# Pedro Ontology Services: A framework for rapid ontology markup

Kevin Garwood[1], Phillip Lord[1], Helen Parkinson[2], Norman W. Paton[1] and Carole Goble[1]

[1] Department of Computer Science
University of Manchester
Oxford Road
Manchester M13 9PL, UK.
[2] European Bioinformatics Institute
Cambridge M13 9PL, UK.
kevin.garwood@cs.man.ac.uk
http://pedro.man.ac.uk

**Abstract.** Semantic Web technologies offer the possibility of increased accuracy and completeness in search and retrieval operations. In recent years, curators of data resources have begun favouring the use of ontologies over the use of free text entries. Generally this has been done by marking up existing database records with "annotations" that contain ontology term references. Although there are a number of tools available for developing ontologies, there are few generic resources for enabling this annotation process. This paper examines the requirements for such an annotation tool, and describes the design and implementation of the Pedro Ontology Service Framework, which seeks to fulfill these requirements.

## 1 Introduction

The development of many high-throughput technologies has industrialised the production of laboratory data. This has led to increased opportunities for performing biological *in silico* experiments. While some aspects of the data can be characterised by formal data models, significant amounts of biological information are represented as free text or semi-structured data. Experiments are often annotated with free text that describes important aspects such as the experimental techniques used. This annotation enables biologists both to make sense of main data sources and to conduct useful searches.

Traditionally, many different formats and formalisms have been used for database annotation. The most common "formalism" has been–and still is–free text. While this has the advantage of expressivity, it responds to only limited forms of computational searching or comparison. The simplest solution to this difficulty is the use of controlled vocabularies. This approach provides limited expressivity unless the vocabularies are made very large, in which case they cease to be controlled. More recently, there has been great interest in the application

of ontological technologies, particularly since the advent of the Gene Ontology [2], which has been widely adopted.

One of the difficulties of applying ontology technology in this way has been the absence of appropriate tools for generating appropriate ontological annotations. Most of the effort made by the Semantic Web community has focused either on providing annotation in a single formalism, or on providing tools for generating ontologies (e.g. [3, 4]). While these are important areas of development, there is a need for tools which support a variety of data sources that in turn support different annotation formalisms. In this paper, we describe these requirements in more detail and the design and implementation of the Pedro tool, which seeks to fulfill them.

## 2 The Case Studies

The $^{my}$Grid project has developed a service-oriented architecture to enable bioinformaticians to: gather distributed data; use data and analysis tools presented as services; compose and enact workflows; and to manage the generated [9] data. There are now more than a thousand different services available for use, which creates a substantial difficulty in terms of service selection. Therefore, $^{my}$Grid has sought to apply user-oriented semantic service selection to support users in their decisions [7]. The project has required tool support for generating semantic descriptions of the services it provides. An ontology of approximately 500 classes was developed, initially using DAML+OIL and later OWL. It is deployed as an asserted hierarchy represented as RDF(S) [6]. Currently, the $^{my}$Grid ontology and associated descriptions are being managed in a relatively informal manner; there is no formal commitment to the maintenance the ontology as a standard.

The Microarray Gene Expression Data (MGED) Society (`http://www.mged.org`) defines standards for the representation of micro-array information, which describes levels of gene expression within some defined sample. Often complex, these data describe the experimental procedures used to gather the data; information about the source of the sample (e.g. the source species, the anatomical location, cell type etc.) and the experimental context. Consequently, MGED has defined the MGED ontology (MO) [10], which comprises approximately 100 concepts using OWL. Hundreds of thousands of records are expected to be annotated with terms from this ontology. The MO has been crafted using a layered approach: while one layer of core information remains static, other layers of the ontology are allowed to change as knowledge evolves.

The case studies share many common themes that are directly relevant to activities of the Semantic Web. In each case, there is an attempt to model many different kinds of data within the same data set. They try to capture both knowledge about biological systems and knowledge about the context of the experiments. For the microarray case study, the context can include descriptions of laboratory equipment and procedures. In $^{my}$Grid, the context describes aspects of the *in silico* experiment. Both projects have attempted to foster reuse of autonomous data sets by carefully crafting model relationships that link the sets.

