SADI for GMOD: Semantic Web Services for Model Organism Databases

Ben Vandervalk[13], Michel Dumontier[2], E Luke McCarthy[1], and Mark D Wilkinson[1]

[1] James Hogg Research Centre, Heart + Lung Institute, University of British Columbia
[2] Department of Biology, Carleton University
[3] ben.vvalk@gmail.com

Abstract. Here we describe work-in-progress on the SADI for GMOD project (SADI: Semantic Automated Discovery and Integration; GMOD: Generic Model Organism Database), a distribution of ready-made Web services that will bring additional model organism data onto the Semantic Web. SADI is a lightweight standard for implementing Web services that natively consume and generate RDF, while GMOD is a widely-used toolkit for building model organism databases (e.g. FlyBase, ParameciumDB). The SADI for GMOD services will provide a novel mechanism for analyzing data across GMOD sites, as well as other bioinformatics resources that publish their data using SADI.

Keywords: Semantic Web, Web services, SADI, GMOD, model organism databases, bioinformatics, sequence features

1 Introduction

One of the most pervasive problems in bioinformatics is the integration of data and software across research labs. While the prevailing method of sharing data is through centrally controlled repositories such as GenBank [6], manual curation of submissions imposes a bottleneck on the quantity and types of data that can be integrated. In addition, centralization also places limits on the types of visualization and analysis tools that can readily be used with the data.

One prominent example of a system for integrating distributed biological data is the Distributed Annotation System (DAS) [7]. A DAS server provides access to sequence annotations (also known as sequence features) via a RESTful [8] interface, and returns the annotations in a simple, standardized XML format. Client applications (e.g. genome browsers) that understand the DAS protocol and XML format are able to provide users with a unified view of sequence annotations from multiple sites. Nevertheless, DAS has its limitations. The XML datasets returned by DAS servers cannot be integrated without specialized software, and cannot be readily combined with other types of data (e.g. protein-protein interaction networks). In addition, the majority of bioinformatics analysis tools (e.g. BLAST) do not natively understand DAS, and thus they require specialized conversion scripts in order to process data from DAS servers.

In this paper we describe work-in-progress on SADI for GMOD, a collection of Semantic Web services that implement DAS-like functionality. The goal of SADI for GMOD is to provide a more general solution for federating sequence data that is compatible with the Semantic Web, and which facilitates automated integration with analysis software and other types of bioinformatics data. Toward this goal, we propose a standard model for representing sequence features in RDF/OWL. The services are implemented according to the SADI (Semantic Automated Discovery and Integration) standard, and are targeted toward maintainers of GMOD (Generic Model Organism Database) sites. Additional information about these two projects is provided in the following section.

2 Related Projects

SADI (Semantic Automated Discovery and Integration) SADI [1] is a lightweight standard for the implementation of Semantic Web services. Services adhering to the SADI recommendations natively consume and generate data in RDF form, and can be invoked by issuing an HTTP POST to the service URL with an input RDF document as the payload. One of the principal strengths of SADI is that there are no specialized protocols or messaging formats. The interfaces to each service – that is, the expected structure of the input and output RDF documents – are described by means of a provider-specified input OWL class and output OWL class, respectively. Further details about SADI are given in [1].

GMOD (Generic Model Organism Database) The GMOD project [2] is a popular collection of open source software which facilitates the construction of a model organism database and its associated website. The central component of GMOD is a database schema called Chado [3], which houses a variety of datatypes such as sequences, sequence features, controlled vocabularies, and gene expression data. Scripts are provided for creating and loading a Chado instance as a Postgres database.

3 Services

SADI for GMOD consists of five services which provide fundamental operations for accessing sequence feature data, as shown in Table 1. A sequence feature is an annotated region of a biological sequence (DNA, RNA, or amino acid) such as a gene, an exon, or a protein domain. Related features are accessible through a hierarchy of parent-child relationships, and the GMOD wiki provides a set of recommendations [3] indicating where particular feature types should be located in the hierarchy. For example, the GMOD conventions assert that a gene should be a child feature of a chromosome and that an mRNA transcript should be a child feature of a gene. The relationship connecting the parent and child feature will be either “has part” or “derives into”, depending on whether the features are spatially or temporally related. For instance, the relationship between a chromosome and a gene is “has part”, whereas the relationship between a gene and a transcript is “derives into”.

Table 1. A functional description of the five SADI services implemented by the SADI for GMOD project. The fundamental input/output datatypes are genomic coordinates, feature descriptions, and database identifiers; further details about the representation of these entities is given in the following section.

