

A Checklist-Based Approach for Quality Assessment of Scientific Information

Jun Zhao¹, Graham Klyne¹, Matthew Gamble², and Carole Goble²

¹ Department of Zoology, University of Oxford, Oxford, OX1 3PS, UK
jun.zhao, graham.klyne@zoo.ox.ac.uk

² Computer Science, University of Manchester, Manchester, M13 9PL, UK
m.gamble@cs.man.ac.uk, carole.goble@manchester.ac.uk

Abstract. The Semantic Web is becoming a major platform for disseminating and sharing scientific data and results. Quality of these information is a critical factor in selecting and reusing them. Existing quality assessment approaches in the Semantic Web largely focus on using general quality dimensions (accuracy, relevancy, *etc.*) to establish quality metrics. However, specific quality assessment tasks may not fit into these dimensions and scientists may find these dimensions too general for expressing their specific needs. Therefore, we present a checklist-based approach, which allows the expression of specific quality requirements, saving users from the constraints of the existing quality dimensions. We demonstrate our approach by two scenarios and share our lessons about different semantic web technologies that were tested during our implementation.

1 Introduction

Information quality assessment aims to provide an indication of the fitness of information. Most existing approaches perform the assessment by integrating assessment of a number of quality dimensions, such as accuracy, completeness, or believability. We argue that although such methodology provides a systematic framework to organise quality assessment, it leaves two outstanding issues: 1) the quality dimensions used are often too abstract and generic for expressing concrete quality requirements, and 2) constrained frameworks are often unable to address different uses a consumer may have for a common resource: data fit for one purpose might not be fit for another. Although quality dimensions are often specialised to support assessment requirements from a specific domain or task, *e.g.* as a formula to compute a quality value by using a certain set of information, such specialisation cannot always be flexible enough to support different quality needs that might arise from different tasks to be applied to the same information. For example, the set of information considered sufficient for supporting access to a linked data resource might not be enough for assessing its freshness. Users need a flexible way to express their different quality requirements according to the task at hand.

This paper addresses these issues by proposing a flexible and extensible data model to support explicit expression of quality requirements. We draw upon the idea of checklists, a well-established tool for ensuring safety, quality and consistency in complex operations, such as manufacturing or critical care [4, 10]. A checklist explicitly defines a list of requirements that must be fulfilled or assessed for a given task. In our checklist-based framework we provide an OWL ontology, the *Minim ontology*, to express quality requirements as RDF, and an assessment tool to evaluate the conformance of target data against a Minim checklist. We demonstrate Minim in practice by applying it to support two quality assessment scenarios: the quality of scientific data, and scholarly artefacts.

The contributions of this paper are: 1) presenting a flexible and extensible data model for explicitly expressing quality requirements according to users' assessment needs; and 2) providing a comparison of several state-of-the-art semantic web technologies in supporting quality assessment tasks, which are learnt from our practical experience. The Minim model presented in this work is an updated version of our previous work [14], which provide two new distinct features: 1) more explicit representation of individual quality requirement as a type of test; and 2) an extensible structure for users to add requirements or tests that are not defined in the model, in order to cope with new emerging requirements from their own domains.

2 Motivating Scenarios

In this section we present our motivating quality assessment scenarios from the scientific and scholarly publishing domains. The scenarios illustrate how our checklist framework can be used to support specific quality assessment tasks. Although these requirements could be fit into a conventional quality dimension, such as correctness or completeness, our approach saved the users from having to take the extra step of identifying the relevant quality dimensions, which is commonly required in an existing dimension-based methodology. Therefore, our scenarios highlight the advantage and convenience of being able to explicitly express the assessment requirements using our approach.