Most important, the projects share the common goal of enabling high fidelity retrieval of data.

MGED has the additional aim of supporting data mining activities. It is apparent that the quantity of micro-array data stored will greatly increase, which will require that it be managed at different sites. The efficacy of queries applied across multiple autonomous data sets will critically depend on technologies that allow the structure and content of the experiment records to be clearly defined.

## 3 User Roles and Ontology Life Cycle

One of the key features of knowledge engineering in bioinformatics is the need for community involvement in the development of schemas and ontologies. Probably the best known and most widely used ontology is the Gene Ontology (GO), a Directed Acyclic Graph (DAG) of terms describing the function, biological role and sub-cellular localisation of gene products. This ontology now has approximately 17,000 terms and several million annotated instances. The key reason for its success has not been the adoption of a particular formalism, but its social engagement with its community of users [2]. In this setting, different users play different roles, such as:

**Schema Developer:** Responsible for developing a data model to which data must conform. The schema developers may be working in the context of a Standards Committee, such as MGED, which seeks to ensure that the model supports the requirements of a wide user community.

**Knowledge Engineer:** Responsible for the generation and curation of the ontologies that are used within the schema.

**Data Provider:** Responsible for the generation of data sets according to the schema and using the ontologies.

**Data Consumer:** Responsible for making effective and systematic use of the data sets generated.

Although these roles are different, specific individuals within the community may fulfil more than one. These different user communities are involved in a development life cycle depicted in Figure 1. As well as producing data, the data providers are critical in providing feedback to the standards committee regarding ease of use and coverage of their standard. Similarly, the data consumers are heavily involved in the knowledge engineers' work. Most of them are highly specialised and provide the knowledge required to model their sub-domain.

## 4 The Requirements

In this section, we define the key requirements for knowledge acquisition within the specific case studies described in the last section.

**Rapid Modelling:** Bioinformatics is a large and complex domain; modelling even a small part of it has proven to be extremely challenging. Moreover,

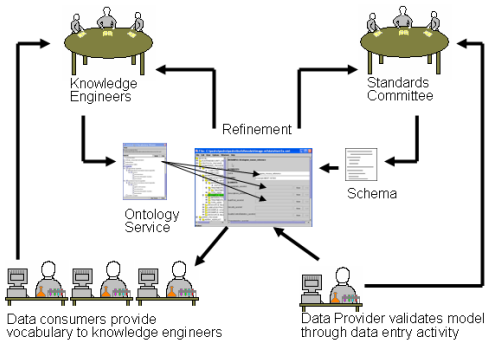



**Fig. 1.** The knowledge enginnering life cycle.

within this domain, a council of perfection is a council of despair: any domain model will be wrong in the initial stages and will need to be evolved iteratively before it approaches correctness. Consequently, there is a strong requirement for a tool that enables collection of data, and which is resilient to change in the schema by which that data is organised.

**Ontological Annotation:** Not all of the data we wish to support are represented ontologically. Some kinds of information, such as probabilistic or numeric data, cannot be well represented using ontologies. Other kinds of information are represented using legacy formalisms. It is unlikely these problems will be fully addressed as the field of bioinformatics evolves. Therefore, the requirement is not to generate instance data according to some ontology, but to use terms from standard ontologies to annotate aspects of the data within some schema.

**Distributed and Autonomous Ontologies:** Although a Standards Committee may control the overall schema, they do not necessarily control the development of the different ontologies in use to populate the fields of the schema. Any tool must be flexible enough to adapt to independent evolution of the ontologies used.

**Multiple Formalisms:** As well as existing legacy data, multiple different formalisms for ontological data are required. In the simplest case, an ontology may be represented as a controlled vocabulary. A more sophisticated form could be a directed acyclic graph, which is the most common representation

within bioinformatics [2]. Other forms include the RDF(S) representation used within projects such as $^{my}$Grid [6] and full property based OWL representations such as those used by the MGED ontology.

