Supplementary Material: Annotation of SBML Models Through Rule-Based Semantic Integration

Allyson L. Lister, Phillip Lord, Matthew Pocock, Anil Wipat

June 12, 2009

Abstract

This is the supplementary material for our paper accepted at ISMB 2009's Bio-Ontologies Special Interest Group (SIG). This paper is now accessible from Nature Precedings at `http://precedings.nature.com/documents/3286/version/1` and from the SIG proceedings at `http://www.bio-ontologies.org.uk/download/Bio-Ontologies2009.pdf`. Please contact helpdesk@cisban.ac.uk for more information. This material may also be downloaded as a PDF (See Section 1).

Section 1 provides links to the downloadable files associated with this material. Section 2 is an overview of the rule-based mediation method. Section 3 describes the syntactic ontologyfiles associated with each data source from the submission, while Section 4 describes the core ontology, or core ontology. Section 5 lists, in full, each mapping description between the syntactic ontologies and the core ontology. Section 6 describes the results of running the mapping, and provides a discussion of those results.

Please note that this supplementary material should only be interpreted with the aid of the main submission.

There are a number of files associated with the work presented here and in the main submission. You can download them all in Section 1.

List of Figures

1 An overview of the rule-based mediation. 7
2 Rule-based mediation in the context of Systems Biology Markup Language (SBML) model annotation. 8
3 The syntactic ontology used for BioGRID as created by the Protégé XMLTab plugin using a PSI-MIF XML file. 13
4 The UniProtKB syntactic ontology. As with the other syntactic ontologies, instances represent the data itself, while classes represent the structure. 14
5 The BioPAX ontology, filled with instances from Pathway Commons describing the near neighbours of RAD9. As with the other syntactic ontologies, instances represent the data itself, while classes represent the structure. 15

6 MFO, the already-published OWL file used as the syntactic ontology for SBML. 17
7 The telomere ontology, the core ontology for the use cases. . . . 19
8 The description of the PSI-MIF syntactic ontology, as shown in SWRLTab. Data sources such as BioGRID make use of this representation. Filters on the mappings are displayed in the SWRLTab representation as constraints within the rules themselves. This is not a complete rule set for the entirety of PSI-MIF, merely a complete set of rules for describing the use cases. 21
9 The description of the UniProtKB syntactic ontology, as shown in SWRLTab. Filters on the mappings are displayed in the SWRLTab representation as constraints within the rules themselves. This is not a complete rule set for the entirety of UniProtKB, merely a complete set of rules for describing the use cases. 22
10 The description of BioPAX, as shown in SWRLTab. Data sources such as Pathway Commons make use of this representation. Filters on the mappings are displayed in the SWRLTab representation as constraints within the rules themselves. This is not a complete rule set for the entirety of BioPAX, merely a complete set of rules for describing the use cases. 23
11 The description of MFO, as shown in SWRLTab. Data sources such as BioModels could make use of this representation. Additionally, output of the query response occurs via this syntactic ontology, allowing output formatted in SBML. Filters on the mappings are displayed in the SWRLTab representation as constraints within the rules themselves. This is not a complete rule set for the entirety of MFO, merely a complete set of rules for describing the use cases. 24
12 Semantic Web Rule Language (SWRL) Rule TUO_00001 26
13 SWRL Rule TUO_00002. 26
14 Semantic Query-Enhanced Web Rule Language (SQWRL) Rule TUO_SQWRL_00003. TUO_SQWRL_00002 (not shown here) is a version of TUO_SQWRL_00003, but with UniProt as the database name rather than SGD. 27
15 SQWRL Rule TUO_SQWRL_00001. 28
16 SQWRL Rule TUO_SQWRL_00004. 28

List of Tables

1 Data sources and the types of information they provide with respect to the use cases. Check marks imply complete presence of that information, while daggers mark data types that are not always available from that data source. 11

2 The description of the SBML syntactic ontology, Model Format OWL (MFO). Read across, each row is a single rule which, when taken together, form the complete description of MFO. The first column contains mappings, where the left-hand side is a class or relation from MFO and the right-hand side is its equivalent in telomere ontology. The second column contains any filters on the syntactic ontology to further restrict what instances are allowed into telomere ontology. Data sources such as BioModels make use of this representation. The dots at the end of the table indicate that this is not a complete rule set for the entirety of SBML, merely a complete set of rules for describing the use cases. 24

1 Links and Downloads

This paper is now accessible from Nature Precedings at `http://precedings.nature.com/documents/3286/version/1` and from the Bio-Ontologies SIG proceedings at `http://www.bio-ontologies.org.uk/download/Bio-Ontologies2009.pdf`.

There are a number of files associated with the work presented here and in the main submission. You can download them all with the following link:

`http://www.cisban.ac.uk/RBM/rbm\_downloads.zip`

We recommend using Protégé 3.4. If you wish to run a rules engine rather than just view the rules, you will need to install Jess in Protégé . Jess is not available for Open Source licensing under any GPL license. To license and download Jess, please visit the Jess website at: http://www.jessrules.com

You can download a PDF of this supplementary material here:

`http://www.cisban.ac.uk/RBM/supplementaryListerEtAl.pdf`

2 Overview of Rule-Based Mediation

2.1 Background

There are many available data integration methodologies. Choosing which methodology is best-suited for a particular task can be difficult. Misunderstandings and mistakes in data integration are possible if data sources do not describe their information in a semantically equivalent manner [1]. Research into semantic as well as syntactic data integration is of primary importance in the life sciences, where multiple data formats and types flourish. Data integration can be classified in many ways: syntactic versus semantic, federated versus warehousing, encapsulation versus translation.

Comprehensive reviews of the problems of semantic heterogeneity as well as of the data integration approaches used in the past have been written [2, 3]. A thorough review of ontology mapping is present in [4, Section 9]. Work on ontology mapping as well as semantic data integration in bioinformatics includes mapping GO to UMLS [5], creating databases using RDF with S3DB [6] and OntoFusion [3]. OntoFusion, a recent example of ontology mapping within the biomedical domain, uses a only a set of syntactic ontologies, thus creating a query system that does not have a single core ontology describing the domain of interest. While multiple data sources can be queried via a single syntactic ontology within OntoFusion, the query is run directly on each syntactic ontology without any further semantic processing that a core ontology can provide.

The methodology presented here, rule-based mediation, uses rules to define mappings between data formats, and is a form of semantic data integration using layered ontologies. These rules are expressed using Description Logic (DL). DLs are formalisms for knowledge representation characterised by various levels of expressivity. The more expressive a DL language is, the less tractable it is for reasoning purposes. Therefore, a language must be chosen that has an appropriate ratio of expressivity to tractability. DLs are widely used in the biomedical community via the OWL constrained by Description Logics (OWL-DL) format (`http://www.w3.org/TR/owl-ref/`). Ontologies written in OWL-DL have access to a number of DL languages, and editors such as Protégé (`http://protege.stanford.edu`) can determine which DL language subset a particular ontology is written in. While other ontology languages such as Open Biomedical Ontologies (OBO) [7] are commonly used in the life sciences, reasoners provided for OWL-DL language subsets are more powerful [8].

DL has the ability to represent complex logic constructs such as number restrictions and relation hierarchies. Research domains can be successfully modelled in OBO, but DLs allow more complex reasoning tasks and richer semantics. Modelling, together with the logical inferences available when reasoning over a DL-based ontology, make a DL format such as OWL-DL the best choice for our rule-based mediation strategy. With DL, the implicit knowledge that is present within an ontology — and which is not immediately obvious to a human — can be made explicit through inference and reasoning [9, p. 61].