2.1 Quality assessment of scientific linked data

The volume of scientific data resources on the linked data web is rapidly expanding. However, their quality does not always stand up to scrutiny, an issue that is caused either by the linked data publication process or is intrinsic to the source data. **Scenario 1** shows how quality assessment can reveal a series of potential quality issues in a linked dataset that contains some basic metadata information about 7,572 chemical compounds. The dataset was used in a previous study [7] and it was created based on the InfoBox information of Wikipedia³. Because of

³ <http://en.wikipedia.org/>

the potential incompleteness of the information available from these InfoBoxes, the resulting linked dataset can also have some potential quality issues. For example, according to domain-specific recommendations, each chemical compound must have one and only one IUPAC International Chemical Identifier (InChI). A quality requirement like this can be easily expressed using the cardinality test construct in our checklist model (see section 3) and an assessment can be automatically performed against all the chemical compounds in the dataset.

2.2 Quality assessment for scholarly communication

Scholarly communication refers to a principled method of making *scientific artefacts* available in order to support their more effective interpretation and reuse. These artefacts include data, methods or tools that were used to generate the findings reported, and providing sufficient information is key to achieving this goal. This is an ongoing quality challenge in scholarly communication that has not been fully addressed.

Scenario 2 uses quality assessment to help boost the effectiveness of scholarly communication in practice. `myExperiment.org` [5] is a popular workflow repository for sharing and releasing scientific workflows, which are important first-class scientific artefacts documenting protocols used to generate experimental results. Re-use of these workflows relies on adequate documentation to facilitate understanding and re-purposing.

A previous study analysed a representative selection workflows from `myExperiment.org` and drew out a minimal set of information that supports their re-execution [14]. This information, presented as a quality checklist, can be used to prompt workflow authors to provide better documentation about the workflows. This early intervention enhances the quality of scholarly communication.

2.3 Summary

No quality dimensions need be mentioned in the quality requirements of our scenarios. Instead, these requirements can be directly expressed using the constructs of our checklist data model, see sections 3 and 6. This provides a novel approach to quality assessment, in comparison to most of the existing work.

3 Approach

Our checklist-based assessment approach is based on two central pieces: 1) a container data model for encapsulating the RDF data/graph to be evaluated, and 2) the Minim data model, for representing quality requirements as a checklist.

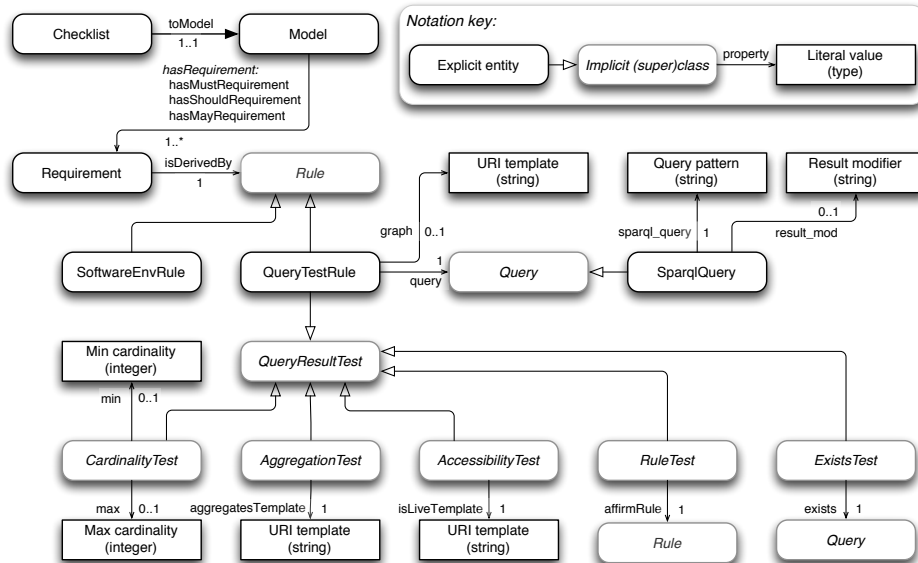
3.1 Research Object Model as a Container

We use an existing data model, namely the Research Object (RO) model [1], for our assessment. This provides a lightweight ‘container’ structure for encapsulating RDF and associated data. Annotation data contained within the RO constitutes the collection of RDF descriptions to be evaluated.