**Multiple Ontology Views:** Biologists tend to have strong aversions to filling in forms[3]. This has effected the development of expressive ontologies that describe anatomy. While large anatomy ontologies are available to the bioinformatics community, work has recently begun on small controlled vocabularies aimed at reducing the complexity of form filling [1], while maintaining the link back towards the more expressive ontologies. Therefore, it is clear that as well as supporting multiple underlying ontology formalisms, we need to support multiple different views of these ontologies (and sometimes of the same ontology!) to support the needs of the different user bases within the community.

## 5 The Architecture

This section describes the architecture of the Pedro software, and in particular its ontology framework. The application supports data capture using screens of the form illustrated in Figure 2. The tree view in the left hand panel illustrates the structure of the model, and the data entry form in the right hand panel is being used to capture the properties of a specific *BIOSAMPLE*. The overall architecture of the Pedro software is illustrated in Figure 3.

The principal components of the architecture are described in the following subsections.

### 5.1 Model Manager

The model manager provides access to the data model that is to be viewed or manipulated. XML Schema is used as the primary mechanism for schema representation, although the adaptor interface could in principle be used to provide access to schemata described using different data models. XML is becoming widely used in bioinformatics, and forms the basis of several data standards activities (e.g. [8, 11]).

The Model Manager reads an XML Schema, along with a configuration file that indicates where special behaviours are to be associated with parts of the model (e.g. a configuration file entry could be used to indicate that the values for a particular element are to be obtained from a specific ontology). The hierarchical structure of the XML Schema is reflected in the tree view, which is used both to provide an overview of the structure of the data conforming to the model and to identify where data are to be added or modified.

[3] Our experiences suggest the problem is somewhat wider: biologists dislike most methods of knowledge acquisition and that this dislike is shared by people other than biologists!

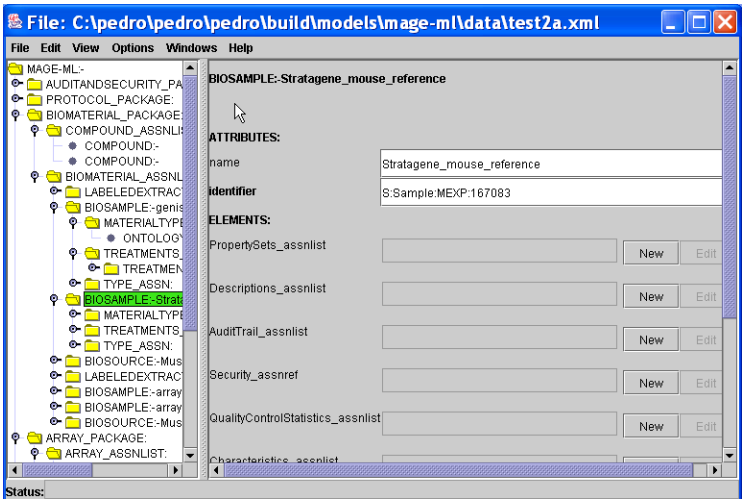



**Fig. 2.** Browsing and manipulating the MAGE-ML microarray data model using the Pedro Data Capture Tool.

### 5.2 Ontology Service

The ontology service provides access to external resources that support the population of XML elements with values drawn from external ontologies. For each kind of external resource, it is necessary: (i) that an adaptor interface has been implemented that provides access to ontologies of the relevant type; and (ii) that a viewer is available that is appropriate for presenting the values from the ontology to annotators. Thus, as illustrated in Figure 4, an element within the model can be associated with both an source location for terms, and an appropriate viewer. Both the *Ontology Source* and the *Ontology Viewer* are defined as Java interfaces, enabling full independent implementation of these components. Through the configuration layer, it is possible to associate one or more ontology services with a given field. This reflects the reality of bioinformatics: that there are often overlapping and non-orthogonal ontologies.