2.2 Rule-Based Mediation

Figure 1 graphically describes the rule-based mediation methodology. Information from one or more data sources sharing the same data format is loaded

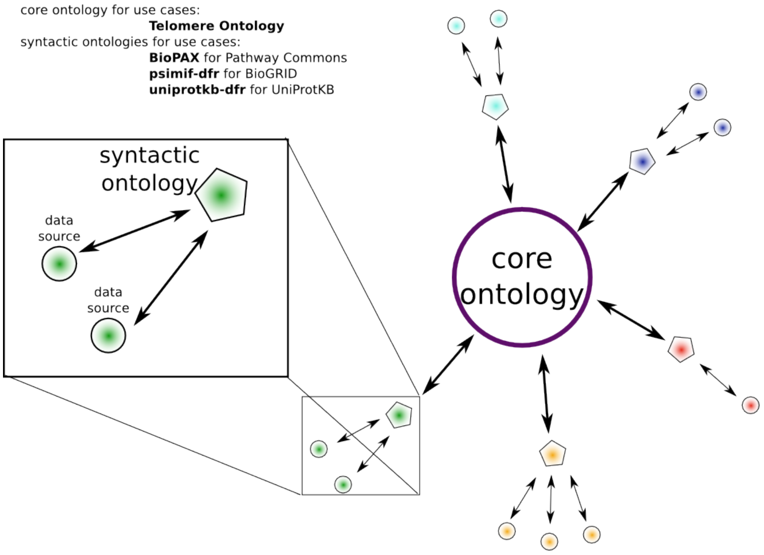

core ontology for use cases:
Telomere Ontology
syntactic ontologies for use cases:
BioPAX for Pathway Commons
psimif-dfr for BioGRID
uniprotkb-dfr for UniProtKB
syntactic ontology
data source
data source
core ontology

Figure 1: An overview of the rule-based mediation.

into that format's syntactic ontology. Each syntactic ontology is linked to the common core ontology using a set of rules called a *description*. The rules are composed of linked mappings and filters on those mappings. Once data has been integrated into the core ontology, it can be queried, and the response formatted according to whichever syntactic ontology the user wishes.

2.2.1 The syntactic ontologies

A syntactic ontology is a syntactic conversion of a non-Web Ontology Language (OWL) format such as XML into OWL-DL.

2.2.2 The core ontology

The core ontology is a tightly-scoped ontology is populated by the data sources via the syntactic ontologies according to the mapping rules.

2.2.3 Ontology Mapping in rule-based mediation

Our rule-based mediation method uses *ontology mapping* to link a source entity or entities to a target entity or entities from different ontologies. In general, ontology mapping does not require either class to be described using the same language. Mapping rewrites the required features of syntactic ontologies as a function of a core ontology, such as in [10]. Querying over those mappings can be performed either over a core ontology directly or over the syntactic ontologies via their core ontology using views.

Mapping data sources to biologically-relevant, ontologically-rigorous core ontologies must be considered carefully. There are two broad mapping strategies: Global As View (GAV) and Local As View (LAV). GAV is when the core ontology is defined as a function of the syntactic ontologies. With LAV, the core ontology is independent of the syntactic ontologies and the syntactic ontologies themselves are described as views of the core ontology. The advantages and disadvantages of both approaches are discussed in [11, 10]. In general, both of these approaches do not materialize data in the core ontology, but retain the data in the syntactic ontologies and use query reformulation to get the data out of the syntactic ontologies.

The rule-based mediation methodology uses a modified version of these approaches. It is similar to the BYU Global-Local as View (BGLaV) approach described by [11], where mappings are generated between syntactic ontologies and the core ontology based on a core ontology which is independent of any of the syntactic ontologies. This approach allows both the straightforward addition of new syntactic ontologies as well as the maintenance of the core ontology as an independent entity. Our rule-based mediation methodology uses an *materialized* BGLaV approach which populates the core ontology with the integrated data from the syntactic ontologies. This allows reasoning and inference to be performed over the integrated data. Detailed information on the descriptions of each data source are available in Section 3.

2.3 Rule-Based Mediation in the Context of Model Annotation

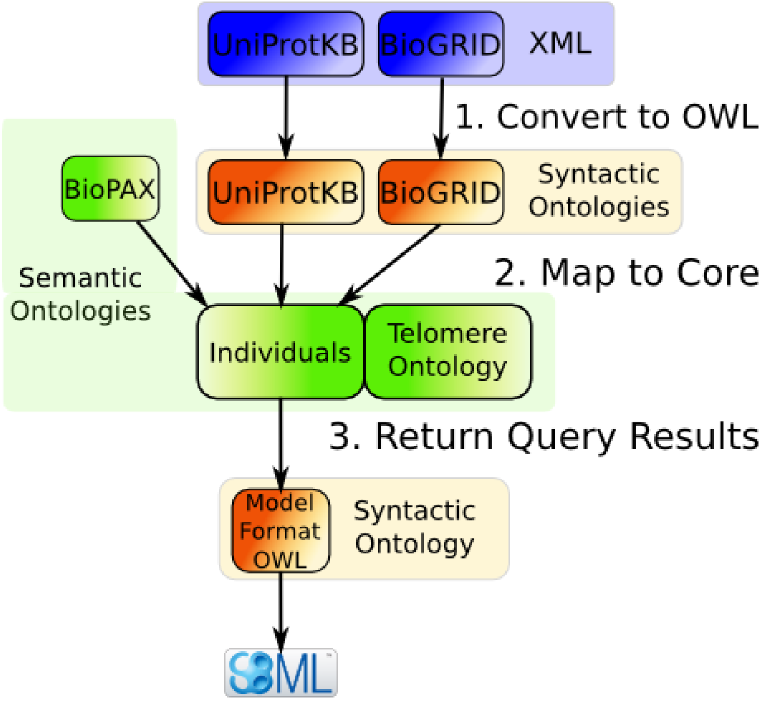


Figure 2: Rule-based mediation in the context of SBML model annotation.

Aids to model annotation exist, but rely extensively on the expert knowledge of the modeller for identification of appropriate additions. SBML Annotation Integration Environment (Saint) (`http://mygrid.ncl.ac.uk/saint`), for example, is a lightweight integration environment wrapped in an easy-to-use web interface. The Saint interface allows modellers to easily upload an initial model, peruse and select appropriate annotation from that suggested by Saint, and then download the newly-annotated model. Similarly, Taverna workflows can be used to pull annotation from data sources to inform models [12]. semanticSBML (`http://sysbio.molgen.mpg.de/semanticsbml/`) focuses only on MIRIAM annotations, and does not add new model elements or any other type of information.

All of these systems have limitations in their understanding of the biology underlying the models themselves. While they resolve a certain amount of syntactic heterogeneity in the data sources, they are unable to process semantic heterogeneity. Semantic heterogeneity can occur when more than one data source uses the same word for a concept while defining that concept differently. For instance, a protein in BioPAX (`http://www.biopax.org`) is strictly defined as having only one polypeptide chain, while a protein in UniProtKB can consist of multiple chains.

The rule-based mediation approach has been implemented as a tool for model annotation via semantic data integration which pulls information for the modeller from data sources in a semantically as well as syntactically integrated way. Such an approach may return information more tailored to their needs and the biology of their models. Figure 2 shows rule-based mediation in the context of SBML model annotation.

3 Syntactic Ontologies

3.1 Notes on the Development of the Syntactic Ontologies

The core of this rule-based mediation strategy for model annotation is the telomere ontology, the core ontology for the Proctor *et al.* model. Each syntactic ontology is separately mapped to this ontology. Just as the syntactic ontologies provide input data to the telomere ontology, they also can provide an output route. This ability gives them the scope to act as a translation system from any syntactic ontology to any other syntactic ontology. It is through this bi-directionality of the information flow that new knowledge can be returned to the originator of a query. Here we present a summary of each of the syntactic ontologies built for these use cases together with a summary of the telomere ontology itself. There are as many syntactic ontologies as there are data formats, with data sources sharing a common format also sharing a syntactic ontology.

The data sources used were BioGRID [13], Pathway Commons (`http://www.pathwaycommons.org`), and UniProtKB [14]. Table 1 provides an overview of the main types of information retrieved from each of the data sources. A ✓ identifies a data type that can always be found from the associated data source, for example downloading a BioGRID interaction file will always include interactions and interaction types. However, some data types are not always available from a given data source. Such partial associations are shown with a †. Table 1 describes the information provided by the data sources in the context of the use cases only. For instance, even though interaction data may be present within UniProtKB entries, as yet no mapping rules have been written and therefore that column is left blank.