3.2 The Minim Model for Expressing Quality Requirements

A *checklist* provides an overall assessment of a dataset for some purpose. It consists of a number of individual *checklist items* which may address specific values within a dataset (typically at the level granularity accessible by a SPARQL query). Borrowing from IETF practice⁴, individual items have a MUST, SHOULD or MAY requirement level. A dataset may be “fully compliant”, “nominally compliant” or “minimally compliant” with a checklist if it satisfies all of its MAY, SHOULD or MUST items respectively.

Fig. 1. An overview of the Minim model schema.



Our Minim data model (see Figure 1) provides 4 core constructs to express a quality requirement:

- `minim:Checklist`⁵, to associate a RO context, a target (the RO or a resource within the RO) and an assessment purpose (*e.g.* runnable workflow) with a `minim:Model` to be evaluated.
- `minim:Model`, to enumerate the requirements (checklist items) to be evaluated, with corresponding MUST, SHOULD or MAY requirement levels.
- `minim:Requirement`, which is a single requirement (checklist item) that is associated with a `minim:Rule` for evaluating whether or not it is satisfied or not satisfied.
- `minim:Rule`: There are several types of rules for performing different types of evaluation of the supplied data. Currently we have `minim:SoftwareEnvRule`, which tests to see if a particular piece of software is available in the current execution environment, and `minim:QueryTestRule`, which uses a query-based approach to assess the fitness of a target.

⁴ <http://tools.ietf.org/html/rfc2119>

⁵ The namespace of `minim` is `pur1.org/minim/minim#`.

The following script, expressed using Turtle format, defines an example Minim checklist, which is to be used to assess each chemical compound must have exactly one InChI number. The checklist has one requirement that *must* be satisfied (line 9), i.e.,:InChI. The test of this rule is expressed by a SPARQL query (lines 19-20), which searches for the InChI identifier of a compound. The cardinality rule (lines 22-23) specifies that there must be exactly 1 matching query result associated with an evaluated compound.

```

1 :runnable_workflow a minim:Checklist ;
2   minim:forTargetTemplate "{+targetro}" ;
3   minim:forPurpose "complete" ;
4   minim:toModel :minim_model ;
5   rdfs:comment "" Checklist to be satisfied if
6     the chemical description is adequate."" .
7
8 :minim_model a minim:Model ;
9   minim:hasMustRequirement : InChI .
10
11 : InChI a minim:Requirement ;
12   rdfs:comment ""Ensures exactly one chembox:StdInChI value
13     is defined on the target resource, and that its value is
14     a string literal."" ;
15   minim:isDerivedBy [
16     minim:query
17       [ a minim:SparqlQuery ;
18         minim:sparql_query
19           ""?targetres chembox:StdInChI ?value .
20             FILTER ( datatype(?value) = xsd:string ) "" ;
21         ] ;
22     minim:min 1 ;
23     minim:max 1;
24     minim:showpass "InChI identifier is present" ;
25     minim:showfail "No InChI identifier is present" ;
26   ] .

```

In the current checklist implementation the `minim:QueryTestRule` is used to handle most of the checklist requirements we encounter. It can be associated with two elements: a query pattern (`minim:Query`) (lines 16-26), which is evaluated against the RDF data from the RO, and an optional external resource, which contains additional RDF statements that may be needed to complete the assessment. Every `minim:QueryTestRule` incorporates a `minim:QueryResultTest`, which takes the query result (which in our current case, a SPARQL query result) and returns a True (pass) or False (fail) result according to the type of test performed. Currently our Minim model defines 5 types of tests.

- `minim:CardinalityTest`, evaluates the minimum and/or maximum number of distinct matches in the query result against the declared conditions.
- `minim:AccessibilityTest`, evaluates whether a target resource indicated by the query result is accessible, by for example performing an HTTP HEAD request to the resource URI.
- `minim:AggregationTest`, tests the presence of resources in an RO that is used as the input to our assessment.
- `minim:RuleTest`, defines the additional rules to be applied to the assessment results returned from the evaluation of another `minim:QueryTestRule`. In this way, we can avoid writing too big rules and combine different types of rules, for example a query test rule with a liveness test rule.

