An Ontology-Based Approach to Handling Information Quality in e-Science

Paolo Missier, Suzanne Embury, Mark Greenwood
School of Computer Science, University of Manchester

Alun Preece, Binling Jin
Department of Computing Science, University of Aberdeen

Abstract

In this paper we outline a framework for managing information quality (IQ) in an e-Science context. In contrast to previous approaches that take a very abstract view of IQ properties, we allow scientists to define the quality characteristics that are of importance to them in their particular domain. For example, "accuracy" may be defined in terms of the conformance of experimental data to a particular standard. User-scientists specify their IQ preferences against a formal ontology, so that the definitions are machine-manipulable, allowing the environment to classify and organise domain-specific quality characteristics within an overall quality management framework. As an illustration of our approach, we present an example Web service that computes IQ annotations for experiment datasets in transcriptomics.

1 Introduction

Information is viewed as a fundamental resource in the discovery of new scientific knowledge. Scientists expect to make use of information produced by other labs and projects in validating and interpreting their own results. Funding bodies expect the results of projects to have much greater longevity and usefulness. As well as publishing their principal results in the scientific literature, scientists are now required to place a much greater proportion of their experimental data in the public domain. A key element of e-Science is the development of a stable environment for the conduct of these information-intensive forms of science. One significant obstacle is the class of problems that arise due to variations in the quality of the information being shared [3]. Data sets that are incomplete, inconsistent, or inaccurate can still be used to good effect by those that are aware of these deficiencies, but can be misleading, frustrating and time-consuming for those who are not.

Research in information quality (IQ) has tended to focus on the identification of generic quality characteristics (such as accuracy, currency and completeness) that are applicable in a wide range of application domains [10]. However, IQ is inherently use-specific, and this "one-size-fits-all" approach offers quality characteristics so broad in their meaning that they provide little discriminating information. An alternative is to try to identify the quality characteristics that are of importance in a particular domain. For example, one group of scientists may record "accuracy" in terms of some calculated experimental error, while others might define it as a function of the type of equipment that captured the data.

The domain-specific approach to the management of IQ for e-Science depends on two assumptions:

1. That it is possible to elicit detailed specifications of the IQ requirements of individual scientists or communities of scientists, preferably in a formal language so that the definitions are machine-manipulable. It must be possible for scientists to *use* the definitions, by creating executable metrics based on them, and also to *reuse* definitions created by others, by browsing and querying an organised collection of definitions.

2. That the annotation of information resources with detailed descriptions of their quality can be performed in a cost-effective manner. This means that the overhead of creating and managing the definition of a new IQ characteristic and its associated metrics should not be too high, and also that it should be possible to operationalise the computation of IQ measurements over sizeable datasets.

The Qurator project[1] aims to test these assumptions by making a detailed study of IQ management in two domains of post-genomic biology: proteomics and transcriptomics. As progress towards (1) above, this paper presents the initial version of our IQ framework for capturing scientists' IQ requirements. We use a motivating example from the domain of microarray data, and show how a domain-specific IQ characteristic can be defined as part of our overall framework. As an instance of (2), we introduce a Web service that automates one kind of IQ annotation of datasets, and apply this service to quality-assessment of microarray data.

[1]Funded by the EPSRC Programme Fundamental Computer Science for e-Science: GR/S67593 & GR/S67609 — *Describing the Quality of Curated e-Science Information Resources*, www.qurator.org.