Data Source	Interaction	Interaction Type	Entity Identification	Localization
BioGRID	✓	✓	†	
Pathway Commons	✓	†	✓	†
UniProtKB			✓	†
SBML	†	†	†	†

Table 1: Data sources and the types of information they provide with respect to the use cases. Check marks imply complete presence of that information, while daggers mark data types that are not always available from that data source.

The † for BioGRID's entity identification column in Table 1 represents the lack of a UniProtKB identifier for some interactors. Specifically for the use cases, the BioGRID entity representing 'rad9' does not have a cross reference to UniProtKB. Localization information is also not available from the BioGRID input data. For Pathway Commons, all data types are theoretically available as the BioPAX format models them. The actual instance data returned from Pathway Commons does not contain information on either localisation or interaction type. Retrieved UniProtKB information consists of entity localisation and identification, though localisation information is not always present. The SBML syntactic ontology is being used as an output rather than as an input for these use cases, however SBML models may provide any of the described data types.

An existing SBML syntactic ontology, MFO, allows both input of user queries and output of rule-based mediation responses [15]. It is used as an input point for all data sources in SBML format. Syntactic ontologies have been deliberately created as direct translations of non-OWL data formats into OWL. The purpose of a syntactic ontology is to act as a literal, syntactic description of the data source in OWL. As it is the core ontology where the integration and the majority of the inference will occur, it is there that all of the semantic modelling is performed.

Of the four data sources required for the use cases described in the submission, one syntactic ontology had been created by the authors in a previous work, another did not need to be explicitly generated because it was already in OWL-DL, and the other two needed to be written. Those latter two syntactic ontologies were generated using the XMLTab plugin for Protégé 3.4 RC1 (`http://protegewiki.stanford.edu/index.php/XML\_Tab`). This plugin has a number of advantages and disadvantages, but overall it was a good choice for the initial creation of the new syntactic ontologies. After initial generation of the OWL files, changes to the initial OWL files can be made at any time, as needed.

The particular advantages of using XMLTab include:

1. very quick initial creation of each syntactic ontology: each one only took a few seconds to be generated;
2. if an XML file was provided instead of an XSD, then both classes and instances were generated in the syntactic ontology — the classes representing structural elements and attributes present in the XML file, and the instances being created from the actual data contained within the XML file;
3. exact duplication of the XML structure within OWL-DL, which is one of the requirements for each syntactic ontology in the rule-based mediation methodology.

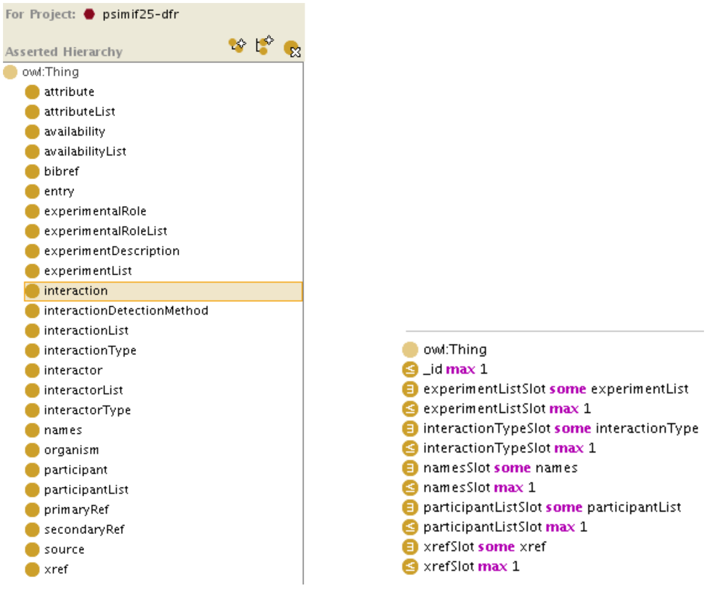
However, XMLTab is not the perfect choice. Some things in particular would be useful in whatever application is used when this work is scaled for larger data integration tasks:

1. *must be able to first load the XSD to generate a complete OWL file with all possible classes, followed by loading multiple XML files to get all necessary instances.* XMLTab only allows the import of one file, either XML or XSD, to generate the OWL file: additional files cannot be applied serially to build up an OWL file based on more than one XML file.
2. it is not clear that it is in active development

The use of existing tools to implement rule-based mediation increases its usability for other researchers as well as decreases the development type. Therefore wherever possible, existing tools were used.

3.2 BioGRID

BioGRID stores 24 different types of interactions, and pairs of interacting entities can be retrieved in PSI-MIF 2.5 format [16]. To fulfil the requirements of Use Case 2, the physical Protein-Protein Interaction (PPI) data were ultimately used, where both participants were proteins. PSI-MIF can easily be used to generate a simple syntactic ontology. The BioGRID syntactic ontology is shown in Figure 3.2. Figure 3a shows an overview of the classes created for the BioGRID syntactic ontology, while Figure 3b shows the conditions applied to the *interaction* class. This syntactic ontology was generated using the XMLTab plugin in Protégé 3.4 RC1 and written in OWL-DL.



For Project: ● psimif25-dfr

Asserted Hierarchy

- owl:Thing
  - attribute
  - attributeList
  - availability
  - availabilityList
  - bibref
  - entry
  - experimentalRole
  - experimentalRoleList
  - experimentDescription
  - experimentList
  - interaction
  - interactionDetectionMethod
  - interactionList
  - interactionType
  - interactor
  - interactorList
  - interactorType
  - names
  - organism
  - participant
  - participantList
  - primaryRef
  - secondaryRef
  - source
  - xref

owl:Thing
≤ _id **max** 1
∃ experimentListSlot **some** experimentList
≤ experimentListSlot **max** 1
∃ interactionTypeSlot **some** interactionType
≤ interactionTypeSlot **max** 1
∃ namesSlot **some** names
≤ namesSlot **max** 1
∃ participantListSlot **some** participantList
≤ participantListSlot **max** 1
∃ xrefSlot **some** xref
≤ xrefSlot **max** 1

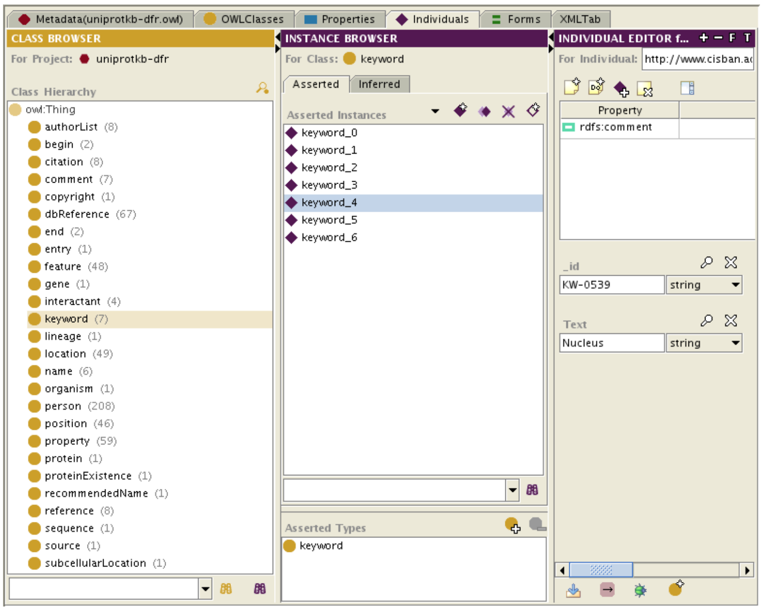
(a) Overview of the PSI-MIF syntactic ontology.

(b) The interaction class.

Figure 3: The syntactic ontology used for BioGRID as created by the Protégé XMLTab plugin using a PSI-MIF XML file.

Pathway Commons and BioGRID store very similar types of data, and yet have a different underlying representation. This exemplifies the challenges facing the biomedical community that can be answered with semantic data integration: very similar data types are being exported in two different data formats.

Very recently (March 2009), Pathway Commons began importing BioGRID data, after this paper was written. This will make it easier in future to retrieve information. Future work will instead focus on adding interaction information from other interaction databases.