Because of the requirement to support multiple, different formalisms, the Ontology Service interface does not exploit their different levels of expressivity. Currently, Pedro provides a number of different default Ontology Sources that can read from local resources. These include:

**A simple text list:** which provides a straightforward mechanism for the deployment of unstructured controlled vocabularies.

**A tab indented list:** which provides a mechanism for representing controlled vocabularies organised as a tree.

Currently, most of our Ontology Sources use local copies of the distributed ontologies because this suits requirements for shrink-wrapped software. However, with the increasing uptake of programmatic interfaces providing access to On-

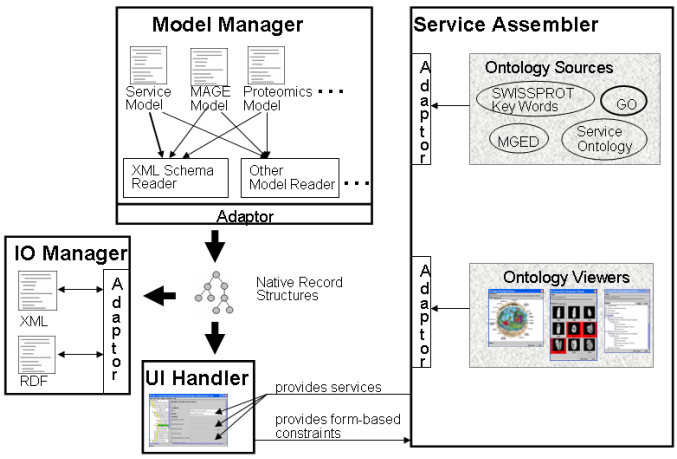



**Fig. 3.** The Pedro architecture.

tological terms [1], we expect that this situation will change, with the majority of ontologies being served remotely.

### 5.3 UI Handler

Following the standard Model-View-Controller design pattern, the *Model Manager* represents the instances defined according to the underlying XML Schema as Java objects, and the *UI Handler* is responsible for rendering this model as a set of Java interface components. While the choice of a model-driven UI fulfills the requirement of a tool that is resilient to change in the underlying models,

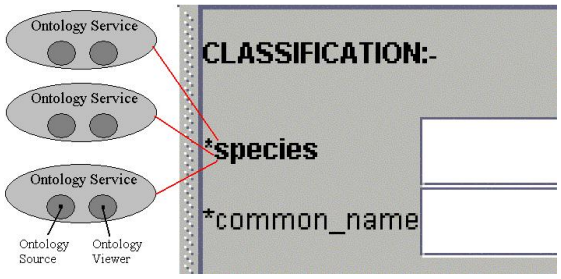



**Fig. 4.** Associating elements with ontology services.

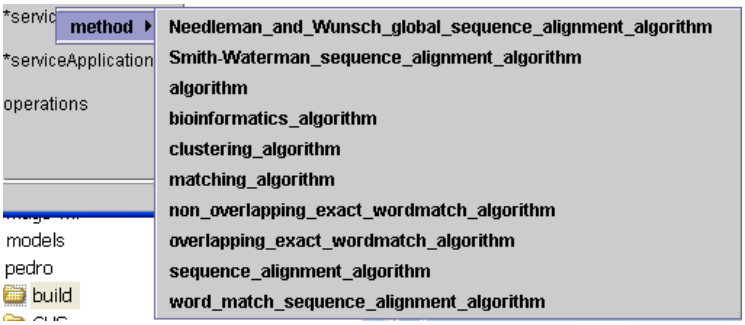



**Fig. 5.** Invoking ontology services through a right-click menu.

we are aware that such a generic approach may produce an interface that is less than ideal for specific users or types of data. In this context, the *Ontology Viewer* offers a key abstraction, separating the concern of term selection from that of serving the ontology. This allows different viewers to be selected based on the user community, or on the size or nature of the ontology.

