Managing Information Quality in e-Science using Semantic Web Technology

Alun Preece[†], Binling Jin[†], Edoardo Pignotti[†], Paolo Missier[*], Suzanne Embury[*], David Stead[‡], and Al Brown[‡]

[†]University of Aberdeen, Computing Science, Aberdeen, UK
[*]University of Manchester, School of Computer Science, Manchester, UK
[‡]University of Aberdeen, Molecular and Cell Biology, Aberdeen, UK
info@qurator.org, http://www.qurator.org

Abstract. We outline a framework for managing information quality (IQ) in e-Science, using ontologies, semantic annotation of resources, and data bindings. Scientists define the quality characteristics that are of importance in their particular domain by extending an OWL DL IQ ontology, which classifies and organises these domain-specific quality characteristics within an overall quality management framework. RDF is used to annotate data resources, with reference to IQ indicators defined in the ontology. Data bindings — again defined in RDF — are used to represent mappings between data elements (e.g. defined in XML Schemas) and the IQ ontology. As a practical illustration of our approach, we present a case study from the domain of proteomics.

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. A key element of e-Science is the development of a stable environment for the conduct of information-intensive forms of science. Problems arise due to variations in the quality of the information being shared [3]. Data sets that are incomplete, inconsistent, or inaccurate can still be useful when scientists are aware of these deficiencies.

The Qurator project[1][6] is developing techniques for managing information quality (IQ) using Semantic Web technology. In contrast to previous IQ research, which has tended to focus on the identification of generic, domain-independent quality characteristics (such as accuracy, currency and completeness) [13], we allow scientists to define the quality characteristics that are of importance in their 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.

In order to support this form of domain-specific IQ, we identify three key requirements, each of which can be met using Semantic Web technologies:

[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*.

– Scientists must be able to *use* the domain-specific IQ descriptions, by giving them precise, meaningful definitions, and creating executable metrics based on them. They must also be able to *reuse* definitions created by others, by browsing and querying an organised collection of definitions. To meet this requirement, we propose an extensible *IQ ontology* containing basic domain-independent IQ terms, upon which definitions of domain-specific concepts can be built. By defining the ontology in OWL DL, new descriptors can be classified automatically within the overall IQ framework, allowing user-scientists to locate useful definitions.
– IQ descriptions for specific resources need to be computed and associated with those resources. IQ descriptions of a resource are essentially quality metadata, and can be used to derive higher-order IQ metrics or rankings over sets of resources. As metadata about resources, IQ descriptions can be captured as *semantic annotations* expressed in RDF and related to concepts in the IQ ontology. Annotations are generated by data checking services, sometimes using secondary data sources (e.g. reference datasets), and it is necessary to retain provenance information about how the annotations themselves were derived. This can be done by attaching provenance information to the RDF annotation instances.
– Resources include data and services; both of these kinds of resource are modelled by concepts in the IQ ontology, so that the ontology can express which kinds of IQ descriptor make sense for which kinds of resource. The relationship between actual types of resource (for example a particular data model expressed as an XML Schema, or as a relational database schema) and the abstract models of those resources in the IQ ontology needs to be stated explicitly in order to determine, for a given resource, which checking services are applicable. We refer to these relationships — between the "ontology space" and the "data/service space" — as *bindings*, which can be captured using an RDF schema.

We claim several novel aspects here. To the best of our knowledge, our IQ ontology is the first systematic attempt to capture domain-specific and domain-independent quality descriptors in a semantic model. Moreover, we argue that the use of OWL DL supports the necessary extensibility of the core ontology with domain-specific quality definitions. The annotation and binding RDF schemas are both intended to be generic, reusable components; we were unable to find any previous solution that met our requirements for these.

To provide a concrete illustration of how the elements of our framework can be used in practice, Section 2 introduces a case study in the domain of biology, specifically proteomics. Section 3 gives an overview of the Qurator framework, and the following sections present each of the three components in detail: Section 4 introduces the IQ ontolology, Section 5 describes the binding schema, and Section 6 presents the annotation model. In Section 7 we show how the various components have been implemented within a desktop tool used by biologists to manage their data and metadata.

2 Case Study: Protein Identification

Proteomics is the study of the set of proteins that are expressed under particular conditions within organisms, tissues or cells. Proteins play a vital role in most, if not all, cellular activities — understanding their regulation and function is therefore of fundamental importance to biologists. One experimental approach that is widely used to gain information about the large-scale expression of proteins involves extracting the soluble proteins from a biological sample, then separating them by a technique known as 2-dimensional gel electrophoresis (2DE). This results in a characteristic distribution of protein spots within a rectangular gel (Figure 1). Many hundreds of proteins can be separated from a single sample in this way and the relative amounts of each determined.

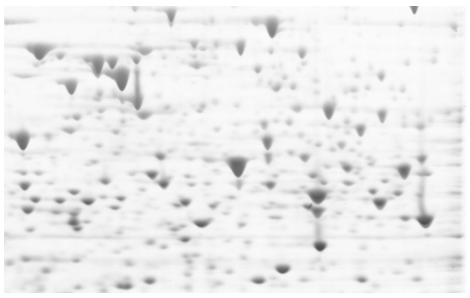


Fig. 1. Sample gel produced by the 2DE technique; the dark areas are protein spots.