For Individual: http://www.cisban.ac

_id: KW-0539 string

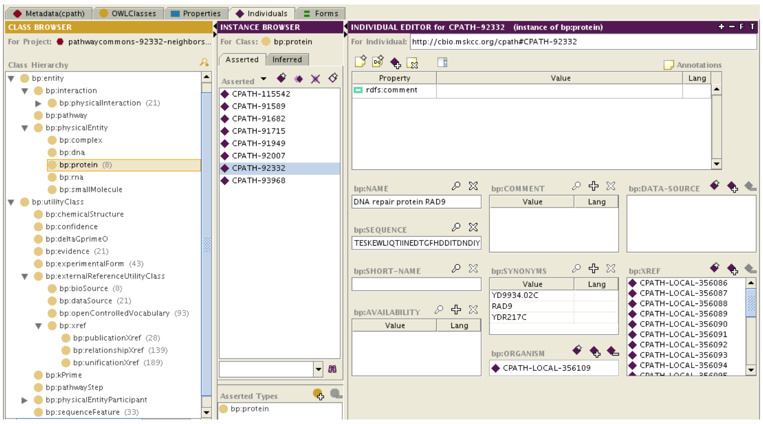
Text: Nucleus string

Figure 4: The UniProtKB syntactic ontology. As with the other syntactic ontologies, instances represent the data itself, while classes represent the structure.

3.3 UniProtKB

The UniProtKB syntactic ontology was generated using Protégé 3.4 RC1 via the XMLTab plugin. The classes represent the element and attribute types, while the instances represent the data themselves as retrieved from UniProtKB. While the original intent was to make use of the UniProtKB RDF format, problems were experienced in reasoning and importing the full set of ontology and RDF files required.

The UniProtKB is a comprehensive public protein sequence and function database, consisting both of manually-curated and automatically-annotated data. Unambiguous assignment of a UniProtKB primary accession to a new species in an SBML document provides a useful way to link disparate instances. For the use cases, UniProtKB was primarily useful for localisation and identification information. Often, new models contain little more than species ids and skeletons of reactions, making even the simple addition of cross-references to UniProtKB useful.



For Project: pathwaycommons-92332-neighbors...

- bp:entity
  - bp:interaction
    - bp:physicalInteraction (21)
  - bp:pathway
  - bp:physicalEntity
    - bp:complex
    - bp:dna
    - bp:protein (8)
    - bp:rna
    - bp:smallMolecule
- bp:utilityClass
  - bp:chemicalStructure
  - bp:confidence
  - bp:deltaGprimeO
  - bp:evidence (21)
  - bp:experimentalForm (43)
  - bp:externalReferenceUtilityClass
    - bp:bioSource (8)
    - bp:dataSource (21)
    - bp:openControlledVocabulary (93)
    - bp:xref
      - bp:publicationXref (28)
      - bp:relationshipXref (139)
      - bp:unificationXref (189)
  - bp:kPrime
  - bp:pathwayStep
  - bp:physicalEntityParticipant
  - bp:sequenceFeature (33)

For Class: bp:protein

- CPATH-115542
- CPATH-91589
- CPATH-91682
- CPATH-91715
- CPATH-91949
- CPATH-92007
- CPATH-92332
- CPATH-93968

Asserted Types: bp:protein

INDIVIDUAL EDITOR for CPATH-92332 (instance of bp:protein)

For Individual: http://cbio.mskcc.org/cpath#CPATH-92332

rdfs:comment

bp:NAME: DNA repair protein RAD9

bp:SEQUENCE: TESKEWLIQTIINEDTGFHDDITDNDIY

bp:SHORT-NAME:

bp:AVAILABILITY:

bp:COMMENT:

bp:SYNONYMS: YD9934.02C, RAD9, YDR217C

bp:ORGANISM: CPATH-LOCAL-356109

bp:DATA-SOURCE:

bp:XREF:
- CPATH-LOCAL-356086
- CPATH-LOCAL-356087
- CPATH-LOCAL-356088
- CPATH-LOCAL-356089
- CPATH-LOCAL-356090
- CPATH-LOCAL-356091
- CPATH-LOCAL-356092
- CPATH-LOCAL-356093
- CPATH-LOCAL-356094

Figure 5: The BioPAX ontology, filled with instances from Pathway Commons describing the near neighbours of RAD9. As with the other syntactic ontologies, instances represent the data itself, while classes represent the structure.

3.4 Pathway Commons

At the time of writing, Pathway Commons integrates seven pathway and interaction databases. Information on network neighbours of any given entity can be accessed as a BioPAX document, written in OWL-DL. As such, no additional syntactic ontology needs to be created: the populated BioPAX ontology returned from nearest neighbour query *is* the syntactic ontology. A portion of the BioPAX ontology is shown in Figure 5. The BioPAX document used in this work is available for download with this supplementary material.

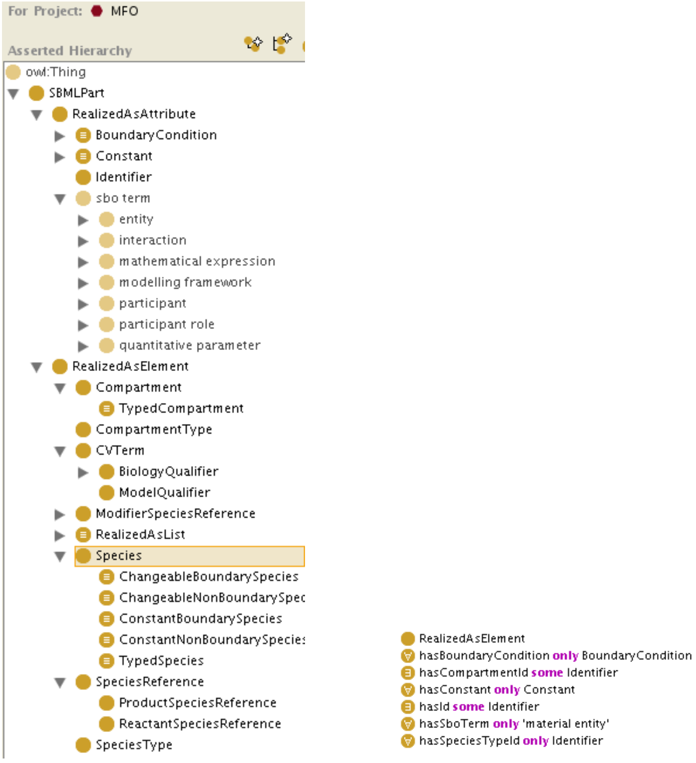
Pathway Commons provides binary interaction data without also providing the direction of the reaction. BioPAX can store information about which species are reactants, products or modifiers, but the data coming from Pathway Commons does not provide this information.

3.5 MFO, the SBML Syntactic Ontology

MFO is a syntactic ontology representing constraints from SBML, Systems Biology Ontology (SBO) and the SBML manual in OWL [15]. Since the original publishing of this research, reasoning and inference time has been dramatically reduced, and more aspects of the three main sources of rules and constraints in SBML have been added. Of particular relevance to this work, classes to describe the annotation elements of SBML have been added. Annotation elements are important to the use cases, as it is in these elements where information such as cross references and taxonomic information reside.

Part of the asserted hierarchy is shown in Figure 6b, while the computer-readable conditions on the $Species$ class, which represents the SBML species element, is shown in Figure 6a. Figure 6c shows a portion of the human-readable comments on the same class. MFO has a dual purpose for the presented use cases: firstly, it acts as the format representation for any data stored in SBML format such as entries from the BioModels database or large-scale networks in SBML format such as [17], and secondly it is used to return the query response in SBML. As this was the only *Saccharomyces cerevisiae (S. cerevisiae)* model for RAD9 currently present in BioModels, no other entries from that database were used as input.

BioModels is an example of a single data source that could be loaded into a core ontology from more than one syntactic ontology. BioModels data are available in both BioPAX and SBML format, and both of these formats have a syntactic ontology. For each situation where more than one syntactic ontology could be used for a data source, a comparison of the richness of the formats is useful. In this instance, both syntactic ontologys need to be created: MFO for, at a minimum, output in SBML format; additionally, the BioPAX format is used for those databases that do not return SBML, such as Pathway Commons. Of the two, BioPAX has more of its constraints and relationships written with logic constructs, while SBML has the majority of its rules not in its XSD, but in the SBML manual and SBO. While more work is planned on a direct comparison of the quality of the input data via the SBML and BioPAX syntactic ontologies, an initial comparison does favour BioPAX.

