemann, G., Schmidt, D., Schrader, T., Beil, M. and Burkhard, H. D. "The System Architecture of the Open European Nephrology Science Center." *World Academy of Science, Engineering and Technology*, 34. 2007. pp. 275-279

[17] Lindemann, G., Schmidt, D., Schrader, T., and Keune D. "The Resource Description Framework (RDF) as a Modern Structure for Medical Data." *World Academy of Science, Engineering and Technology*, 31. 2007. pp. 240-243

[18] Lyman, J., Scully, K., and Harrison, J. "The Development of Health Care Data Warehouses to Support Data Mining." *Clinics In Laboratory Medicine,* 28. 2008. pp. 55-71

[19] Pipino L., Lee Y.W., and Wang R.Y. "Data Quality Assessment." *Communications of the ACM,* 45(4). 2002. pp. 211-218

[20] Savchenko, S. "Automating Objective Data Quality Assessment." *Proceedings of the 8th International Conference on Information Quality*, 2003

[21] Schrader, T., Zhou, Y., and Hanss, S. "The problem of data quality in medical research science centers." *Proceedings of 17th International Conference on Concurrency, Specification and Programming*, 2008. pp. 463-470

[22] Sycara, K., Klusch, M., Wido, S., and Lu, J. "Dynamic Service Matchmaking Among Agents in Open Information Environment." *J. ACM SIGMOD Record, special issue on semantic interoperability in global information systems,* 28(1). 1999. pp.47-53

[23] Szirbik, N. B., Pelletierm, C., and Chaussalet, T. "Six methodological steps to build medical data warehouses for research." *International Journal of Medical Informatics*, 75. 2006. pp. 683-691

[24] Wand, Y. and Wang, R. Y. "Anchoring data quality dimensions in ontological foundations." *Communications of the ACM*, 39 (11). 1996. pp.86-95