Service Name	Input	Relationship	Output
get_feature_info	a database identifier	is about	a feature description
get_features_overlapping_region	a set of genomic coordinates	overlaps	a collection of feature descriptions
get_sequence_for_region	a set of genomic coordinates	is represented by	a DNA, RNA, or amino acid sequence
get_child_features	a feature description	has part / derives into	a collection of feature descriptions
get_parent_features	a feature description	is part of / derives from	a collection of feature descriptions

4 Proposal for Modeling Sequence Features in RDF

The implementation of the SADI for GMOD services is relatively straightforward. The main point of interest is how the data is modeled in RDF/OWL. The entities that need to be modeled are feature descriptions, genomic coordinates, and database identifiers, as shown in Table 2.

In Listing 1, we show an example feature description for a tRNA gene in *Drosophila melanogaster*, encoded in TURTLE format. The principal ontology used for the encoding is SIO (Semantic Science Integrated Ontology) [4], which provides a large collection of properties for capturing mereological, temporal, and other types of relationships. In addition, features are typed using terms from the Sequence Ontology [5]. Some readers may initially balk at the apparent complexity and opacity of Listing 1; however, it is important to emphasize that the primary goal of the encoding is to facilitate automatic integration of data, whereas simplicity and human-readability are secondary considerations. There are several data modeling practices that, when understood, should help to clarify Listing 1:

1. **Distinct entities are always modeled as distinct nodes in the graph.** In non-RDF formats (e.g. relational databases), it is easy to conflate related entities. For example, the sequence of a chromosome and the chromosome itself are often thought of as the same entity. However, this is not precisely true; the sequence is an abstract string representation of one of the strands of the chromosome. In order to facilitate accurate and automated processing of the data, it is often helpful to make such distinctions explicit. In Listing 1, the tRNA gene has a ranged sequence position in relation to a sequence that represents the minus strand of a chromosome.

Table 2. The fundamental input/output datatypes of the SADI for GMOD services.

Entity	Components	Example
feature description	• a feature type • a set of genomic coordinates • one or more database identifiers	Lines 11..41 of Listing 1
genomic coordinates	• a start position • an end position • a reference sequence	Lines 17..23 of Listing 1
database identifier	• a identifier type • an identifier string	Lines 14..15 of Listing 1

2. **URIs are frequently opaque.** Ontologies providers (e.g. OBI, GO, SO) assign numeric URIs to classes and relationships in their ontologies for two reasons: i) the URIs can have labels in multiple languages, and ii) the labels can be updated without requiring updates to dependent datasets.
3. **Literals are modeled as typed resources.** It is simplest to represent literals in RDF as plain strings or numbers, with the type of the literal indicated by the XSD datatype (e.g. `xsd:float`). Here, literals are modeled as instances of a particular `rdf:type` (e.g. `range:StartPosition`), with the actual values being specified by the "has value" property (i.e. `SIO_000300`). This approach provides a more flexible typing mechanism and allows additional information such as provenance to be attached to the values.
4. **Database identifiers are modeled as typed string values.** In Listing 1, the feature URI `http://lsrn.org/FLYBASE:FBgn0011935` has an attached identifier with an `rdf:type` of `lsrn:FLYBASE_Identifier` and a value of "FBgn0011935". This may seem redundant, as the URI already acts as a unique identifier for the feature. We have adopted the practice of attaching typed, string-encoded database identifiers to URIs in order to address a common problem on the Semantic Web, namely the tendency of data providers to invent their own URI schemes. For example, the URI for UniProt protein P04637 is alternatively represented on the Semantic Web as `http://purl.uniprot.org/uniprot/P04637` (UniProt and LinkedLifeData), `http://bio2rdf.org/uniprot:P04637` (Bio2RDF and Linked Open Drug Data), and `http://lsrn.org/UniProt:P04637` (SADI). While the existence of multiple URIs for the same entity impedes data integration across sites, data providers often create their own URI schemes so that the URIs will resolve to datasets

or webpages on their own sites. We propose attaching database identifiers to URIs as shown here, so that equivalent URIs can automatically be reconciled across sites, while still allowing the URIs created by each provider to resolve to their own data.