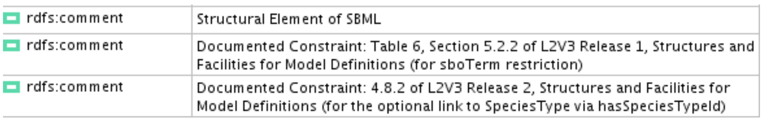


- owl:Thing
- SBMLPart
  - RealizedAsAttribute
    - BoundaryCondition
    - Constant
    - Identifier
    - sbo term
      - entity
      - interaction
      - mathematical expression
      - modelling framework
      - participant
      - participant role
      - quantitative parameter
  - RealizedAsElement
    - Compartment
      - TypedCompartment
    - CompartmentType
    - CVTerm
      - BiologyQualifier
      - ModelQualifier
    - ModifierSpeciesReference
    - RealizedAsList
    - Species
      - ChangeableBoundarySpecies
      - ChangeableNonBoundarySpec
      - ConstantBoundarySpecies
      - ConstantNonBoundarySpecie:
      - TypedSpecies
    - SpeciesReference
      - ProductSpeciesReference
      - ReactantSpeciesReference
    - SpeciesType

RealizedAsElement
hasBoundaryCondition **only** BoundaryCondition
hasCompartmentId **some** Identifier
hasConstant **only** Constant
hasId **some** Identifier
hasSboTerm **only** 'material entity'
hasSpeciesTypeId **only** Identifier

(a) Overview of MFO.

(b) The *Species* class.



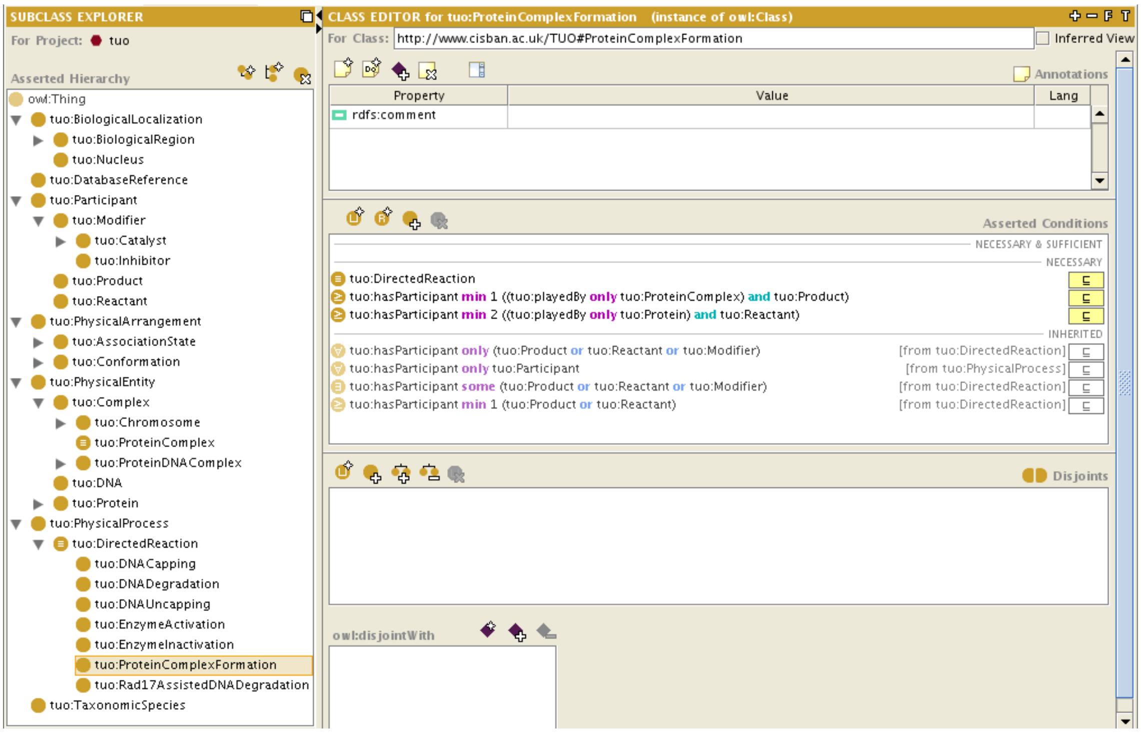
rdfs:comment	Structural Element of SBML
rdfs:comment	Documented Constraint: Table 6, Section 5.2.2 of L2V3 Release 1, Structures and Facilities for Model Definitions (for sboTerm restriction)
rdfs:comment	Documented Constraint: 4.8.2 of L2V3 Release 2, Structures and Facilities for Model Definitions (for the optional link to SpeciesType via hasSpeciesTypeId)

(c) Partial list of comments on *Species*.

Figure 6: MFO, the already-published OWL file used as the syntactic ontology for SBML.

4 Telomere Ontology: The Core Ontology for the Use Cases

Whereas each syntactic ontology is a direct representation of a data source in OWL-DL, an core ontology is a biologically-relevant, logically-rigorous ontology. A core ontology is not intended to capture all of biology; instead, it is scoped tightly to its purpose, which is modelling the research domain of interest. The core ontology used for these use cases is the telomere ontology, and Figure 7 shows a portion of this ontology. While the telomere ontology is not yet complete, the aspects of this ontology necessary for the use cases have been fully constructed.



For Class: http://www.cisban.ac.uk/TUO#ProteinComplexFormation

rdfs:comment

tuo:DirectedReaction
tuo:hasParticipant **min** 1 ((tuo:playedBy **only** tuo:ProteinComplex) **and** tuo:Product)
tuo:hasParticipant **min** 2 ((tuo:playedBy **only** tuo:Protein) **and** tuo:Reactant)

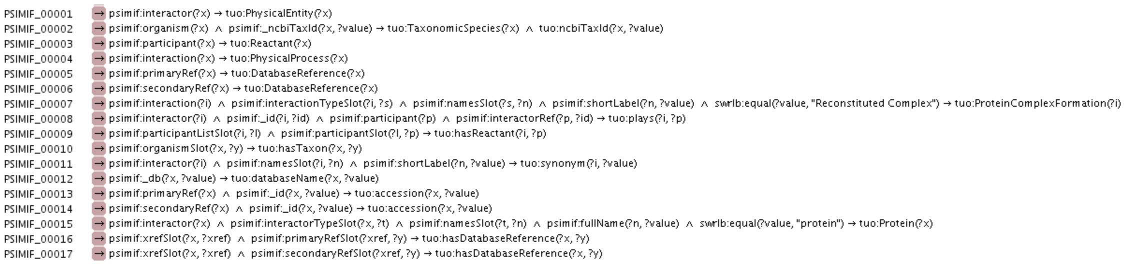
tuo:hasParticipant **only** (tuo:Product **or** tuo:Reactant **or** tuo:Modifier) [from tuo:DirectedReaction]
tuo:hasParticipant **only** tuo:Participant [from tuo:PhysicalProcess]
tuo:hasParticipant **some** (tuo:Product **or** tuo:Reactant **or** tuo:Modifier) [from tuo:DirectedReaction]
tuo:hasParticipant **min** 1 (tuo:Product **or** tuo:Reactant) [from tuo:DirectedReaction]

Figure 7: The telomere ontology, the core ontology for the use cases.

Whereas syntactic ontology are designed to be syntactic representations of the underlying data sources and formats, an core ontology in the rule-based mediation methodology is an explicit description of the semantics of the research domain. Traditionally, mediator-based approaches for information integration have viewed the purpose of an core ontology as a union of syntactic ontologys rather than as a semantically-rich description of the research domain in its own right [18, 19, 10]. If a core ontology is defined as merely an ontology which models a set of data sources, the core ontology becomes brittle with respect to the addition of new data sources and new formats. By creating an core ontology which is more than the entailment of a set of syntactic ontologies, and which stands on its own as a semantically-rich model of a research domain, the core ontology becomes much more flexible with respect to changes.