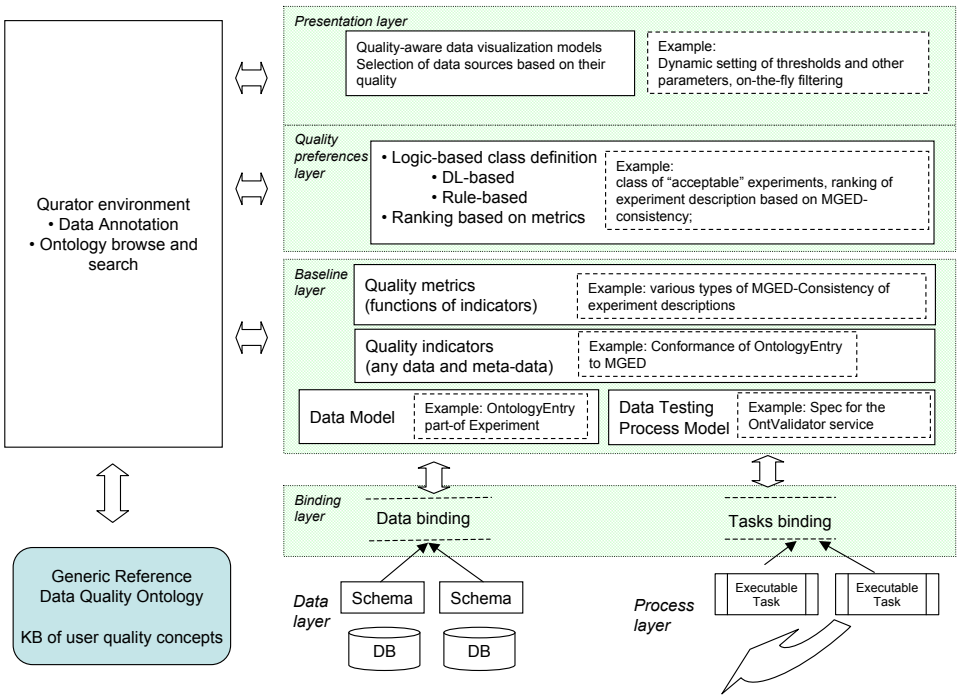

Figure 1: The Qurator conceptual framework

2 An IQ framework

We address our first assumption in Section 1 by providing the user with an environment for the creation, management and use of semantic annotations regarding the quality of data sets. The Qurator environment consists of formal models, languages, and software tools that let users express various facets of quality meta-data in a formal way, so that the resulting annotations can be (i) exploited by quality-aware data management applications, and (ii) shared with other users within a community of interest. The Qurator models are structured into a framework, shown in Figure 1, whose layers reflect the different levels of abstraction used during the annotation process.

At the core of this environment is an ontology for data quality, i.e., a formally-specified conceptualization of generic as well as domain-specific data quality concepts and terms [1]. The ontology, represented using the Web Ontology Language OWL[2], includes the main concepts of the framework; a small fragment is shown in Figure 2. For example, `QualityMetric` is a generic ontology concept whose semantic relationships to `QualityIndicator`, represented by the property `metric-based-on-indicator`, means that a metric is computed as a function of zero or more indicators. This root concept can be extended to include many different domain-specific, user-defined metrics, for instance `MGED-global-consistency`, described futher below.

Note the ontology only records the abstract relationship between concepts, i.e., a metric and an indicator, without specifying it further. A more complete specification is provided by the framework models, namely by the Quality Metrics model, which includes the definition of the actual function used to compute a specific metric like `MGED-global-consistency`. Thus, the ontology and the framework models play complementary roles: the ontology defines concepts across all the models in the framework, and provides an external view of the quality meta-data used in the annotation, and of their semantic relationships; while the different models provide a way to specify the internal semantics of each of these concepts. Each model is described using a potentially different, ad hoc formalism for the required specification.

The functions of each of the models, and of the ontology, are best described through a complete example. In transcriptomics, microarray experiment data is routinely captured using MAGE-OM, the MAGE Object Model recommended by the Microar-