The identification of proteins in such experiments is routinely obtained by peptide mass fingerprinting (PMF). In this technique, the protein within the gel spot is first digested with an enzyme that cleaves the protein sequence at certain predictable sites. The fragments of protein that result (called peptides) are extracted and their masses are measured in a mass spectrometer. The experimental list of peptide masses (the “fingerprint”) is then compared against theoretical peptide mass lists, derived by simulating the process of digestion on sequences extracted from a protein database (e.g. NCBInr[2]). Since, for various reasons, it is unlikely that an exact match will be found, the protein identification search engines (e.g. Mascot[3]), that perform this task typically return a list of potential protein matches, ranked in order of search score. Different search engines calculate these scores in different ways, so their results are not directly comparable. It may therefore be difficult for the experimenter and subsequent users of the data to decide whether a particular protein identification is acceptable or not.

There is a debate in progress that seeks to define what information is required when reporting the results of protein identifications by mass spectrometry. For peptide mass fingerprinting, it has been suggested that this should include the number of peptides matched to the identified protein, the number that were not matched in the mass spectrum, and the sequence coverage observed [1].

It would be useful for biologists seeking to interpret the results of proteomic experiments to have a tool that can apply certain quality preferences to a list of protein matches, for the purposes of accepting or questioning a protein identification result.

[2] ftp://ftp.ncbi.nlm.nih.gov/blast/db/blastdb.html
[3] http://www.matrixscience.com/

Such functionality would be particularly useful to scientists wishing to compare protein identification results generated by other labs with those produced within their own. There are two readily accessible indicators that can be used to rank protein identification data and which are independent of the particular search engine used:

– *Hit ratio*: the number of peptide masses matched, divided by the number of peptide masses submitted to the search. This indicator effectively combines the number of matched peptides and the number of unmatched peptides mentioned above. Ideally, most of the peaks in the spectrum should be accountable for by the protein identified, but because of the presence of other components and unpredicted modifications to the matched peptides the hit ratio is unlikely to reach unity.
– *Mass coverage*: the number of amino acids contained within the set of matched peptides, expressed as a fraction of the total number of amino acids making up the sequence of the identified protein and multiplied by the total mass (in kDa) of the protein. Mass coverage is considered superior to the sequence coverage, because peptide mass fingerprints of equal quality give low (percent) sequence coverage for large proteins and high (percent) coverage for small proteins.

These two indicators can be combined in a logical expression that allows us to classify protein matches as acceptable or unacceptable. A software tool could then allow the user-scientist to set threshold values (that is, acceptance criteria) for each metric independently and to see the effect in real time of altering any or all of the threshold values on the acceptability of the data set. This is an example of the kind of quality-aware data analysis that Qurator aims to support.

3 Overview of the Qurator IQ Framework

Before we present the details of the three main components of the Qurator framework — IQ ontology, bindings, annotations — this section gives an overview of how these elements fit together. Figure 2 sets out the key relationships between the various individuals and classes. At the top we have the elements of the IQ ontology, which includes definitions of domain-independent IQ concepts such as Accuracy[4] and also classes of domain-specific indicator such as Hit Ratio and Mass Coverage from the proteomics domain. The IQ ontology also models the various kinds of abstract data entities to which we might wish to apply IQ indicators, such as a Protein Hit obtained from a PMF database search. The ontology then captures the fact that the Hit Ratio indicator applies to a Protein Hit. Finally, the ontology defines the various kinds of data checking function available, as described in detail in Section 4.

At the bottom of Figure 2 we have instances of specific resources (r), for example a particular protein hit derived from a database search. These are often represented in XML; in the proteomics case the PEDRo data model [12] is widely used for this purpose, by means of the PEDRo XML Schema[5].

[4] Throughout this paper, sans-serif font is used for ontology and schema terms from the Qurator framework.
[5] http://pedro.man.ac.uk/files/PEDRoSchema.xsd

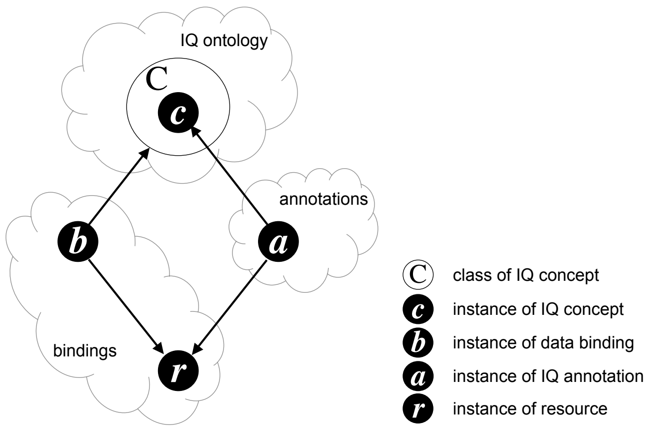


Fig. 2. Overview of the elements of the Qurator IQ framework.