5 Mapping Rules Linking Syntactic Ontologies to the Telomere Ontology

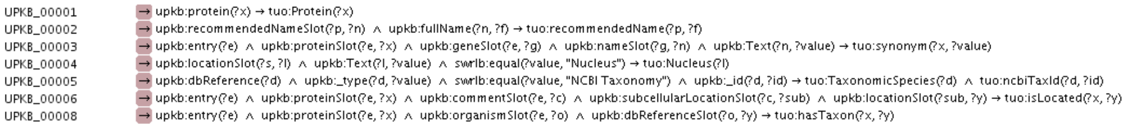
5.1 Rule Set for PSI-MIF to the Telomere Ontology



```
PSIMIF_00001   → psimif:interactor(?x) → tuo:PhysicalEntity(?x)
PSIMIF_00002   → psimif:organism(?x)  ∧  psimif:_ncbiTaxId(?x, ?value) → tuo:TaxonomicSpecies(?x)  ∧  tuo:ncbiTaxId(?x, ?value)
PSIMIF_00003   → psimif:participant(?x) → tuo:Reactant(?x)
PSIMIF_00004   → psimif:interaction(?x) → tuo:PhysicalProcess(?x)
PSIMIF_00005   → psimif:primaryRef(?x) → tuo:DatabaseReference(?x)
PSIMIF_00006   → psimif:secondaryRef(?x) → tuo:DatabaseReference(?x)
PSIMIF_00007   → psimif:interaction(?i)  ∧  psimif:interactionTypeSlot(?i, ?s)  ∧  psimif:namesSlot(?s, ?n)  ∧  psimif:shortLabel(?n, ?value)  ∧  swrlb:equal(?value, "Reconstituted Complex") → tuo:ProteinComplexFormation(?i)
PSIMIF_00008   → psimif:interactor(?i)  ∧  psimif:_id(?i, ?id)  ∧  psimif:participant(?p)  ∧  psimif:interactorRef(?p, ?id) → tuo:plays(?i, ?p)
PSIMIF_00009   → psimif:participantListSlot(?i, ?l)  ∧  psimif:participantSlot(?l, ?p) → tuo:hasReactant(?i, ?p)
PSIMIF_00010   → psimif:organismSlot(?x, ?y) → tuo:hasTaxon(?x, ?y)
PSIMIF_00011   → psimif:interactor(?i)  ∧  psimif:namesSlot(?i, ?n)  ∧  psimif:shortLabel(?n, ?value) → tuo:synonym(?i, ?value)
PSIMIF_00012   → psimif:_db(?x, ?value) → tuo:databaseName(?x, ?value)
PSIMIF_00013   → psimif:primaryRef(?x)  ∧  psimif:_id(?x, ?value) → tuo:accession(?x, ?value)
PSIMIF_00014   → psimif:secondaryRef(?x)  ∧  psimif:_id(?x, ?value) → tuo:accession(?x, ?value)
PSIMIF_00015   → psimif:interactor(?x)  ∧  psimif:interactorTypeSlot(?x, ?t)  ∧  psimif:namesSlot(?t, ?n)  ∧  psimif:fullName(?n, ?value)  ∧  swrlb:equal(?value, "protein") → tuo:Protein(?x)
PSIMIF_00016   → psimif:xrefSlot(?x, ?xref)  ∧  psimif:primaryRefSlot(?xref, ?y) → tuo:hasDatabaseReference(?x, ?y)
PSIMIF_00017   → psimif:xrefSlot(?x, ?xref)  ∧  psimif:secondaryRefSlot(?xref, ?y) → tuo:hasDatabaseReference(?x, ?y)
```

Figure 8: The description of the PSI-MIF syntactic ontology, as shown in SWRLTab. Data sources such as BioGRID make use of this representation. Filters on the mappings are displayed in the SWRLTab representation as constraints within the rules themselves. This is not a complete rule set for the entirety of PSI-MIF, merely a complete set of rules for describing the use cases.

5.2 Rule Set for UniProtKB to the Telomere Ontology

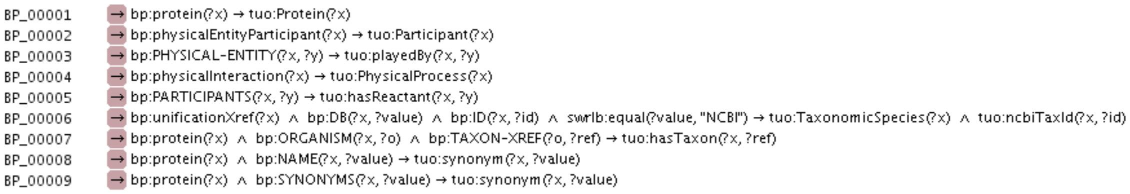


UPKB_00001	→ upkb:protein(?x) → tuo:Protein(?x)
UPKB_00002	→ upkb:recommendedNameSlot(?p, ?n) ∧ upkb:fullName(?n, ?f) → tuo:recommendedName(?p, ?f)
UPKB_00003	→ upkb:entry(?e) ∧ upkb:proteinSlot(?e, ?x) ∧ upkb:geneSlot(?e, ?g) ∧ upkb:nameSlot(?g, ?n) ∧ upkb:Text(?n, ?value) → tuo:synonym(?x, ?value)
UPKB_00004	→ upkb:locationSlot(?s, ?l) ∧ upkb:Text(?l, ?value) ∧ swrlb:equal(?value, "Nucleus") → tuo:Nucleus(?l)
UPKB_00005	→ upkb:dbReference(?d) ∧ upkb:_type(?d, ?value) ∧ swrlb:equal(?value, "NCBI Taxonomy") ∧ upkb:_id(?d, ?id) → tuo:TaxonomicSpecies(?d) ∧ tuo:ncbiTaxId(?d, ?id)
UPKB_00006	→ upkb:entry(?e) ∧ upkb:proteinSlot(?e, ?x) ∧ upkb:commentSlot(?e, ?c) ∧ upkb:subcellularLocationSlot(?c, ?sub) ∧ upkb:locationSlot(?sub, ?y) → tuo:isLocated(?x, ?y)
UPKB_00008	→ upkb:entry(?e) ∧ upkb:proteinSlot(?e, ?x) ∧ upkb:organismSlot(?e, ?o) ∧ upkb:dbReferenceSlot(?o, ?y) → tuo:hasTaxon(?x, ?y)

Figure 9: The description of the UniProtKB syntactic ontology, as shown in SWRLTab. Filters on the mappings are displayed in the SWRLTab representation as constraints within the rules themselves. This is not a complete rule set for the entirety of UniProtKB, merely a complete set of rules for describing the use cases.

The UniProtKB description, shown in Figure 9, is quite different from the other data sources'. While UniProtKB contains a large amount of information, it is protein-centric and not reaction-centric. However, some reactions are presented via comment sections and cross-references to other databases. More importantly for our purposes, rich information about a protein, described in other databases with much less detail, is available to telomere ontology via this data source. Specifically, information regarding protein localisation within the cell as well as synonyms of gene and protein names are available. This description is the simplest of all of the syntactic ontologies': many classes required for the use cases have direct equivalents in telomere ontology, and the cardinalities of many relations are the same. While the UniProtKB does contain some limited information on reactions a protein is involved in via the comment and cross-reference sections, these are not currently modelled by the UniProtKB syntactic ontology.