- `minim:ExistsTest`, which can be used as a shortcut for a structure that combines a `minim:RuleTest` and `minim:CardinalityTest` to evaluate the existence of a particular resource in the evaluated data.

The Minim model is a refactor of our previous work [14], which addressed quality needs for enhancing scholarly communication (such as scenario 2). It has been extended by 1) explicitly defining an expandable set of test types; and 2) providing extension points allowing definitions of new assessment rules, assessment tests, and types of queries used to perform query-based tests (see *Rule*, *Query* and *QueryResultTest* in Figure 1).

Clearly, not every measure of quality can be evaluated automatically. For example, establishing correctness of stated facts may require independent validation [13]. Our approach allows direct tests to be combined with such independent validation or review, the latter of which may be simply expressed as quality metadata about the target dataset. A systematic assessment of how our checklist-based approach can support most of the existing known quality dimensions is a key part of our future work. Our focus on extensibility allows new automatic assessments to be introduced in a principled fashion. Examples of checklists that combine automatic evaluation with manual review may be found in our GitHub repository ⁶.

4 Implementation: The Minim Checklist Framework

The checklist framework is implemented in Python as both a command-line tool, `ro-manager`, and a RESTful service^{7,8}. Source code is in GitHub⁹.

As shown in Figure 2, the evaluation framework takes four inputs: a Research Object (RO) that containing a set of RDF annotations, a Minim file, a purpose indication, and an optional target resource URI (if not specified, the RO itself is the target). The framework uses a checklist from the Minim file selected by the purpose and target, applying each of the assessment tasks described by each checklist item to the RDF graph presented by the Research Object.

We chose SPARQL to express the `QueryTestRules` within a Minim checklist, as SPARQL is a widely available standard for querying and accessing RDF data. Our comparison with other semantic web technology choices is presented in Section 6.

The assessment result contains quite extensive content in the form of an RDF graph. For web applications using these results, our implementation provides two additional services that return JSON or HTML checklist results that facilitate presentation of a more user-friendly “traffic-light” display, with “green ticks” for satisfied requirements, and “red crosses” and “yellow crosses” meaning failure of a MUST and SHOULD requirement respectively.

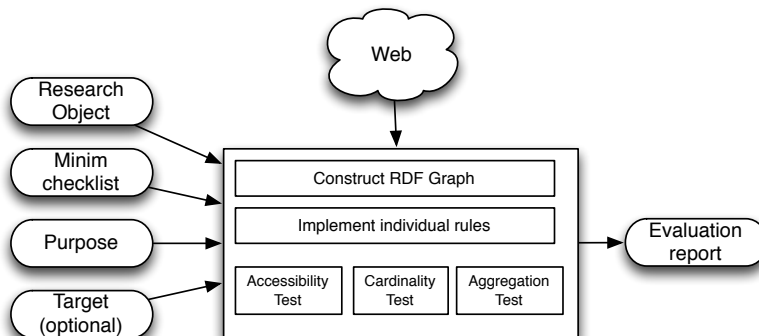
⁶ <https://github.com/wf4ever/ro-catalogue/tree/master/minim>

⁷ <http://purl.org/minim/checklist-service>

⁸ Example REST service use is at <https://github.com/wf4ever/ro-catalogue/blob/master/minim/REST-invoke-checklist.sh>

⁹ <https://github.com/wf4ever/ro-manager/>

Fig. 2. An outline of the checklist evaluation implementation



5 Quality Assessment in Action

In this section we show how the two motivating scenarios can be supported by our checklist tool. All the resources used for these case studies can be accessed in our Github repository¹⁰. Our exercise shows that our model and tool can sufficiently support assessment tasks from diverse domains, and at the same time, enable an explicit representation of the quality requirements from these tasks, which themselves can be valuable asset to a community.

5.1 Assess quality of scientific data using community checklist

In the first practical assessment we show how our checklist tool can be used to express existing community checklists from scientific domains in order to identify any potential quality issues of a scientific linked dataset. This actually reproduces the assessment by the previous MIM study [7] in our first motivating scenario. We reuse the chemical compound linked data and the checklist requirements defined in that study.