[2]http://www.w3.org/2001/sw/WebOnt/

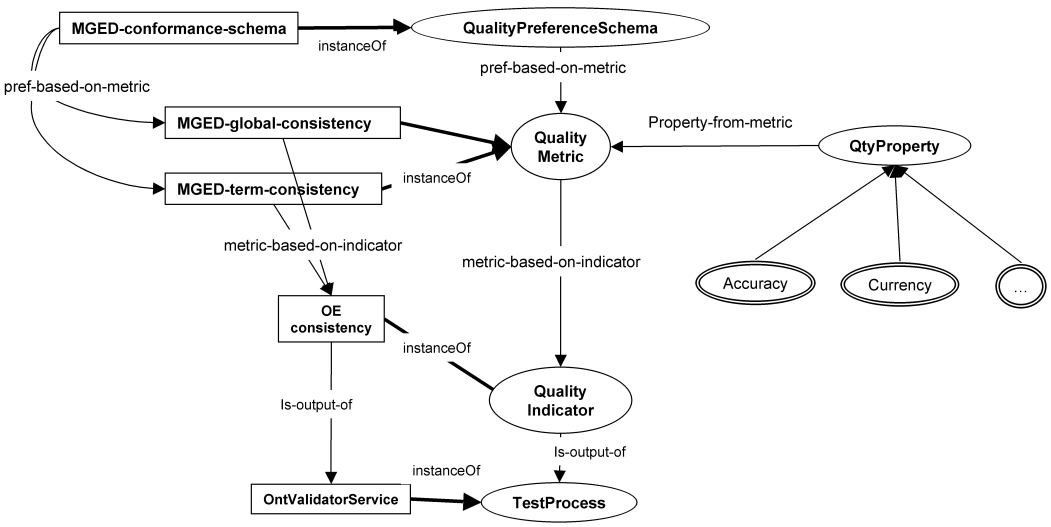

Figure 2: Small fragment of the Qurator quality ontology

ray Gene Expression Data Society (MGED)[3], and encoded using a standard XML syntax (MAGE-ML). The MGED Ontology[4] provides common terminology for describing all aspects of the experiment design and of its execution. Let us suppose that, in searching for suitable microarray experiment data within a database, a biologist decides to adopt the consistency of use of MGED terms as an indicator for the overall quality of the experiment. Also, for the sake of simplicity, we assume this to be the only indicator considered in the example; in reality, our experience suggests that this is one of several actual indicators that biologists are likely to use.

In general, a quality indicator for a piece of underlying data is an objectively measurable quantity whose value can be either computed from the data using an automated procedure, or be otherwise obtained interactively from the user. In our example, the MAGE standard prescribes which MAGE-OM entities, called `OntologyEntry` (OE for short), may refer to MGED Ontology entries. A sample XML fragment of a microarray experiment data file is shown in Figure 3, with the `OntologyEntry` elements highlighted. It is therefore possible to automatically verify the consistency of multiple references (possibly hundreds) to MGED terms, found in an experiment description. The consistency status of each OE thus becomes an elementary quality indicator that annotates the corresponding XML element. From this fine-grain collection of indicators, useful *quality metrics* can then be computed by aggregation, such as the fraction of consistent values over the entire collection, or the consistency of use of particular MGED terms across the entire experiment (indicated as `MGED-global-consistency` and `MGED-term-consistency` in Figure 2).

```
<BioSample
 identifier="S:Sample:MEXP:167278"
 name="CH131_1">
 <MaterialType_assn>
  <OntologyEntry
   category="MaterialType"
   value="whole_organism" />
 </MaterialType_assn>
 <Treatments_assnlist>
  <Treatment order="1"
  identifier="T:Sample:MEXP:167278">
   <Action_assn>
    <OntologyEntry
     category="Action"
     value="specified_biomaterial_action" />
   </Action_assn>
```

Figure 3: Fragment of a MAGE-ML/XML experiment data file, with highlighted `OntologyEntry` elements

This information is captured in the ontology as instances of existing concepts (the square elements in Figure 2), as well as in the *baseline* layer of the framework (lower part of Figure 1), which contains the data, test process, indicators and metrics models. The data model is an abstraction of the portions of the underlying data whose quality we are characterizing, that is, the experiment description, the OE elements, and their part-of relationships. The test process model describes the process used to compute the quality indicators, in our case an "OntValidator service" that produces the OE consistency annotations. Depending on the level of abstraction required, the model may define a service interface (for instance, the WSDL description of a Web service), or some of its implementation properties. The formalization should be adequate to the main purpose of the model, which is sharing and reuse of the annotations produced by the process. Similarly, the quality metrics model de-

[3] http://www.mged.org/
[4] http://mged.sourceforge.net/ontologies/MGEDontology.php

scribes how indicators are combined functionally into one or more metrics.

Underneath the baseline layer we find a *binding layer*, which accounts for the data mapping between the native representation of the data and its abstraction in the data model. In the example, the binding maps the `OntologyEntry` data entity to the set of XML elements in the actual experiment files, in this case using XPath expressions. Similarly, the process binding provides a mapping from the definition of a OntValidator service, to its implementation. Note that, if the service is a Web service, the binding is contained in its WSDL specification.