5.3 Rule Set for BioPAX to the Telomere Ontology



BP_00001	bp:protein(?x) → tuo:Protein(?x)
BP_00002	bp:physicalEntityParticipant(?x) → tuo:Participant(?x)
BP_00003	bp:PHYSICAL-ENTITY(?x, ?y) → tuo:playedBy(?x, ?y)
BP_00004	bp:physicalInteraction(?x) → tuo:PhysicalProcess(?x)
BP_00005	bp:PARTICIPANTS(?x, ?y) → tuo:hasReactant(?x, ?y)
BP_00006	bp:unificationXref(?x) ∧ bp:DB(?x, ?value) ∧ bp:ID(?x, ?id) ∧ swrlb:equal(?value, "NCBI") → tuo:TaxonomicSpecies(?x) ∧ tuo:ncbiTaxId(?x, ?id)
BP_00007	bp:protein(?x) ∧ bp:ORGANISM(?x, ?o) ∧ bp:TAXON-XREF(?o, ?ref) → tuo:hasTaxon(?x, ?ref)
BP_00008	bp:protein(?x) ∧ bp:NAME(?x, ?value) → tuo:synonym(?x, ?value)
BP_00009	bp:protein(?x) ∧ bp:SYNONYMS(?x, ?value) → tuo:synonym(?x, ?value)

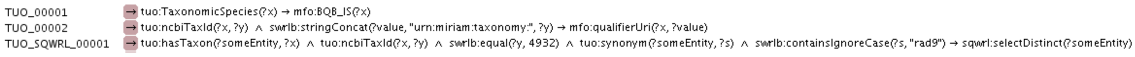
Figure 10: The description of BioPAX, as shown in SWRLTab. Data sources such as Pathway Commons make use of this representation. Filters on the mappings are displayed in the SWRLTab representation as constraints within the rules themselves. This is not a complete rule set for the entirety of BioPAX, merely a complete set of rules for describing the use cases.

$Descr(MFO)$

Mappings (MFO → Telomere Ontology)	Filters
$Species(X) \Rightarrow PhysicalEntity(X)$	
$Annotation(X) \Rightarrow Taxon(X)$	
$SpeciesReference(X) \Rightarrow Participant(X)$	
$Reaction(X) \Rightarrow DirectedReaction(X)$	
$Compartment(X) \Rightarrow BiologicalLocalization(X)$	
$hasNestedSbmlPart(X, Y) \Rightarrow hasParticipant(X, Y)$	$X \sqsubseteq Reaction$
$(^{-}hasNestedSbmlPart \circ^{-} hasNestedSbmlPart \circ hasAnnotation)(X, Y) \Rightarrow hasTaxon(X, Y)$	X $\sqsubseteq$ $Species \sqcap Y \sqsubseteq startsWith(``uri : miriam : taxonomy :'')$
$(hasCompartmentId \circ^{-} hasId)(X, Y) \Rightarrow located(X, Y)$	
$(hasSpeciesId \circ^{-} hasId)(X, Y) \Rightarrow hasPhysicalEntity(X, Y)$	
...	

Table 2: The description of the SBML syntactic ontology, MFO. Read across, each row is a single rule which, when taken together, form the complete description of MFO. The first column contains mappings, where the left-hand side is a class or relation from MFO and the right-hand side is its equivalent in telomere ontology. The second column contains any filters on the syntactic ontology to further restrict what instances are allowed into telomere ontology. Data sources such as BioModels make use of this representation. The dots at the end of the table indicate that this is not a complete rule set for the entirety of SBML, merely a complete set of rules for describing the use cases.

5.4 Rule Set for Telomere Ontology to MFO



```
TUO_00001         → tuo:TaxonomicSpecies(?x) → mfo:BQB_IS(?x)
TUO_00002         → tuo:ncbiTaxId(?x, ?y)  ∧  swrlb:stringConcat(?value, "urn:miriam:taxonomy:", ?y) → mfo:qualifierUri(?x, ?value)
TUO_SQWRL_00001   → tuo:hasTaxon(?someEntity, ?x)  ∧  tuo:ncbiTaxId(?x, ?y)  ∧  swrlb:equal(?y, 4932)  ∧  tuo:synonym(?someEntity, ?s)  ∧  swrlb:containsIgnoreCase(?s, "rad9") → sqwrl:selectDistinct(?someEntity)
```

Figure 11: The description of MFO, as shown in SWRLTab. Data sources such as BioModels could make use of this representation. Additionally, output of the query response occurs via this syntactic ontology, allowing output formatted in SBML. Filters on the mappings are displayed in the SWRLTab representation as constraints within the rules themselves. This is not a complete rule set for the entirety of MFO, merely a complete set of rules for describing the use cases.

Unlike the other syntactic ontologies, MFO is used for output of the query response. While all syntactic ontologiesin rule-based mediation are capable of being used as both inputs and outputs, the use cases presented here specify that in this instance, MFO is used for output. Therefore, the mappings presented in Figure 11 are from telomere ontology to MFO. Please note that all of the rules described in Table 2 have not yet been loaded into the SWRLTab implementation shown in Figure 11. These are imminent, and the image above will change shortly to reflect the implementation of the DL rules.

6 Results

6.1 Results Summary

The results and their implications are summarised in the SIG paper (linked in Section 1). However, here we go into more detail about how these results were produced. You can view the final version of the integrated schema used in this paper by downloading the zip file linked in Section 1 and opening

```
is-container.pprj
```

in Protégé 3.4.

One note on SWRL and SQWRL: while running SWRL rules modifies the target ontology when mapping instances from one ontology to another (or from one location in a single ontology to another), SQWRL queries do not change the query ontology.

Please note that this section is not completely written yet.

6.2 Use Case 1 Results: Annotation

A full mapping of the results from the first use case. This section describes the rules and queries used to generate the results described in Use Case 1.

6.2.1 Discovery of RAD9 Proteins

Other than the rules described for the syntactic ontologies, some telomere ontology-specific rules needed to be created to aid querying.

The first two (TUO_00001 in Figure 12, and TUO_00002 in Figure 13) assign instances containing particular names or synonyms to the Rad9 class.

$tuo : Protein(?someEntity) \wedge$
$tuo : synonym(?someEntity, ?s) \wedge$
$swrlb : containsIgnoreCase(?s, \text{“}rad9'')$
$\rightarrow tuo : Rad9(?someEntity)$

Figure 12: SWRL Rule TUO_00001

These two SWRL rules add the following instances to Rad9:

```
cpath:CPATH-92332
upkb:protein_0
```

These two instances can then be declared equivalent through the use of the `owl:sameAs` construct.

$tuo : Protein(?someEntity) \wedge$
$tuo : recommendedName(?someEntity, ?s) \wedge$
$swrlb : containsIgnoreCase(?s, \text{”}rad9\text{”})$
$\rightarrow tuo : Rad9(?someEntity)$

Figure 13: SWRL Rule TUO_00002.

$tuo : Protein(?someEntity) \wedge$
$tuo : hasDatabaseReference(?someEntity, ?dbref) \wedge$
$tuo : databaseName(?dbref, ?name) \wedge$
$swrlb : containsIgnoreCase(?name, \text{“}SGD'') \wedge$
$tuo : accession(?dbref, ?acc) \wedge$
$swrlb : equal(?acc, \text{“}S000002625'') \rightarrow$
$sqwrl : selectDistinct(?someEntity)$

Figure 14: SQWRL Rule TUO_SQWRL_00003. TUO_SQWRL_00002 (not shown here) is a version of TUO_SQWRL_00003, but with UniProt as the database name rather than SGD.

6.2.2 Identification of Equivalent Instances

The next step is to discover equivalent instances elsewhere in the integrated schema and mark them as such. We do this by mimicing the behaviour of the OWL2 construct `owl:hasKey` (see `http://www.w3.org/TR/owl2-syntax/#Keys`). The end result of this procedure is that all instances with the same SGD accession will be marked as identical using the `owl:sameAs` construct. In future, this will happen automatically as the relation linking a telomere ontology Protein to its SGD accession number will be classed as the key for the Protein class in an `owl:hasKey` construct.

Until the `owl:hasKey` construct is available, those instances having the same SGD accession are identified using the SQWRL query described in Figure 14, and then manually adding the `owl:sameAs` assertions. We use a SQWRL query rather than a SWRL rule here, as SWRL rules modify the target ontology.

There are already two instances of Rad9 as a result of the rules described in Figure 12 and Figure 13. After running the query in Figure 14 and viewing the results (`upkb:protein_0` and `psimif:interactor_59`), the `owl:sameAs` construct can be applied between those two instances. The new instance, `psimif:interactor_59`, does not contain ‘rad9’ in its name or synonyms, but does contain a matching SGD accession.