Bindings and annotations both relate resources to elements of the IQ ontology. An instance of a binding (b) relates a resource instance (r) to the corresponding class in the ontology (C); e.g. a specific PEDRo protein hit list to the model class Protein Hit. One of the main uses of bindings is to determine which parts of the IQ conceptualisation are relevant to a particular concrete data model. So, for example, the binding from a PEDRo protein hit structure to the ontology Protein Hit class also lets us identify relevant indicators (such as Hit Ratio) and associated checking functions. For details, see Section 5.

An instance of an annotation (a) relates a specific resource instance (r) to the instance of a quality concept (c). For example, an instance of Hit Ratio with a specific value (e.g. 0.45) might be associated with an individual concrete PEDRo protein hit via an annotation. The IQ instance is said to annotate the associated resource. Further details of annotations are given in Section 6. The difference between bindings and annotations is that the former relate data schema elements (or service types) to the corresponding ontology classes, while the latter relate individual items of data to individual pieces of quality evidence. In other words, bindings define which IQ concepts relate to which kinds of data or service, while annotations associate individual computed IQ descriptors with specific pieces of data.

4 A Semantic Model for Information Quality

As explained in Section 1, a scientist's goal with respect to quality is to determine the suitability of a data set for a given purpose. In our case study, scientists want to assess whether a set of protein identification (PI) experiment results can be safely used as input to a new *in silico* experiment. The scientists' exercise is one of knowledge elicitation: the tacit knowledge regarding quality properties of interest needs to be made explicit and formalized. As we will see, this is also a novel opportunity for scientists to test hypotheses regarding their understanding of quality within a domain. One such hypothesis, described in the next section, is that a small number of measurable quantities associated with the output of protein identification algorithms can be used to discriminate effectively between acceptable and unacceptable matches.

We now present a semantic model that supports such a knowledge elicitation process, by providing a vocabulary and semantic structure for expressing information quality. The model allows scientists to share and reuse their understanding of quality, as well as to perform semi-automated quality assessments on data sets of interest to them.

4.1 Basic Ontology Structure

A number of different existing quality properties can potentially describe suitability, such as *Currency*, *Completeness* or *Accuracy*, definitions of which have been proposed in the existing information quality literature (e.g. [3, 8, 13]). Some of these definitions are given in abstract terms: accuracy for example is defined as the "distance" between a value v and a second value v' that is considered correct, with further distinctions being made based on how the distance is measured [10]. Our model is based on the assumption that scientists should not be concerned with such definitions, and that they should instead be able to state their quality requirements in operational terms, by describing *decision procedures* that determine the suitability of the data.

Nevertheless, our goals of knowledge sharing and reuse mandate the use of a common vocabulary for quality. Our approach is therefore to let users express operational properties of quality in their own terms, while at the same time providing a semantic structure that includes suitable axiomatizations of the definitions found in the literature. We argue, and demonstrate on the practical example presented in this section, that the knowledge representation framework can then be used to establish a relationship between user-provided operational definitions and the axiomatizations.

In practice, let us suppose that our scientists are interested in the "credibility" of published PI experimental results, defined in terms of likelihood of false positives — that is, that a claim of a given protein being present in a sample is false. The biologists involved in this research are proposing decision procedures for computing the likelihood of false positives, based on a small set of measurable quantities, namely the Hit Ratio and the Mass Coverage, which are combined using a logical expression to produce an overall quality score.

In general, the task of defining decision procedures amounts to identifying a collection of measurable indicators, and demonstrating, usually in an experimental way, that they indeed allow a distinction to be made between acceptable and unacceptable data. In some cases, decision models can be semi-automatically generated from sets of examples, with the help of machine learning techniques [14]. In other cases, ad hoc methods have been developed for statistical quality control of experimental data [5, 7].

In the ontology, we model these concepts by introducing Quality Assertions (QA for short); these are decision procedures that are based upon some Quality Evidence (QE), which consists either of measurable attributes called Quality Indicators, or recursively, of functions of those indicators, Quality Metrics. Three main sources of indicators are common in practice:

– *Provenance* metadata, which provides a description of the processes that were involved in producing the data [4, 15].

– *Quality functions* that explicitly measure some quality property, for instance the completeness of a data set relative to a second, reference data set; these functions are typically available from toolkits for data quality assessment with reference to specific issues [2].
– Metadata that is produced as part of the data processing; for example, the Hit Ratio and Mass Coverage indicators are defined as the output of the matching algorithm used for protein identification.

Focusing primarily on the second and third category, we model the indicator-bearing environment as a collection of Data Analysis Tools that may incorporate multiple Data Test Functions, and which are applied to some Data Entity. Indicators are either parameters to or output of these analysis tools. Thus, Hit Ratio and Mass Coverage are part of the output of a test function called PIMatch, used in the PMFMatchAnalysisTool. To continue with our example, a quality metric called PMF Match Ranking associates a "credibility score" to each data in the set, using a function of our two indicators. This score can be used either to classify data as acceptable/non acceptable according to a user-defined threshold, or to rank the data set. Here we will assume that our decision procedure is a classification function called PI-Topk, that provides a simple binary classification of the data set according to the credibility score and to a user-defined threshold.