The scientist may then use quality metrics to formulate *quality preference schemas* that indicate how a quality-based view of the data can be produced using the metrics. Typical schemas include the ranking of the experiments collection based on the metric value, their classification into acceptable / non-acceptable classes based on a user-defined threshold, or a finer classification, for example, "top", "acceptable", or "fair". The purpose of the preference schema layer is to capture the details of how metrics are used to construct such schemas. Consistent with the other layers, this too has a counterpart in the ontology, where `QualityPreferenceSchema` concepts are associated to the underlying metrics.

The *preferences model* is used to represent these schemas, and it is perhaps the most interesting, considering the variety of types and of formalisms available to formulate quality-based rules. Using a general rule language like SWRL[5], one can define classes by giving necessary and sufficient conditions, like "an MGED-conformant experiment is one in which for at least 75% of ontology entries the references are consistent, and the experiment was submitted within the past 2 years". This expression makes a reference to one of the metrics mentioned earlier, as well as to a timestamp (defined in the Indicators model), and it partitions the space of experiments into two classes.

A finer classification can be defined using the same metrics, for instance "a *top* experiment is one in which more than 90% of ontology entries are MGED-consistent", "an *acceptable* experiment is one in which between 50% and 90% are consistent", and so forth. Given these class definitions and a dataset of annotated experiments, the Qurator environment will try to classify them according to the class definitions. Clearly, the ability to perform automatic classification depends on the types of conditions expressed by the user, and on the formalism used to express them. We are currently experimenting with various Description Logics [1], for which automated reasoners are available (OWL-DL being a prominent example). In the example, Qurator would use a DL reasoner against a knowledge base of semantic quality annotations, to automatically identify the MGED-conformant members of a dataset. Expressing preference schemas in OWL has the advantage that the preferences model can be naturally integrated into our quality ontology, resulting in more useful querying capabilities over the classified datasets.

The ontology is aligned with the ^{my}Grid project data ontology [11] (for example, `QualityMetric` is defined as a subclass of `mygrid:data`). While the framework allows for the definition of highly domain-specific (and scientist-specific) IQ preferences, it also allows for the classification of these preferences under a generic IQ categorisation drawn from the earlier literature (including [3, 10]). For example, the specific notion of MGED Ontology-conformance may be defined to be a special case of the generic notion of "Data Accuracy". Using this classification, a biologist could use our ontology to browse for specialisations of `Accuracy` pertinent to their own domain, and potentially reuse preferences defined by other scientists.

We exploit the flexibility of the Qurator environment to investigate various possibilities afforded by different class representations. Specifically, we are trying to determine what trade-offs between expressiveness and automated processing of class rules are most suitable for real-life quality preferences.

To conclude the description of the Qurator framework, we mention that preference schemas can in turn be used to provide quality-aware views of the data at the application level, using Qurator's presentation components, which are described in the *presentation layer*. For instance, a Qurator data filter that can access the quality preferences just described, as well as their underlying indicators and metrics annotations, is a presentation component that may be used in conjunction with a client database application to provide a quality-aware view of query results. Note also that in the example, metrics and preferences include user-defined parameters, namely the choice of MGED terms with respect to which metrics are computed, and some filtering threshold. Thus, an interactive presentation component that lets users change the parameter settings at query time may offer a rich and dynamic quality-oriented data manipulation environment.

It is worth noting that the framework by itself does not provide an architecture for the Qurator software environment. However, the stacking of the layers suggests dependencies among the models, which hold for the runtime processing of quality annotations as well. For instance, presentation components that exploit quality annotations may require that values be provided for them by the lower layers, that is, by computing metric functions and, if necessary, by providing the underlying indicator values for them. Thus, the Qurator conceptual framework translates quite naturally into a generic implementation framework

[5] http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

that can accommodate specialized software components, for instance for computing a particular metric function from a specific set of indicators.

3 An example IQ service: OntValidator

As a concrete example of the Qurator approach to IQ management, we have implemented the OntValidator ontology-conformance checking facility described in the previous section as a Web service. This section discusses this sample service and its current Web-based client. In general, OntValidator is designed to check the conformance of an XML representation of a dataset to a set of definitions specified in an ontology. To be realistic, we do not insist that the ontology is defined using one particular representation; nor do we require that the experiment data structure contain `OntologyEntry` elements.