In many cases, the size of the ontology in use is relatively small, while the number of annotated records is much larger. For this reason, providing convenient "in place" access to ontology terms is the most common mechanism for term selection. Figure 5, which uses an example from $^{my}$Grid, shows the selection of terms describing the functionality of a web service. The rapidity of these interface is of critical importance for data producers who generate large numbers of records. In this figure, we also show the use of "anchoring". The $^{my}$Grid ontology's some 500 classes are too numerous to place in a context menu when only some of these are appropriate for use within the current form field. It is possible to configure Pedro to display only this appropriate subset of classes from the context menu.

While anchoring can reduce the number of concepts to a manageable size, it is not always possible to determine the appropriate subset at design time. For this reason, Pedro supports the notion of "Ontology Contexts"; the values of local fields can be used to restrict the concepts. In Figure 6, the presence of a "laboratory" field is used to restrict later fields to only the members of this laboratory. This Ontology Context functionality is a property of the Ontology Service; Pedro's support for multiple formalisms means that the task of expressing constraints must be devolved to its ontology framework.

When appropriate anchors or contexts cannot be used to restrict the number of concepts on display, Pedro uses a component that displays as a table or a tree as shown in Figure 7. In our experience, viewing as a table is often the most appropriate representation. While the structure of an ontology is intrinsic to its functionality, data providers are often intimately knowledgeable about the terms available and already know which term they wish to use. The table view provides a simple mechanism for rapidly selecting such terms.

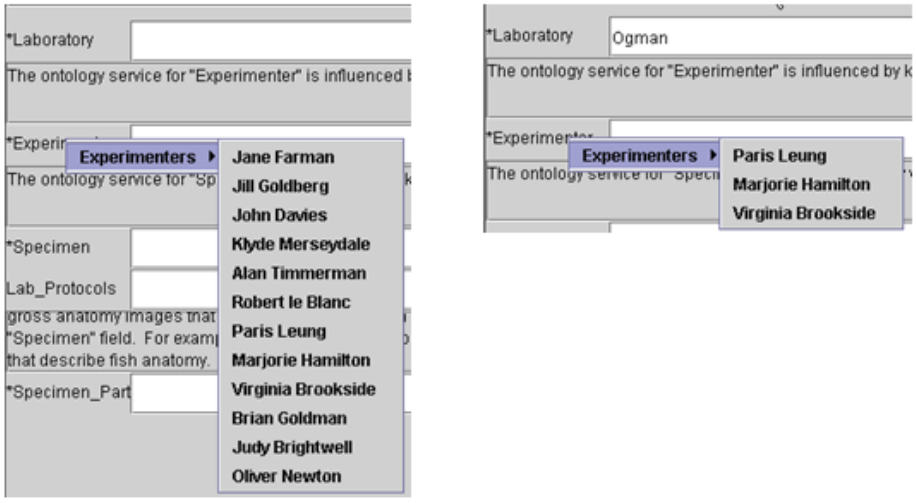



**Fig. 6.** The use of Ontology Context: This figure shows a right-click menu offering terms from a controlled vocabulary.

Pedro also provides "type ahead" incremental search functionality for term selection which, again, enables rapid use. However, not all data providers have the level of intimate knowledge required to use such a facility. Moreover, biological knowledge is often hard to express textually. For this reason, Pedro also provides a selection of image-based Ontology Viewers. In Figure 8(a), we show selection of terms based on an image map, different parts of the image corresponding to different sections of the ontology. This is useful for an ontology that describes concepts which have a spatial relationship to each other. In Figure 8(b), a set of thumbnail images is presented to the data provider, each one representing a specific ontology term.

As well as providing convenient access to ontology terms for some users, the support of images has another advantage: implicit internationalisation. This is a significant problem in biology, where data providers are often geographically distributed, and translation of technical terminology used in both concept names and associated documentation is difficult and expensive. It is also apparent that the use of images, to represent ontological terms, could have significant advantages for generating queries over data.