After inferring the placement of all individuals in the ontology using a reasoner, `psimif:interactor_59` is inferred as an instance of Rad9, bringing the total number of Rad9 instances to three:

```
cpath:CPATH-92332
upkb:protein_0
psimif:interactor_59
```

As all of these instances are marked as equivalent, the knowledge contained within each of them is accessible as a single logical unit.

6.2.3 Final Retrieval of Results For Use Case 1

In summary, the requirement for Use Case 1 is that information is retrieved about a protein containing the word “rad9” which belongs to the taxonomic species represented by the NCBI Tax ID 4932. The final step of this use case is to restrict the instances of Rad9 to the specified taxonomic identifier. The SQWRL query TUO_SQWRL_00001, shown in Figure 15, performs this function.

$tuo : Rad9(?someEntity) \land$
$tuo : hasTaxon(?someEntity, ?x) \land$
$tuo : ncbiTaxId(?x, ?y) \land$
$swrlb : equal(?y, 4932) \rightarrow$
$sqwrl : selectDistinct(?someEntity)$

Figure 15: SQWRL Rule TUO_SQWRL_00001.

$tuo : Rad9(?rad9instance) \land$
$tuo : plays(?rad9instance, ?participant) \land$
$tuo : hasParticipant(?process, ?participant) \rightarrow$
$sqwrl : select(?rad9instance, ?process)$

Figure 16: SQWRL Rule TUO_SQWRL_00004.

TUO_SQWRL_00001.csv in the zip file mentioned in Section 1 shows these results. All three instances are correctly returned in this query. The information contained within these three instances is the result of Use Case 1.

6.3 Use Case 2: Interactions

A full listing of all interactions discovered, including comments on those which are also in the SBML model used.

6.3.1 SWRL and SQWRL Used

Figure 16 contains the SQWRL query for identifying interactions instances of Rad9 are involved in, and the file TUO_SQWRL_00004.csv in the zip file mentioned in Section 1 shows these results.

6.4 Discussion of Results

While we have made extensive use of pre-existing applications, plugins, and libraries, we can forsee a time in the very near future where we will reach the limit of some of these technologies. Particularly, improvements will need to be made to the way the XML is converted into OWL, and the large number of instances/individuals required in a larger-scale semantic data integration project will most likely necessitate the use of a database back-end for the ontologies.

Further work will focus around automation of the proof-of-principle work related here. A large part of this research, while performed with existing tools, required an amount of manual intervention.

6.4.1 False Positives

By emulating the behaviour of the OWL2 hasKey construct we have shown the potential for false positives in defining two instances as identical. For instance, BioPAX has multiple types of cross-references. These include `unificationXref` (for cross-references linking to semantically identical entries in the other data resources) and `relationshipXref`, which implies a less rigorous link. If the wrong type of key is created, or an imprecise rule is used, then two instances could be incorrectly marked as identical (via the owl:sameAs construct). This could lead to false positives with respect to related, but not identical, instances.

Another source of false positives is the large number of interactions available relating directly to the gene or protein of interest.

6.4.2 Applicability to other Modeling Formalisms

References

[1] Stephan Philippi and Jacob Köhler. Addressing the problems with life-science databases for traditional uses and systems biology. *Nature Reviews Genetics*, 7(6):482–488, May 2006.

[2] W. Sujansky. Heterogeneous database integration in biomedicine. *J Biomed Inform*, 34(4):285–298, August 2001.

[3] R. Alonso-Calvo et al. An agent- and ontology-based system for integrating public gene, protein, and disease databases. *J Biomed Inform*, 40(1):17–29, February 2007.

[4] Natalya F. Noy and Mark A. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *Int. J. Hum.-Comput. Stud.*, 59(6):983–1024, December 2003.

[5] J. Lomax and A. T. McCray. Mapping the gene ontology into the unified medical language system. *Comparative and functional genomics*, 5(4):354–361, 2004.

[6] H. F. Deus, R. Stanislaus, D. F. Veiga, C. Behrens, I. I. Wistuba, J. D. Minna, H. R. Garner, S. G. Swisher, J. A. Roth, A. M. Correa, B. Broom, K. Coombes, A. Chang, L. H. Vogel, and J. S. Almeida. A semantic web management model for integrative biomedical informatics. *PLoS ONE*, 3(8), 2008.

[7] Barry Smith et al. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat Biotech*, 25(11):1251–1255, November 2007.

[8] Christine Golbreich et al. OBO and OWL: Leveraging Semantic Web Technologies for the Life Sciences. In *The sixth International Semantic Web Conference (ISWC 2007)*, pages 169–182. 2007.

[9] Franz Baader, Diego Calvanese, Deborah Mcguinness, Daniele Nardi, and Peter Patel-Schneider, editors. *The Description Logic Handbook - Cambridge University Press*. Cambridge University Press, first edition, January 2003.

[10] Marie C. Rousset and Chantal Reynaud. Knowledge representation for information integration. *Inf. Syst.*, 29(1):3–22, 2004.

[11] Li Xu and David W. Embley. Combining the Best of Global-as-View and Local-as-View for Data Integration. In Anatoly E. Doroshenko, Terry A. Halpin, Stephen W. Liddle, Heinrich C. Mayr, Anatoly E. Doroshenko, Terry A. Halpin, Stephen W. Liddle, and Heinrich C. Mayr, editors, *ISTA*, volume 48 of *LNI*, pages 123–136. GI, 2004.

[12] Peter Li, Tom Oinn, Stian Soiland, and Douglas B B. Kell. Automated manipulation of systems biology models using libSBML within Taverna workflows. *Bioinformatics*, December 2007.

[13] C. Stark et al. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res*, 34(Database issue), January 2006.

[14] The UniProt Consortium. The universal protein resource (UniProt). *Nucleic Acids Res*, 36(Database issue), January 2008.

[15] A. L. Lister, M. Pocock, and A. Wipat. Integration of constraints documented in SBML, SBO, and the SBML Manual facilitates validation of biological models. *Journal of Integrative Bioinformatics*, 4(3):80+, 2007.

[16] H. Hermjakob et al. The HUPO PSI's molecular interaction format–a community standard for the representation of protein interaction data. *Nat Biotechnol*, 22(2):177–183, February 2004.

[17] Markus J. Herrgard, Neil Swainston, Paul Dobson, Warwick B. Dunn, Yalcin K. Arga, Mikko Arvas, Nils Buthgen, Simon Borger, Roeland Costenoble, Matthias Heinemann, Michael Hucka, Nicolas Le Novere, Peter Li, Wolfram Liebermeister, Monica L. Mo, Ana P. Oliveira, Dina Petranovic, Stephen Pettifer, Evangelos Simeonidis, Kieran Smallbone, Irena Spasie, Dieter Weichart, Roger Brent, David S. Broomhead, Hans V. Westerhoff, Betul Kurdar, Merja Penttila, Edda Klipp, Bernhard O. Palsson, Uwe Sauer, Stephen G. Oliver, Pedro Mendes, Jens Nielsen, and Douglas B. Kell. A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nat Biotech*, 26(10):1155–1160, October 2008.

[18] Holger Wache et al. Ontology-based integration of information — a survey of existing approaches. In H. Stuckenschmidt, editor, *Proceedings of the IJCAI'01 Workshop on Ontologies and Information Sharing, Seattle, Washington, USA, Aug 4-5*, pages 108–117, 2001.

[19] Jinguang Gu, Baowen Xu, and Xinmeng Chen. An XML query rewriting mechanism with multiple ontologies integration based on complex semantic mapping. *Information Fusion*, 9(4):512–522, October 2008.

BGLaV BYU Global-Local as View

DL Description Logic

GAV Global As View

LAV Local As View

MFO Model Format OWL

OBO Open Biomedical Ontologies

OWL Web Ontology Language

OWL-DL OWL constrained by Description Logics

PPI Protein-Protein Interaction

Saint SBML Annotation Integration Environment

SBO Systems Biology Ontology

S. cerevisiae Saccharomyces cerevisiae

SBML Systems Biology Markup Language

SQWRL Semantic Query-Enhanced Web Rule Language

SWRL Semantic Web Rule Language