The service requires two pieces of input information to be provided in the input SOAP message:

- The URI of an HTTP-accessible XML document containing the experiment data.
- An XML *control file* specifying:
  - the elements to check in the experiment data (as XPath expressions)
  - a reference to the ontology against which the conformance of the elements is checked (normally this will be either a "well-known" reserved term such as "`MGED`", or the base URI reference of a Semantic Web ontology such as "`http://mged.sourceforge.net/ontologies/MGEDontology.daml#`")

The OntValidator service returns a report in RDF/XML[6] which details the conformance of each specified element. This conformance report can then be used to generate IQ preference classifications and results for presentation according to a user's IQ preferences. RDF is used as it provides a natural way of making statements about Web resources, which is precisely what the OntValidator service is doing, and also integrates seamlessly with the OWL ontology.

For example, Figure 3 shows part of an XML document `PBMC_HIV_Patients.xml` containing a microarray experiment dataset in MAGE-ML; the highlighted parts of the figure show two `OntologyEntry` elements to be checked. Figure 4 shows a very simple control file which specifies (by XPath expression `//OntologyEntry`) that the OntValidator service should validate every `OntologyEntry` element of the user data file `PBMC_HIV_Patients.xml` against the MGED Ontology. A more elaborate control file could of course specify only a subset of the ontology entry elements (for example, those in a particular subtree of the document).

[6] http://www.w3.org/RDF/

```
<completeness-test-control refURL="PBMC_HIV_Patients.xml">
  <entry pathToNode="//OntologyEntry" ref="MGED"/>
</completeness-test-control>
```

Figure 4: An example XML control file provided to the OntValidator service

```
<rdf:Description rdf:nodeID="A1">
  <ontval:pathToNode>
    /MAGE-ML[1]/BioMaterial_package[1]
    /BioMaterial_assnlist[1]/BioSource[9]
    /Characteristics_assnlist[1]/OntologyEntry[1]
  </ontval:pathToNode>
  <ontval:qtyIndicatorValue>VAL_OK
  </ontval:qtyIndicatorValue>
</rdf:Description>
<rdf:Description rdf:nodeID="A2">
  <ontval:pathToNode>
    /MAGE-ML[1]/BioMaterial_package[1]
    /BioMaterial_assnlist[1]
    /LabeledExtract[10]/MaterialType_assn[1]/OntologyEntry[1]
  </ontval:pathToNode>
  <ontval:qtyIndicatorValue>VAL_BAD_IND
  </ontval:qtyIndicatorValue>
</rdf:Description>
```

Figure 5: An example RDF conformance report generated by the OntValidator service

In addition, Figure 5 presents part of an RDF conformance report as an example. In a conformance report, each `rdf:Description` element specifies the validity of an individual element of a dataset specified in the control file: the `ontval:pathToNode` property states the XPath of that element, while the `ontval:qtyIndicatorValue` property records the conformance value, explained below.

The OntValidator service is designed to cope with different ontology formats by means of plug-in handlers. In the current version of the Web service, an ontology handler has been implemented to validate the MAGE-ML experiment data against the DAML version of the MGED Ontology. (We chose the DAML version rather than the more recent OWL version for two reasons: firstly, our test experiment files were created against this version, and secondly, this emphasises the point that IQ tools must work with "legacy" data formats.) The DAML ontology handler is able to check conformance of both ontology classes and individuals. The `category` and `value` attributes of the OntologyEntry element — see Figure 3 — are interpreted as a DAML class name and the name of an individual value of this class in the MGED Ontology. The ontology handler returns three kinds of quality indicator values:

- `VAL_OK` indicating that the category corresponds to an existing class, and the value is actually defined as an individual of that class in the MGED Ontology;
- `VAL_BAD_IND` indicating that the category corresponds to an existing class, but the value is