A QA is applied to collections of data items, which are individuals of the Data Entity class, using the values for the indicators associated to those items. Our example of Data Entity is a protein hit generated by the mass spectrometer, as explained in Section 2, which is used as input to our PMFMatchAnalysisTool.

The following is a summary of the classes and relationships introduced above,using informal notation for the sake of readability; user-defined axioms for the proteomics case study are in bold:[6]

1. Quality-Assertion is based on Quality-Evidence;
2. Quality-Indicator is-a Quality-Evidence;
3. Quality-Metric is-a Quality-Evidence;
4. Quality-Metric is based on Quality-Indicator;
5. Quality-Evidence is output of Data-test-function;
6. Data-analysis-tool is based on Data-test-function;
7. **MassCoverage is-a Quality-Evidence**;
8. **HitRatio is-a Quality-Evidence**;
9. **PIMatch is-a Data-test-function**;
10. **PMFMatchAnalysisTool is-a Data-analysis-tool**;
11. **PMFMatchAnalysisTool is based on PIMatch**;
12. **PIMatch requires input ProteinHit**;
13. **HitRatio is output of PIMatch**;
14. **MassCoverage is output of PIMatch**;
15. **PMF-Match-Ranking is a Quality-Metric**;
16. **PMF-Match-Ranking is based on MassCoverage**;
17. **PMF-Match-Ranking is based on HitRatio**;
18. **PI-Topk is based on PMF-Match-Ranking**.

[6] The full IQ ontology is available from the "Downloads" section at http://www.qurator.org.

4.2 Classification of User-Defined Quality Properties through Reasoning

As mentioned, one goal of this model is to provide a shared collection for top-level, abstract information quality concepts like "accuracy", and to enforce their consistent use. Specifically, we claim that it should be possible to let scientists add only concepts that are familiar to them to the ontology, like those described earlier, while at the same time providing useful entailments that enrich the shared top-level concepts.

In this section, we report on early experiments that support this claim. The main idea is to encourage users to annotate their domain-specific concepts with simple and concrete quality features, to the extent that they are familiar with them, and to use reasoning over OWL DL to entail additional quality properties, or to determine inconsistencies.

Building on the structure described so far, we begin by adding a top-level Quality Property class, with a number of subclasses for Consistency, Timeliness, Currency, and more. Our collection for these concepts currently includes about 20 classes, organized into a three-level hierarchy. Also, we add a root class for Quality Characterization, whose subclasses include Confidence-QC, Reputation-QC, Specificity-QC, and more. These are examples of the "concrete" properties that scientists can more easily associate to specific indicators, or indicator-bearing functions or tools. Thus, we expect users to be able to assert that the PIMatch function has a Confidence-QC, because its purpose, from the quality perspective, is to provide information on the confidence in the experiment result. Note that the ontology model allows a single piece of evidence, or function, to have multiple quality characterizations. The only user assertion for the example is:

PIMatchReport has quality characterization Confidence-QC.

We then introduce OWL DL axioms that describe classes of evidence that have the same quality characterization; given that users may quality-characterize either indicators, metrics, functions, or tools, a sample definition is as follows:

> Confidence evidence includes all and only the quality metrics or indicators whose quality characterization includes Confidence-QC, union all indicators that are output of functions, or of tools that use functions, whose quality characterization includes Confidence-QC.

Here is the OWL DL definition for this class:

$$
\begin{aligned}
&\text{ConfidenceEvidence} \equiv \\
&\quad (\text{QtyMetric} \sqcap (\exists\ \text{metric-based-on-indicator ConfidenceEvidence}) \sqcup \\
&\quad (\text{QtyIndicator} \sqcap \exists\ \text{is-output-of}\ (\exists \text{hasQC ConfidenceQC})) \sqcup \\
&\quad (\text{QtyIndicator} \sqcap \exists\ \text{is-parameter-of}\ (\exists \text{hasQC ConfidenceQC})) \sqcup \\
&\quad (\text{QtyIndicator} \sqcap \exists\ \text{hasQC ConfidenceQC})
\end{aligned}
$$

Using the user-defined assertion above, the definitions in the previous section, and this class definition, an OWL DL reasoner[7] entails the following:

$\text{PIMatchReport} \sqsubseteq \text{ConfidenceEvidence}$,
$\text{HitRatio} \sqsubseteq \text{ConfidenceEvidence}$,
$\text{MassCoverage} \sqsubseteq \text{ConfidenceEvidence}$,

[7] RacerPro has been used for these experiments, http://www.racer-systems.com/

$\mathsf{PMFMatchRanking} \sqsubseteq \mathsf{ConfidenceEvidence}$.

We now define the Accuracy class in terms of the underlying quality characterization, expressing the following:

> Any quality property that is based on a decision procedure that makes use of Confidence or Specificity evidence, can be classfied as Accuracy.