In that study 11 quality requirements were defined, based on a guideline from the chemistry domain. We analysed the tests required by each requirement¹¹ and categorised them into 3 different types: existence of information, type of information present, and cardinality of values provided. Our Minim model can be used to express these types of test, and the complete Mimim representation of these requirements is in our Github repository. We applied this checklist to 100 (limited by a performance constraint of the RO access mechanism used, currently being addressed) of the total 7,572 chemical compounds used in [7] and our checklist tool was able to reproduce exactly the same assessment result

¹⁰ <http://purl.org/minim/in-use-submission/>

¹¹ <https://github.com/wf4ever/ro-catalogue/blob/master/v0.1/minim-evaluation/checklist-item-survey.md>

as the MIM checklist tool. Whilst we see this limited assessment as sufficient to demonstrate that we can reproduce the results of the MIM checklist, future work (discussed in Section 8) will include a full validation for completeness.

5.2 Assess quality of scholarly communication research objects for specific purpose

In our second case study we apply our checklist tool to a set of scientific workflows from the myExperiment.org repository. These workflows commonly rely on a third-party bioinformatics Web service provided by a research organisation in Japan¹². At the end of year 2012, they announced that these services which were available as WSDL service would be upgraded to RESTful services and the WSDL service endpoints would no longer be supported, leading to failure of dependent workflows. Although it is impossible for them to be executable after the service upgrade, our assessment can enhance the quality of documentations about these workflows so that they can at least be understandable, repairable, and verifiable in the future.

Therefore, we designed a specific checklist, based on our previous analysis of causes to workflow quality issues [14]. In the checklist we define a list of requirements to be assessed, including: the presence of all input data; the presence of the workflow definition file; the presence of provenance logs of previous runs; and the accessibility of all the Web services used in a workflow.

22 workflows from myExperiment.org were applicable to our test. Our assessment managed to ensure that all the required information was associated with each workflow (see the full assessment result in our Github repository). After the service update took place, our checklist tool was able to successfully detect quality degradation for all the workflows and highlight explicitly the set of problematic services which caused the workflow no longer executable (see an example assessment result¹³). The assessment can be reproduced using resources in our Github repository.

6 Discussions

As an approach that is substantially based on semantic web technologies, the goals and features of our checklist-based framework can be seen to overlap with some major semantic web technologies like the Web Ontology Language (OWL)¹⁴ and SPIN¹⁵, which have been considered in our design process. However, our focus was to provide a higher level data model, which can more directly reflect quality requirements from users or specific scenarios. Although these semantic web technologies can be complementary to our approach, they cannot in isolation (fully) support all the quality assessment requirements identified from our scenarios.

¹² <http://www.genome.jp/kegg/>

¹³ <http://tinyurl.com/btxdlmv> - this is a live service link

¹⁴ <http://www.w3.org/TR/owl2-overview/>

¹⁵ <http://spinrdf.org>

6.1 Comparison with an OWL-based Approach

OWL ontologies support the description of classes that detail the features necessary for an individual data item to be a member of that class. These class descriptions are analogous to the description of requirements in our checklist. OWL also has an RDF serialisation and extends RDF semantics¹⁶ to operate over RDF data. We can express our InChI requirement in OWL as follows:

```
1 Class: InChI
2   SubClassOf: chembox:StdInChI some :InChIValue .
```

However, the current OWL 2 RDF semantics contain two features that are incompatible with our quality checking scenario:

- The Open World Assumption (OWA). If an InChI were to be defined without a corresponding InChIValue, this would not be highlighted as an error by an OWL reasoner. Instead the OWA results in the inference that there exists an InChIValue, but that it is currently unknown. This directly conflicts with our need for an existence check.
- No Unique Names Assumption. We can extend the above requirement to include a cardinality restriction to say that there must be one and only one InChIValue. The presence of two different InChI values would not however raise an error. Instead the assumption would be made that the two InChIValues are in fact the same. This directly conflicts with our need for cardinality checks in a quality assessment scenario.