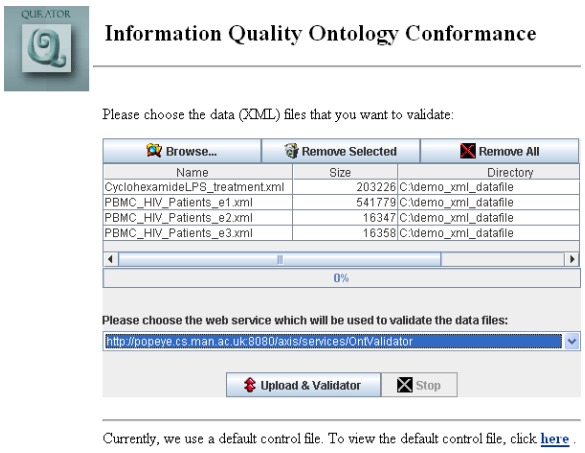

Figure 6: OntValidator upload page screenshot

not defined as an individual of that class in the MGED Ontology;

- `VAL_BAD_CLASS` indicating that the category does not correspond to an existing class in the MGED Ontology (making the value irrelevant).

Other forms of ontology and controlled vocabulary in common use can be checked by the OntValidator service, given alternative handlers, including simple textual/lexical lists of terms, RDF Schemas, and the various sub-languages of OWL. Slightly different notions of conformance apply in each case, of course, but these are abstracted to the more general notion of conformance used at the interface between the metric and preferences layers.

One aim of the Qurator project is to embed the IQ management tools within the scientists' working environment. To this end, we have created a general-purpose Web-based client interface to the OntValidator Web service. Through this interface, users specify the experiment data files to be validated and the control files, and in turn, the validation results are displayed. Figure 6 shows the upload page.

Furthermore, this interface provides an implementation of a simple preference facility, which enables users to specify their own preference rules over the datasets, and calculates metrics to rank checked datasets according to the preferences. This facility corresponds to the Preference Layer of the IQ framework proposed in section 2. Figure 7 illustrates a validation result for four microarray experiment datasets, where the experiment datasets are rated according to a user IQ preference that more than 70% of ontology entries must be "`VAL_OK`" and less than 25% are permitted to be "`VAL_BAD_IND`". Using the form in the lower part of the screen, users may modify the IQ preferences and re-calculate the metrics.

Figure 8 shows a fragment of the detailed validation information for the `VAL_BAD_CLASS` cases in microarray experiment dataset `CyclohexamideLPS_treatment.xml`. The XPaths to

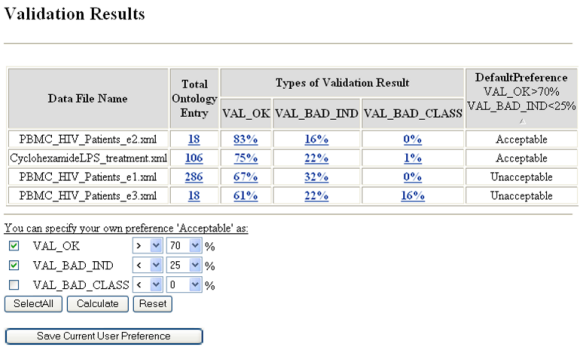

Figure 7: An example of experiment data ranking

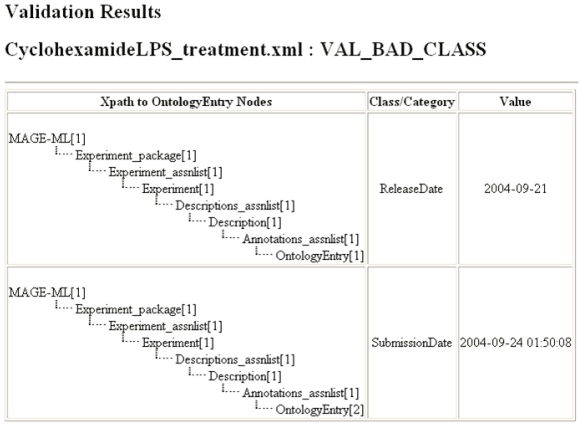

Figure 8: An Example of Detailed Validation Result

the "offending" `OntologyEntry` elements are shown in each case. (These results are due to the fact that the classes `ReleaseDate` and `SubmissionDate` are not defined in the DAML version of the MGED Ontology.)