Formally:

$$\begin{aligned}&\mathsf{Accuracy} \equiv \\ &\quad (\exists\ \mathsf{QtyProperty\text{-}from\text{-}QtyPreference}\ (\exists\ \mathsf{pref\text{-}based\text{-}on\text{-}evidence} \\ &\quad (\mathsf{ConfidenceEvidence} \sqcup \mathsf{SpecificityEvidence}))\end{aligned}$$

This last definition allows the ontology to be extended in a consistent way using standard reasoning. Firstly, given a user-defined but yet unclassified quality property, let us call it PI-Acceptability, that is based on the PI-Topk procedure, the reasoner entails that the property is a subclass of Accuracy. Conversely, users may classify PI-Acceptability within the IQ top-level taxonomy; in this case, the reasoner verifies the consistency of this classification.

The experiment shows that it is possible, using suitable DL assertions, to (i) provide axiomatic definitions of traditional quality properties, in terms of an underlying quality characterization vocabulary, and (ii) to use those axioms to propagate, or test the consistency of, user-defined and domain specific quality assertions. As explained in the introduction, the motivation here is to facilitate the use and reuse of definitions in the ontology: consistency checking supports extension of the ontology, and the classification of domain-specific descriptors under generic concepts (such as “accuracy”) is intended to assist users in locating useful concepts.

5 Bindings

As we have shown, the IQ ontology includes semantic models of data resources and the quality analysis services which can be applied to them. The actual data resources have a native definition and presentations; quality test functions applicable on the data might have multiple implementations in different programming languages. For example, a `ProteinHit`[8] XML element, defined in the PEDRo XML schema, may be an input parameter of a HitRatioCalculator function, implemented as a Web service. We designed a generic data model to capture the mapping relationships between data or service resources and their semantic definition. The basic structure of the binding model is presented in Figure 3. There are four core concepts in the Binding model:

Resource refers to any resource that can be located on the Web. We distinguish two sub types of resource: DataResource and ServiceResource. The former refers to any resource which stores information (e.g. an XML file or database table); the latter type represents any service, application or procedure which performs action on a DataResource (e.g. a Web service). We define three categories of DataResource:

[8] Throughout the remainder of this paper, `typewriter` font is used for data elements from XML schemas, and XML syntax fragments.

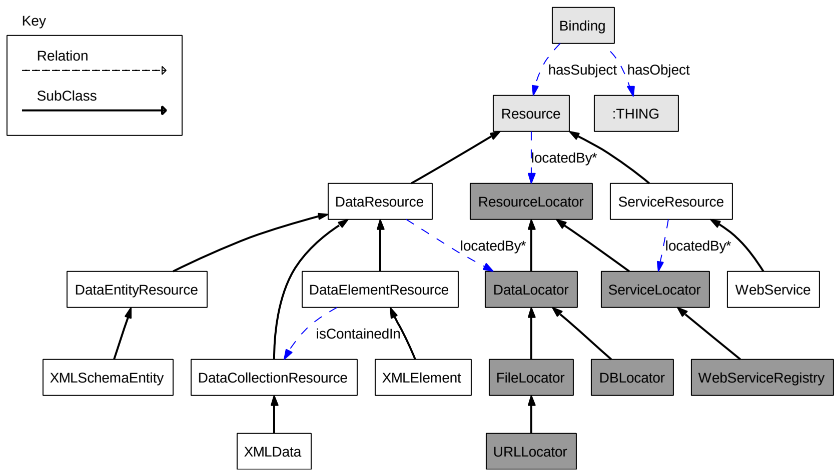


Fig. 3. Overview of the Qurator binding model.

– DataEntityResource represents elements defined in a data schema/structure, for example the `ProteinHit` element defined in the *PEDRo.xsd* schema, or a column defined in a DB table.
– DataElementResource represents a data element inside a collection, for example an XML element specified by an XPath, or a database tuple.
– DataCollectionResource represents a collection of data elements, for example an XML document, a database table, or a text file.

A Binding relates a Resource to a semantic concept in some ontology (for example, in the IQ ontology from the previous section). This relationship is defined by two properties on the binding:

– hasSubject identifies the subject of the binding, which is always a locatable Resource (data or service).
– hasObject identifies the object of the binding, which can be any semantic concept in any ontology (represented in our ontology diagram with the most general concept :THING — for example, this could be any class in our IQ Ontology).

ResourceLocator identifies a global locator for a specific resource. Since the resource is categorised into DataResource and ServiceResource, the ResourceLocator has two types: DataLocator and ServiceLocator. Due to various ways to access the data resources, the data locator can have different types. For example, for a data document, we can use a URL to retrieve it; while for a DB table, a DB connector API could be used (such as JDBC). Similarly, ServiceLocator has different types; for example, the locator of a quality annotation web service can be referred to a WSDL description and the endpoint of the service is presented in this WSDL description.