Currently, the ontology views within Pedro are limited to the selection of named concepts, rather than general class expressions. We would like to investigate adding more capabilities to Pedro, as widget sets for the easy generation of such expressions become available [5]. However, currently the use of reasoning technology in the downstream applications of $^{my}$Grid and MGED is limited. Therefore, the use of these expressions would require significant changes to the architecture of these applications.

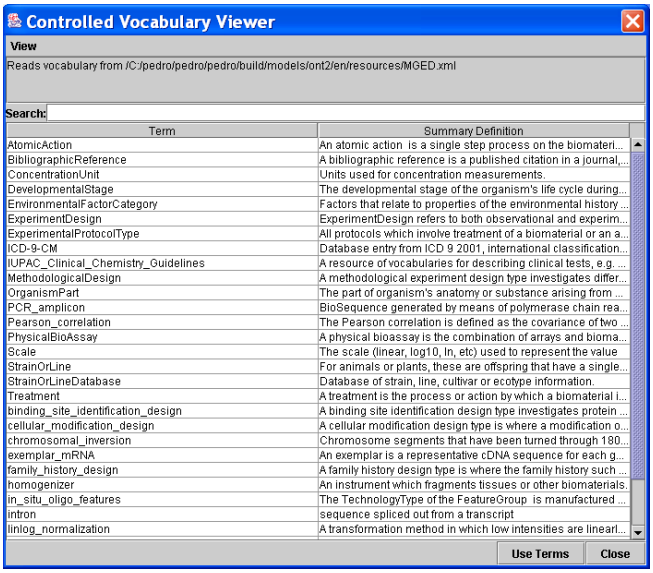



**Fig. 7.** An ontology viewer that visualises terms in a table with term and definition columns.

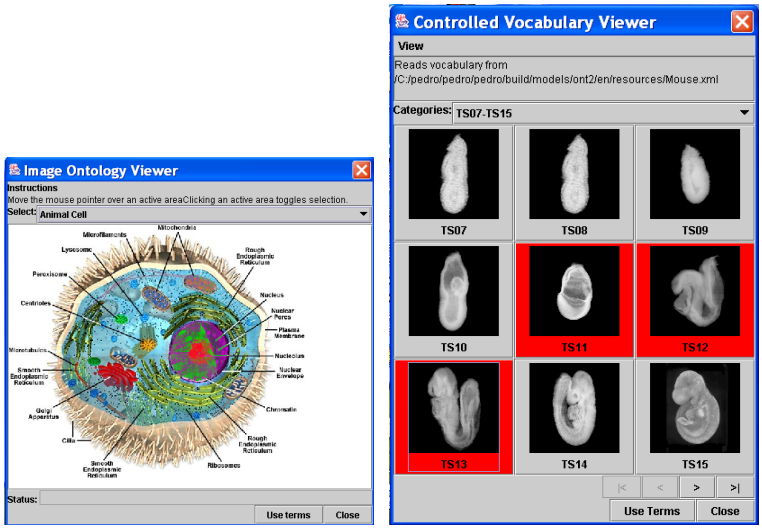
### 5.4 IO Manager

The IO Manager supports the reading and writing of part or all of a data set from or to an external resource. Again, this functionality is pluggable; Pedro comes with IO managers for XML file formats and for an internal representation. However, additional IO manager components have been written that, for example: (i) read data from a relational database into specific elements in a model; and (ii) use an XML database as the source or destination of data that is to be updated using Pedro.

## 6 Meeting the Requirements

Section 4 described various requirements for data annotation tools in bioinformatics. This section revisits the requirements, indicating the approach taken by Pedro to try to satisfy the requirements and the level of support provided.