In the present version of the Web client, the preference rules are captured in a slight variant of RuleML[7] to make them syntactically portable, and mappable to SWRL or other Web rule formalisms in the future. For example, the user preference which defines an "acceptable" microarray experiment data as one in which more than 70% of ontologyEntry elements conform to the MGED Ontology is presented in Figure 9.

As part of our objective to make it easy for user-scientists to adopt the Qurator approach and services, we aim to embed the IQ framework within software tools already familiar to the users. The Pedro data entry tool[8] has been designed for capturing and annotating genomic data for storage or dissemination using XML and has been widely used by biologists, so we are currently working to create an interface to the OntValidator Web service as a plugin to this tool.

[7] http://www.ruleml.org
[8] http://sourceforge.net/projects/pedro

```
<preference>
  <preference_id>1</preference_id>
  <username>qurator</username>
  <rule>
    <if>
      <and>
        <atom>
          <opr><rel>gt</rel></opr>
          <var>VAL_OK</var>
          <ind>70</ind>
        </atom>
        <atom>
          <opr><rel>lt</rel></opr>
          <var>VAL_BAD_IND</var>
          <ind>25</ind>
        </atom>
      </and>
    </if>
    <then>
      <atom>
        <opr><rel>is</rel></opr>
        <var>IQOC</var>
        <ind>Acceptable</ind>
      </atom>
    </then>
    <else>
      <atom>
        <opr><rel>is</rel></opr>
        <var>IQOC</var>
        <ind>Unacceptable</ind>
      </atom>
    </else>
  </rule>
</preference>
```

Figure 9: An example user preference using RuleML

4 Related work: data quality and provenance

One area of quality that has received significant attention within e-Science is the quality of services that are composed into in-silico experiments. As Grids make an increasing number and variety of services available to scientists, the problem of choosing appropriate quality services grows. This has led to research on annotating services to aid their retrieval, e.g. the work on the ^{my}Grid service ontology [11], and research on appropriate service registries for e-Science [8]. When composing services there is the particular problem of services that are compatible at the semantic level, but require special glue because of their incompatible input and output formats. Beyond this most research assumes that a few generic Quality of Service (QoS) criteria, e.g. cost, reliability, availability and response time, will be adequate.

The increasing number and variety of services is not the only quality problem facing e-Science. There is also a deluge of potentially relevant data, and uncertainty over how much of this data should be curated and preserved [7]. This a particular problem in biosciences where the development of high-throughput experimental, and the use of the web for publishing, has lead to an explosion in both the amount and type of data available. One strategy for coping with this is the use of semantic web techniques, making the data and particularly metadata machine manipulable, to filter data according to the requirements of scientists. One ultimate goal is for every scientist to be at the centre of a semantic web of knowledge [5], where rich semantic annotation makes innovative browsing and querying possible. One source of metadata is to record the provenance of scientific data [2, 4, 9, 12]. This can include both the originating context of the data, when an experiment was performed, by whom, using which resources etc., and the processes used to derive results from original data.

In the fields of art and antiques, provenance is used in terms of being able to prove both the origin of an object, and that a seller really has the right to sell the object to a potential purchaser. In e-Science, provenance data is recorded to allow other scientists, or the original scientists at a later date, to fully understand how, and why, a specific result was obtained (and if necessary to prove this). The originator provides the additional metadata and this helps the future user interpret the quality of the data in their context. When results of some experiments are used as inputs of others, one use of data derivation provenance records is to orchestrate the automatic re-running of in-silico experiments as new results emerge.

The use of more generic, and less experiment specific, quality indicators and quality preferences addresses the related problem of finding and filtering from a set of potential data items. The scientist is interested in which items from a set of experimental results, which may have been produced by quite different methods, have their required quality characteristics.

5 Conclusion

We have described the foundation concepts underlying our domain-specific approach to the management of information quality (IQ) in an e-Science context. Rather than forcing users to describe their requirements in terms of a fixed collection of generic (and therefore imprecise) quality concepts, we aim to provide an environment in which individual users can specify their own IQ requirements, in terms of the kinds of information and analysis tool that are available within their domain. This is in line with many standard definitions of quality, which relate to fitness for use for a particular purpose rather than adherence to some absolute quality standard. If we take the notion of fitness for use seriously in relation to IQ in e-Science, then it becomes clear that responsibility for managing high IQ cannot be wholly the duty of information providers and curators. In fact, there is some evidence from our early experiences in Qurator that the most significant contribution information providers can make to the IQ problem is to provide the raw information needed to support a wide-range of domain-specific quality terms, rather than focussing on expensive and time-consuming data cleaning efforts.