Figure 4 shows an example binding between a data resource and an IQ ontology class; here, the entity `&q;` refers to the Qurator IQ Ontology and the prefix `b:` identifies terms from the binding model. The data resource `#XMLSchemaEntity1` represents the

```
<b:Binding rdf:about="#binding0">
  <b:hasSubject rdf:resource="#xmlSchemaEntity1"/>
  <b:hasObject  rdf:resource="&q;ProteinHit"/>
</b:Binding>

<b:XMLSchemaEntity rdf:about="#xmlSchemaEntity1">
  <b:locatedBy>
    <b:URLLocator rdf:about="#urlLocator2>
      <b:hasURL>http://example.org/schema/PEDRo.xsd</b:hasURL>
    </b:URLLocator>
  </b:locatedBy>
  <b:hasEntityName>ProteinHit</b:hasEntityName>
</b:XMLSchemaEntity>
```

Fig. 4. An example data binding

entity `ProteinHit` defined in the PEDRo XML Schema, located by a URLLocator instance. The instance `#binding0` represents a binding between the `ProteinHit` data entity and the concept ProteinHit in the IQ Ontology.

Bindings are bi-directional: the binding from resource to concept is used to identify which IQ indicators and associated checking functions are applicable to a particular concrete (e.g. XML) data model; the binding from concept to resource is used to locate concrete data and service implementations (e.g. Web services) to run a data check. Examples of this usage are given in Section 7. Bindings are defined as RDF resources to allow metadata to be associated with the bindings themselves, such as provenance information.

Our binding model was influenced by the XML-to-RDF mappings in WEESA [9]. The key difference, however, is that we are not aiming to map data from the XML "data space" to the RDF "semantic space" or vice versa. In our framework, concrete data and service instances are associated with corresponding ontological concepts by means of the bindings, but there is no translation or transformation of one to the other.

6 Annotation Model

An important aspect of the Qurator approach is to share and reuse quality annotation information on data resources among user-scientists. In order to achieve this we provide a data model which formalises annotation information with semantic support. The structure of our annotation model is shown in Figure 5. The concepts shaded in the figure are defined externally to the annotation model: prefixes b and q identify the binding model and the IQ ontology respectively.

The property hasAnnotation represents the relationship that a b:Resource is annotated with quality information recorded in an AnnotationResult. AnnotationResult defines a class of resource that records the output and related information from one run of some quality annotation service. These annotation results are a group of instances of one particular q:QtyEvidence class; the property referenceTo records the name of the relevant class, and the property hasAnnotationElement records individual annotation result elements, each of which contains an individual q:QtyEvidence instance.

An AnnotationElement relates one individual instance of q:QtyEvidence to one individual annotated resource, using the properties hasQtyEvidence and hasResourceRef

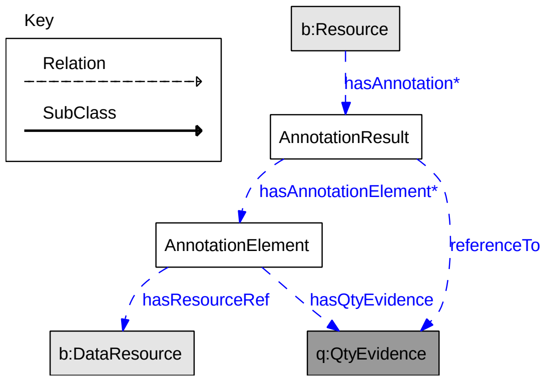


Fig. 5. Structure of Annotation Model

```
<a:AnnotationResult rdf:about="#hitRatio1">
  <a:referenceTo rdf:resource="&q;HitRatio" />
  <a:hasAnnotationElement rdf:resource="#aElement1" />
  <a:hasAnnotationElement rdf:resource="#aElement2" />
  …
</a:AnnotationResult>
```

(a) An AnnotationResult.

```
<a:AnnotationElement rdf:about="#aElement1">
  <a:hasResourceRef rdf:resource="#proteinHit1" />
  <a:hasQtyEvidence>
    <q:HitRatio rdf:about="#qtyEvidence1">
      <q:hasValue> 0.45 </q:hasValue>
    </q:HitRatio>
  </a:hasQtyEvidence>
</a:AnnotationElement>

<b:XMLElement rdf:about="#proteinHit1">
  <b:isContainedIn rdf:about="#xmlData1" />
  <b:locatedBy>
    <b:XMLElementLocator>
      <b:hasXPath>/.../Spot[2]/PeakList[1]/
        DBSearch[1]/ProteinHit[1]</b:hasXPath>
    </b:XMLElementLocator>
  </b:locatedBy>
</b:XMLElement>
```

(b) An AnnotationElement.

```
<b:Binding>
   <b:hasSubject rdf:resource="#proteinHit1" />
   <b:hasObject rdf:resource="&q;ProteinHit" />
</b:Binding>
```

(c) A Binding to an XML element.

Fig. 6. Examples of the annotation model in use.

respectively. It is worth noting that the annotated data elements here are individual `ProteinHit` XML elements, not a protein identification experiment as a whole.

Figure 6 shows an instance of Annotation which refers to the quality evidence class q:HitRatio and has several annotation elements. Figure 6(b) shows one of the AnnotationElements, which indicates that the XML element identified by the XPath to `DBSearch[1]/ProteinHit[1]` is annotated by an instance of q:HitRatio with the value 0.45.