An alternative to the traditional OWL 2 Semantics are Integrity Constraint Semantics (ICs)¹⁷. ICs are a semantics for OWL that employ a Closed World Assumption as well as a form of the Unique Names assumption. These semantics therefore allow the use of OWL classes to be interpreted as integrity constraints. The Stardog database¹⁸ currently provides an implementation of OWL with ICs.

One practical implementation of ICs is achieved by transforming the OWL classes to SPARQL queries. Each axiom in an OWL IC Ontology is transformed into a corresponding SPARQL query. This ability to realise ICs as SPARQL queries implies that by supporting a SPARQL based approach for requirement description, Minim achieves at least some of the expressiveness as an approach based upon OWL ICs. However, a purely OWL ICs based approach presents a number of restrictions with respect to what can be expressed in our requirements:

- Expression of different requirement levels such as MUST, SHOULD, and MAY. OWL IC semantics are primarily concerned with binary *satisfiability*, where we capture more nuanced levels of satisfaction. We believe would be more difficult to create checklists in OWL that capture these.

¹⁶ <http://www.w3.org/TR/rdf-mt/#MonSemExt>

¹⁷ <http://stardog.com/docs/sdp/icv-specification.html>

¹⁸ <http://www.stardog.com/>

- Flexibility and extensibility to perform broader resource accessibility and software environment tests that can be supported by our Minim tool. For example verifying the web-accessibility of workflow input files lies outside the expressive scope of OWL (though might conceivably be handled through the introduction of new primitive classes and OWL resoner extensions).
- Expressing rules that validate data literal values. This has previously been highlighted as a restriction of an OWL based approach to data validation in the life sciences [3].

6.2 Comparison with a SPIN-based Approach

SPIN provides a query-based modelling language to express rules and logical constraints over RDF data. It is used by the previously discussed MIM checklist-based assessment framework.

The property of `spin:constraint` can support a set of features in common with our Minim tool. `spin:constraint` can be associated with an `rdfs:Class`, *e.g.* `chembox:InCHI`, and defines the constraints that instances of the class should comply with. The constraints can be expressed using SPARQL ASK or CONSTRUCT queries that are expressed using SPIN syntax in RDF. This structure can be used to support most of our query-based tests, apart from the accessibility tests. Additionally, `spin:Template`, which provides a meta-modelling function to group SPARQL queries so that they can be reused, is very similar to the role of `minim:Rule` in our model. However, at the time of writing, SPIN was not yet established as a standard and implementations of SPIN engines were limited. A purely SPIN-based approach also shares the first two restrictions as an OWL ICs based approach, as analysed above.

6.3 Summary

OWL, OWL ICs, and SPIN are clearly complementary to our Minim model approach. Although they cannot be directly used to support expressing quality assessment requirements, they can complement our SPARQL-based implementation of the checklist tool. SPARQL was chosen for our tool implementation because it is a more established standard for querying RDF data, with a number of known implementations. Combined with our Minim model, SPARQL can support all the expression of constraints and most of the inference functions as SPIN. However, our Minim model can also be extended and implemented using these alternative technologies. The `minim:Query` class is one extension point for supporting SPIN-like queries, and `minim:Rule` can be extended to define other than query-based test rules.

7 Related Work

Zaveri et al. [13] provides a timely and extensive survey on quality assessment of linked data. The survey is mainly organised by quality dimensions rather than

the actual methodologies used by the reviewed works. Of the 21 works included in the review, a larger portion of them are based on specific algorithms, such as the trust evaluation by Golbeck [8], or use a dimension-driven approach, such as Bizer et al [2], or take a purpose-built approach to provide solutions to a specific problem in a specific application scenario, such as Guéret et al. [9]. 3 of the works take an approach more closely related to ours by supporting an explicit expression of quality requirements. However, the quality schema provided by Sieve [12] is rather simple, mainly targeted to express the configuration parameters and the functions to be used for the assessment; and the quality ontologies proposed by SemRef [11] and SWIQA [6] are based on a series of quality dimensions.