In designing the quality ontology and the Qurator framework, it has been necessary to balance a number of competing considerations. For example, ideally, we would have fully declarative definitions of each kind of quality term, but in practice this is not possible. Instead, therefore, we are aiming for a compromise position in which some aspects of the quality terms are described declaratively within the ontology, while others are implemented procedurally as Web services (and workflows) that are bound to the declarative terms in the ontology, in a way that allows two-way navigation between them. A further issue is the tension between the need to allow highly domain-specific quality metrics and indicators to be defined, so that the users' IQ requirements are captured precisely, and the desire to have quality terms that are general enough to be shared across a wide range of users. Our solution to this issue is to create a hierarchical model of quality, in which each level defines examples of the terms in the preceding level in more detail. For example, we might have a generic concept of *Consistency*, which is then specialised as *Ontological Consistency*, then as the various forms of *MGED Consistency*. Finally, at run-time, the complete (i.e. evaluable) version of the quality metric is created, when the user provides values for the remaining unbound parameters. Commonalities to support sharing can then be found at the higher levels, while precision and operational completeness are given by the lowest levels.

In the next stages of the project, we are considering how to embed the software components implied by the Qurator framework into existing data browsing and loading tools, such as maxd[9] and the aforementioned Pedro data entry tool. We also need to expand the set of indicators that we have available, so that they include a variety of indicators from both our target domains. Examples elicited so far include IQ assessment relative to publication quality, to the reputation of the lab which produces the data, and to the statistical properties of the data set [6]. We also plan to explore the different uses that can be made of the navigational links between computed quality annotations and the quality ontology itself.

References

[1] F. Baader, I. Horrocks, and U. Sattler. Description logics. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 3–28. Springer-Verlag, 2004.

[2] P. Buneman, S. Khanna, and W.C̃. Tan. Why and where: A characterization of data provenance. In *International Conference on Database Theory (ICDT)*, 2001.

[3] L. English. *Improving Data Warehouse and Business Information Quality*. Wiley, 1999.

[4] P. Groth, M. Luck, and L. Moreau. Formalising a protocol for recording provenance in Grids. In *Proc 3th UK e-Science All Hands Meeting*, pages 147–154, 2004.

[5] J. Hendler. Communication: Enhanced science and the Semantic Web. *Science*, 299:520–521, 2003.

[6] D. C. Hoyle, M. Rattray, R. Jupp, and A. Brass. Making sense of microarray data distributions. *Bioinformatics*, 18(4):576–584, 2002.

[7] P. Lord, A. Macdonald, L. Lyon, and D. Giaretta. From data deluge to data curation. In *Proc 3th UK e-Science All Hands Meeting*, pages 371–375, 2004.

[8] J. Papay, S. Miles, M. Luck, L. Moreau, and T. Payne. Principles of personalisation of service discovery. In *Proc 3th UK e-Science All Hands Meeting*, pages 139–146, 2004.

[9] S. Rajbhandari and D.W̃. Walker. Support for provenance in a service-based computing grid. In *Proc 3th UK e-Science All Hands Meeting*, pages 524–531, 2004.

[10] R. Wang and D. Strong. Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–34, 1996.

[11] C. Wroe, R. Stevens, C. Goble, A. Roberts, and M. Greenwood. A suite of DAML+OIL ontologies to describe bioinformatics web services and data. *International Journal of Cooperative Information Systems*, 12(2):197–224, 2003.

[12] J. Zhao, C. Wroe, C. Goble, R. Stevens, D. Quan, and M. Greenwood. Using semantic web technologies for representing e-science provenance. In S.Ã. McIlraith, D. Plexousakis, and F. van Harmelen, editors, *3rd International Semantic Web Conference (ISWC2004)*, pages 92–106. Springer-Verlag, 2004.

[9] http://bioinf.man.ac.uk/microarray/maxd/