**Rapid Modelling:** Although Pedro was designed to support data capture and annotation, and not data modelling, in practice several projects have exploited Pedro as part of an iterative modelling activity. That the Pedro interface is model-driven means that a data capture interface can be created directly from a proposed data model. A schema designer can thus develop a version of a model and validate the model by leading the users through a data capture task using the latest version of the model. This has proved an effective way of detecting both errors of omission and commission. Data




(a) (b)

**Fig. 8.** Selection of terms with images.

models change frequently in bioinformatics, reflecting an evolution both in experimental practice and in understanding. Therefore, it is important that bioinformatics tools can be readily evolved to work with new models. Pedro provides no direct support for the versioning of models or ontologies, but its model-driven architecture means that minimal coding is likely to be required to take account of changes to external models. Where software does need to be changed, these changes are likely to be contained within specific adaptors (e.g. within the IO Manager, where the storage format of an external data resource is modified).

**Ontological Annotation:** Pedro integrates ontological annotation with other aspects of data capture by associating elements in an XML Schema with ontology sources and viewers. As such, in Pedro, values for elements can be obtained: (i) through direct user entry; (ii) by selecting from a list of *enumeration* values within the Schema; (iii) by reading values from tab-delimited files using an interactively tailorable import facility; (iv) by reading values from a database using an *IO Manager* adaptor; or (v) by obtaining a value from an ontology accessed through an *Ontology Source* adaptor and presented using visual representations accessed using an *Ontology Viewer* adaptor. Therefore, annotation using ontologies is one of several pluggable components of the Pedro architecture, whereby annotation using ontologies is seamlessly integrated with other forms of data capture.

**Distributed and Autonomous Ontologies:** While a Standards Committee may control an overall schema, the committee is unlikely to exercise centralised control over the development and use of different ontologies. For example, different ontologies may be applied to different families of organism or environmental settings. Pedro supports the use of different ontologies and ontology languages by using pluggable adaptors to access either bundled lists of terms or programmatic interfaces to external reasoning services.

**Multiple Formalisms:** In a community that decides to develop different ontologies, individual groups may choose to use different languages. Pedro communicates with ontology services via an adaptor interface that reflects what the user is offered. The interface shields the rest of the software from details about how offerings are made(e.g. this may be by looking-up an asserted ontology, or by inferring relationships in a description logic ontology).

**Multiple Ontology Views:** Considering the diversity in user communities and the variablility of size and complexity of ontologies, there is no single best way to present terms to end-users. Therefore, Pedro provides an adaptor interface for ontology viewers that has been used to support a broad range of visualisations (Section 5.3). Several of these representations have been widely used in practice, although no systematic usability evaluations have yet been conducted.

Pedro's extensible, model-based architecture can provide some measure of support for a wide range of requirements. We see the development of tools that integrate ontologies with other aspects of an application as being important to their efficient and effective deployment in challenging domains such as bioinformatics.

## 7 Discussion

Bioinformatics is already a significant adopter of ontological and semantic web technologies because they allow data sets to be indexed and retrieved using expressive domain models. However, the community wants to adopt these technologies in an incremental fashion. Scientists may intially want to add ontological annotation to existing database records, rather than recasting all of their data in an ontological formalism.

The Pedro Ontology Service Framework provides a common point of entry, from within the Pedro data capture tool, which enables users to generate potentially rich and contextualised annotations, using multiple distributed and autonomous ontologies, with multiple different and independent formalisms.

The Pedro Ontology Service Framework allows the Pedro data capture tool to uniformly access multiple, distributed, autonomous ontologies, each having its own formalisms. Using the tool's ontology services, end-users can generate contextualised annotations.

These capabilities have ensured that it has already found significant use in independent projects within the bioinformatics domain. We anticipate that its

ontology-based annotation capabilities will also be of significant interest to other domains.

Most of the efforts of the Semantic Web community have focused on developing tools that demonstrate the application of a specific formalism. Relatively little effort has been spent on making end-user tools. We suggest that our experiences designing for these use cases will help spur the development of more open, adaptable Semantic Web technologies.

In aiming to support a highly iterative style of knowledge capture and engineering, we have also been surprised by some requirements. In most cases, where Pedro offers the end-users terms from an ontology, schema designers have generally also allowed them to enter free text noun phrases. While this defeats the purpose of using controlled vocabularies, it suggests that designers believe there is a possibility users will not find the terms that they are looking for. Moreover this capability provides significant feedback to the knowledge engineers, who often wish to incorporate these terms into later versions of their ontologies.