8 Conclusions and Future Work

Quality assessment is a paramount issue in supporting the successful re-use of Scientific Linked Data. Not being able to express specific quality assessment requirements according to the needs from specific assessment tasks has been a bottleneck to the quality enhancement of linked data resources. To fill in this critical gap, we propose a checklist-based approach that allows explicit expression of quality requirements that can directly reflect users' needs from their concrete quality assessment tasks, and at the same provides flexible extensibility to cope with new needs. We show how our approach can support two exemplar case studies from scientific domains. We learnt valuable lessons about how various state-of-the-art semantic web technologies could support our concrete use in practice. The very lightweight SPARQL-based implementation has shown great promise in supporting these practical needs.

Our next steps will focus on the extensibility of the tool architecture, by exploring the possibility of a plug-in framework to enable plugging-in of third-party services. We are also prototyping a user interface tool to facilitate the creation of Minim checklists. Finally we are planning a systematic mapping between the existing quality dimensions and the constructs available in our checklist data model, to extend the function evaluation of our model.

References

1. Khalid Belhajjame, Oscar Corcho, Daniel Garijo, Jun Zhao, and et al. Workflow-centric research objects: First class citizens in scholarly discourse. In *Proceeding of SePublica2012*, pages 1–12, 2012.
2. Christian Bizer and Richard Cyganiak. Quality-driven information filtering using the wiqa policy framework. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(1):1–10, 2009.
3. Jerven Bolleman, Alain Gateau, Sebastien Gehant, and Nicole Redaschi. Provenance and evidence in uniprotkb. In *Proceedings of the 3rd International Workshop on Semantic Web Applications and Tools for the Life Sciences*, Berlin, Germany, 2010.
4. Sangyoon Chin, Kyungrai Kim, and Yea-Sang Kim. A process-based quality management information system. *Automation in Construction*, 13(2):241–259, 2004.

5. D. De Roure, C. Goble, and R. Stevens. The design and realisation of the my-experiment virtual research environment for social sharing of workflows. *Future Generation Computer Systems*, 25:561–567, 2009.
6. Christian Fürber and Martin Hepp. Swiqa—a semantic web information quality assessment framework. In *Proceedings of European Conference on Information Systems*, 2011.
7. Matthew Gamble, Carole Goble, Graham Klyne, and Jun Zhao. Mim: A minimum information model vocabulary and framework for scientific linked data. In *E-Science (e-Science), 2012 IEEE 8th International Conference on*, pages 1–8. IEEE, 2012.
8. Jennifer Golbeck and Aaron Mannes. Using trust and provenance for content filtering on the semantic web. In *Proceedings of the Models of Trust for the Web Workshop*, 2006.
9. Christophe Guéret, Paul Groth, Claus Stadler, and Jens Lehmann. Assessing linked data mappings using network measures. In *Proceedings of the European Semantic Web Conference*, pages 87–102. Springer, 2012.
10. Brigitte M. Hales and Peter J. Pronovost. The checklist—a tool for error management and performance improvement. *Journal of critical care*, 21(3):231–235, 2006.
11. Yuangui Lei, Victoria Uren, and Enrico Motta. A framework for evaluating semantic metadata. In *Proceedings of the 4th international conference on Knowledge capture*, pages 135–142. ACM, 2007.
12. Pablo N Mendes, Hannes Mühleisen, and Christian Bizer. Sieve: linked data quality assessment and fusion. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pages 116–123. ACM, 2012.
13. Amrapali Zaveri, Anisa Rula, Andrea Maurino, Ricardo Pietrobon, Jens Lehmann, and Sören Auer. Quality assessment methodologies for linked open data. *Semantic Web Journal*. submitted on 12/14/2012.
14. Jun Zhao, Jose Manuel Gomez-Perez, Khalid Belhajjame, and et al. Why workflows break-understanding and combating decay in taverna workflows. In *IEEE eScience*, pages 1–8, 2012.