Figure 6(c) shows a b:Binding which binds the annotated protein hit `ProteinHit[1]` to the class ProteinHit in the IQ ontology. Although space prevents us from showing this, the various annotated elements are all part of a b:DataCollectionResource, which in practice would normally be located by a LSID.[9]

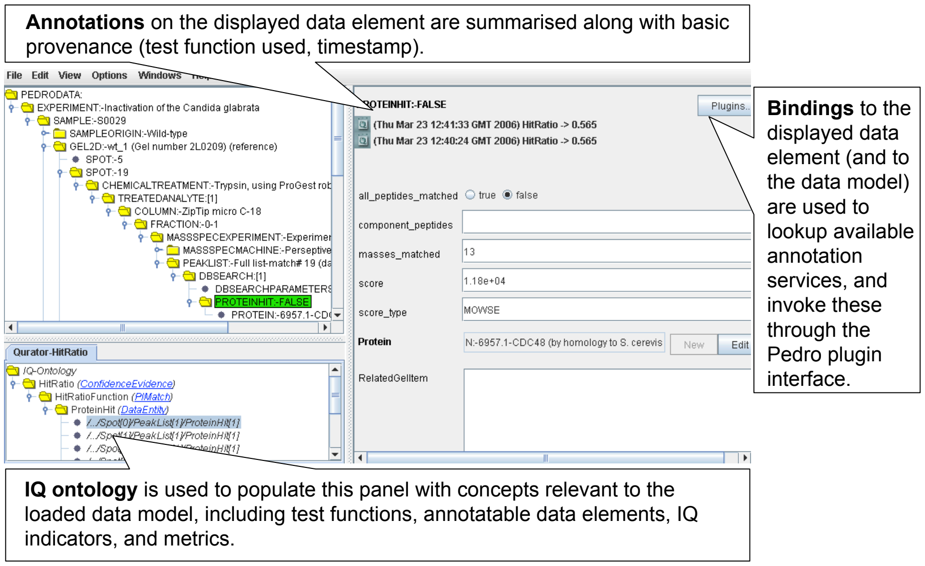
The main difference between our annotation model and that of other frameworks, for example *my*Grid [11], is the way in which annotations are related to both ontology concepts and data resources. The (abstract) conceptual space and the (concrete) data space are kept separate, with annotations — like the bindings in Section 5 — associating elements in the two spaces. The main advantage of this approach is flexibility: an annotation can be easily attached to any kind of resource, and easily associated with any IQ ontology concept. We also support the attachment of provenance information to instances of AnnotationResult, including the identify of the particular checking function used to generate the annotations, and the data selections used as input. Details of this provenance information are omitted for space reasons; however, we are exploring the use of existing provenance architectures for capturing some of these data [4, 15].

7 Protein Identification IQ Service

The Pedro[10] data entry tool is commonly used in proteomics — and several other e-Science domains — to enter and manage XML-based data. To make our approach convenient to user-scientists we have therefore embedded elements of the Qurator framework in the Pedro desktop software. Figure 7 shows a screenshot of the augmented Pedro tool. The top-left area of the screen is the XML document tree and the right-hand panel is the data entry area. When the user starts-up the tool, they are prompted to select the data model on which they will work, for example the PEDRo model for proteomics data. Choice of the data model then drives the content of the top-left and right-hand panels in the standard Pedro environment: users may enter and edit data, and export it to various formats.

Our augmented version of Pedro introduces the lower-left panel, which contains a tree view of the portions of the IQ ontology relevant to the loaded data model. These elements are obtained by querying the ontology dynamically. For the PEDRo data model, they include domain-specific elements such as ProteinHit and HitRatio as well as associated generic concepts such as ConfidenceEvidence. This panel allows users to discover available indicators for the data model at hand, and follow hyperlinks to explore the ontology.

9 http://lsid.sourceforge.net/
10 http://pedrodownload.man.ac.uk/


Annotations on the displayed data element are summarised along with basic provenance (test function used, timestamp).

Bindings to the displayed data element (and to the data model) are used to lookup available annotation services, and invoke these through the Pedro plugin interface.

IQ ontology is used to populate this panel with concepts relevant to the loaded data model, including test functions, annotatable data elements, IQ indicators, and metrics.

```
PEDRODATA:
  EXPERIMENT:-Inactivation of the Candida glabrata
    SAMPLE:-S0029
      SAMPLEORIGIN:-Wild-type
      GEL2D:-wt_1 (Gel number 2L0209) (reference)
        SPOT:-5
        SPOT:-19
          CHEMICALTREATMENT:-Trypsin, using ProGest rob
            TREATEDANALYTE:[1]
              COLUMN:-ZipTip micro C-18
                FRACTION:-0-1
                  MASSSPECEXPERIMENT:-Experimer
                    MASSSPECMACHINE:-Perseptive
                    PEAKLIST:-Full list-match# 19 (da
                      DBSEARCH:[1]
                        DBSEARCHPARAMETERS
                        PROTEINHIT:-FALSE
                          PROTEIN:-6957.1-CD(
```

```
Qurator-HitRatio
IQ-Ontology
  HitRatio (ConfidenceEvidence)
    HitRatioFunction (PIMatch)
      ProteinHit (DataEntity)
        /../Spot[0]/PeakList[1]/ProteinHit[1]
        /../Spot[1]/PeakList[1]/ProteinHit[1]
        /../Spo        /ProteinHit[1]
```

```
ROTEINHIT:-FALSE
(Thu Mar 23 12:41:33 GMT 2006) HitRatio -> 0.565
(Thu Mar 23 12:40:24 GMT 2006) HitRatio -> 0.565

all_peptides_matched   ( ) true  (•) false
component_peptides
masses_matched         13
score                  1.18e+04
score_type             MOWSE
Protein                N:-6957.1-CDC48 (by homology to S. cerevis
RelatedGelItem
```

Fig. 7. Augmented "quality-aware" version of the Pedro data entry tool.