Listing 1. Example RDF encoding for a tRNA gene in *Drosophila melanogaster*.

```
@prefix feature: <http://sadiframework.org/ontologies/GMOD/Feature.owl#> .
@prefix range: <http://sadiframework.org/ontologies/GMOD/RangedSequencePosition.owl#> .
@prefix strand: <http://sadiframework.org/ontologies/GMOD/Strand.owl#> .
@prefix FlyBase: <http://lsrn.org/FLYBASE:> .
@prefix GB: <http://lsrn.org/GB:> .
@prefix lsrn: <http://purl.oclc.org/SADI/LSRN/> .
@prefix sio: <http://semanticscience.org/resource/> .
@prefix so: <http://purl.org/obo/owl/SO#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

FlyBase:FBgn0011935
     a so:SO_0001272; # 'tRNA_gene'
     sio:SIO_000008 # 'has attribute'
        [ a lsrn:FLYBASE_Identifier;
          sio:SIO_000300 'FBgn0011935'^^xsd:string ]; # p = 'has value'
     sio:SIO_000008 # 'has attribute'
        [ a range:RangedSequencePosition;
          range:in_relation_to _:minus_strand;
          sio:SIO_000053 # 'has proper part'
            [ a range:StartPosition; sio:SIO_000300 2077634 ];
          sio:SIO_000053 # 'has proper part'
            [ a range:EndPosition; sio:SIO_000300 2077707 ]
        ] .

GB:AE013599 # chromosome arm '2R'
     a so:SO_0000105; # 'chromosome_arm'
     sio:SIO_000008 # 'has attribute'
        [ a lsrn:GB_Identifier;
          sio:SIO_000300 'AE013599'^^xsd:string ] . # p = 'has value'

_:plus_strand
    a sio:SIO_000030; # o = 'sequence'
    sio:SIO_000210 # 'represents'
        [ a strand:PlusStrand;
          sio:SIO_000093 GB:AE013599 ] . # p = 'is proper part of'

_:minus_strand
    a sio:SIO_000030; # o = 'sequence'
    sio:SIO_000210 # 'represents'
        [ a strand:MinusStrand;
          sio:SIO_000093 GB:AE013599 ] . # p = 'is proper part of'
```

5 Deploying the Services

The SADI for GMOD services are implemented as Perl CGI (Common Gateway Interface) scripts. There will be three main steps to deploy the services at a GMOD site:

1. **Set up a Bio::DB::SeqFeature::Store database.** For performance reasons, the services do not query a Chado database directly, but instead use a `Bio::DB::SeqFeature::Store` database which must be loaded separately

by the GMOD site maintainer. The most common scenario is to load the data from a set of GFF files into a mysql database; `Bio::DB::SeqFeature::Store` provides the `bp_seqfeature_load.pl` script for this purpose.
2. **Unpack the SADI for GMOD tarball in the cgi-bin directory.** The tarball will be unpacked into a `SADI` directory tree which will contain the Perl CGI scripts as well as the required Perl modules.
3. **Add database connection parameters to the SADI for GMOD configuration file.** The configuration file will be located in the `SADI` subdirectory of `cgi-bin`.

6 Conclusion

While the majority of existing biological Web services use XML for data exchange, SADI services use RDF/OWL in order to facilitate automatic integration of data across service providers. As such, the SADI for GMOD services will provide a novel tool for conducting analyses across model organism databases, as well as other biological data sources and tools that are published using SADI.

7 Acknowledgements

Initial development of SADI and SHARE has been funded by a special initiatives award from the Heart and Stroke Foundation of British Columbia and Yukon, with additional funding from Microsoft Research and an operating grant from the Canadian Institutes for Health Research (CIHR). In addition, core laboratory funding has been supplied by the National Sciences and Engineering Research Council of Canada (NSERC). Development of SADI for GMOD, as well as hundreds of other SADI services, has been funded by a grant from Canada's Advanced Research and Innovation Network (CANARIE).

References

1. Wilkinson, M.D., Vandervalk, B.P., McCarthy E.L.: SADI Semantic Web Services - cause you cant always GET what you want! Services Computing Conference (APSCC) 2009, 13-18 (2009)
2. GMOD homepage, http://gmod.org
3. Introduction to Chado, GMOD Wiki, http://gmod.org/wiki/Introduction_to_Chado
4. Semantic Science on Google Code, http://code.google.com/p/semanticscience/
5. Eilbeck, K., Lewis, S.E., Mungall, C.J., et al.: The Sequence Ontology: a tool for the unification of genome annotations. Genome Biology 6:5 (2005)
6. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., et al.: GenBank. Nucleic Acids Research 36, D25-D30 (2008)
7. Dowell, R.D., Jokerst, R.M., Day, A. and et al.: The Distributed Annotation System. BMC Bioinformatics 2:7 (2001)
8. Fielding, R.T.: Architectural styles and the design of network-based software architectures. University of California, Irvine (2000)