Significant future work remains for Pedro to fulfill its potential. Currently its weakest area of development is its treatment of versioning and change management. Different ontology communities use different methods for representing updates. This is a severe problem for those performing rolling updates on a daily basis. This lack of common procedures between different groups reflects the lack of clear best practices within the community at large. If these experiences are reflected in the Semantic Web community, it will present a significant barrier to adoption of these technologies. Currently, the Pedro framework provides a rudimentary abstraction over these different methods, devolving the task of change management to the various Ontology Service providers; but we are actively seeking ways to improve this abstraction.

**Availability**: the Pedro software is freely available in open source form from `http://pedro.man.ac.uk/`; at the time of writing Pedro has been downloaded around 1000 times, and is being used in a wide range of application communities.

**Acknowledgements:** this work is supported by the UK e-Science Programme $^{my}$Grid and North-West Regional e-Science Centre grants, and through a BBSRC grant under the Proteomics and Cell Function Initiative.

## References

1. Start Aitken, Richard Baldock, Jonathan Bard, Albert Burger, Duncan Davidson, Terry Hayamizu, Helen Parkinson, Alan Rector, Martin Ringwald, Jeremy Rogers, Cornelius Rosse, and Chris Stoeckert. The SOFG Anatomy Entry List (SAEL): an annotation tool for functional genomics data. *Comparative and Functional Genomics*, 2005. *In Press.*
2. Michael Bada, Robert Stevens, Carole Goble, Yolanda Gil, Michael Ashburner, Judith A. Blake, J. Michael Cherry, Midori Harris, and Suzanna Lewis. A Short Study on the Success of the Gene Ontology. Accepted for publication in the Journal of Web Semantics, 2004.
3. S. Bechhofer, R. Möller, and P. Crowther. The dig description logic interface. In *Description Logics*. CEUR Workshop Proceedings, 2003.

4. H. Knublauch, R.W. Fergerson, N.F. Noy, and M.A. Musen. The protégé owl plugin: An open development environment for semantic web applications. In *3rd International Semantic Web Conference*, pages 229–243, 2004.
5. Holger Knublauch, Mark A. Musen, and Alan L. Rector. Editing description logic ontologies with the Protégé owl plugin. In *International Workshop on Description Logics*, Whistler, BC, Canada, 2004.
6. Phillip Lord, Pinar Alper, Chris Wroe, and Carole Goble. Feta: A light-weight architecture for user oriented semantic service discovery. In *European Semantic Web Conference*. Accepted for Publication, 2005.
7. Phillip Lord, Sean Bechhofer, Mark D. Wilkinson, Gary Schiltz, Damian Gessler, Duncan Hull, Carole Goble, and Lincoln Stein. Applying semantic web services to bioinformatics: Experiences gained, lessons learnt. In *International Semantic Web Conference*, pages 350–364, 2004.
8. P.T. Spellman et al. Design and implementation of microarray gene expression markup language (mage-ml). *Genome Biology*, 3(9):research0046.1–0046.9, 2002.
9. R.D. Stevens, H.J. Tipney, C.J. Wroe, T.M. Oinn, M. Senger, P.W. Lord, C.A. Goble, A. Brass, and M. Tassabehji. Exploring Williams Beuren Syndrome Using $^{my}$Grid. In *Bioinformatics*, volume 20, pages i303–310, 2004. Intelligent Systems for Molecular Biology (ISMB) 2004.
10. C.J. Stoeckert and H. Parkinson. The mged ontology: a framework for describing functional genomics experiments. *Comp. Funct. Genom.*, 4:127–132, 2003.
11. C.F. Taylor et al. A systematic approach to modeling, capturing and disseminating proteomics experimental data. *Nature Biotech.*, 21(3):247–254, 2003.