The augmented tool also uses Pedro's plugin model to invoke any available test functions for the model at hand. If the user clicks on the *Plugins* button at the top-right of Figure 7 they are offered two services, to annotate the data with respect to the HitRatio and MassCoverage indicators which are important to biologists (see Section 2). The choice of available service is determined dynamically, using available bindings obtained from an online *binding repository*. Invoking these services results in annotations being added to an online *annotation repository*. (The augmented Pedro desktop tool is configured to act as a client to these two repositories.) By querying the annotation repository, Pedro can retrieve any annotations associated with the displayed data elements shown in the right-hand panel.

It is worth emphasising that the augmented Pedro tool is intended to be a natural and convenient way for user-scientists to access the facilities of the Qurator framework; however, there is nothing in the framework specific to its use in the Pedro tool. In fact, we also have Web interfaces to the various data-checking services, and are developing interfaces that allow them to be invoked as part of e-Science workflows.

8 Conclusion

The Qurator project offers a framework for managing information quality in an e-Science context, allowing user-scientists to specify their IQ requirements against a formal ontology, so that the definitions are machine-manipulable. To the best of our knowledge, this ontology is the first systematic attempt to capture generic and domain-dependent quality descriptors in a semantic model. In this paper, we have shown how the use of OWL DL supports extensibility of the core ontology with domain-specific quality definitions. We have also introduced binding and annotation models that serve to associate concepts in the IQ ontology with data and service entities. Bindings allow IQ-aware tools to identify parts of the IQ ontology relevant to a specific data model. Annotations attach quality metadata to resources. Both the binding and annotation models are to some extent intended to be generic, reusable components.

The Qurator framework has been implemented in a collection of services accessible from a scientist's desktop environment. We are currently gathering feedback from our collaborating users, after which we aim to further develop the IQ framework and associated toolset.

References

1. S. Carr, R. Aebersold, M. Baldwin, A. Burlingame, K. Clauser, and A. Nesvizhskii. Editorial: The need for guidelines in publication of peptide and protein identification data. *Molecular and Cellular Proteomics*, 3:531–533, 2004.
2. M.G. Elfeky, A.K. Elmagarmid, and V.S. Verykios. Tailor: a record linkage tool box. In *Proceedings of the 18th International Conference on Data Engineering (ICDE 2002)*, San Jose, CA, Feb. 2002. IEEE Computer Society.
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. Listgarten and A. Emili. Statistical and computational methods for comparative proteomic profiling using liquid chromatography-tandem mass spectrometry. *Molecular & Cellular Proteomics*, 4(4):419–434, 2005.
6. P. Missier, S. Embury, M. Greenwood, A. Preece, and B. Jin. An ontology-based approach to handling information quality in e-science. In *Proc 4th e-Science All Hands Meeting*, 2005.
7. A.I. Nesvizhskii and R. Aebersold. Analysis, statistical validation and dissemination of large-scale proteomics datasets generated by tandem ms. *Drug Discovery Today*, 9(4):173–181, 2004.
8. T.C. Redman. *Data quality for the information age*. Artech House, 1996.
9. G. Reif, H. Gall, and M. Jazayeri. WEESA - web engineering for semantic web applications. In *Proceedings of the 14th International World Wide Web Conference*, 2005.
10. M. Scannapieco, P. Missier, and C. Batini. Data quality at a glance. *Databanken-Spektrum*, 14:6–14, 2005.
11. N. Sharman, N. Alpdemir, J. Ferris, M. Greenwood, P. Li, and C. Wroe. The myGrid information model. In *Proc 3rd e-Science All Hands Meeting*, 2004.
12. C. F. Taylor et al. A systematic approach to modeling, capturing, and disseminating proteomics experimental data. *Nature Biotechnology*, 21(3):247–254, March 2003.
13. R. Wang and D. Strong. Beyond accuracy: what data quality means to data consumers. *Journal of Management Information Systems*, 12(4):5–34, 1996.
14. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, 2nd Edition*. Morgan Kaufmann, 2005. ISBN 0-12-088407-0.
15. J. Zhao, C. Wroe, C. Goble, R. Stevens, D. Quan, and M. Greenwood. Using semantic web technologies for representing e-science provenance. In *Third International Semantic Web Conference (ISWC2004)*, number 3298 in LNCS, pages 92–106, Hiroshima, Japan, November 2004. Springer-Verlag.