

Using Ontology with Semantic Web Services to Support Modeling in Systems Biology

Zhouyang Sun, Anthony Finkelstein, Jonathan Ashmore

CoMPLEX, University College London, WC1E 6BT, London

{j.sun, a.finkelstein}@cs.ucl.ac.uk, j.ashmore@ucl.ac.uk

Abstract. Modeling in systems biology is concerned with using experimental information and mathematical methods to build quantitative models at different biological scales. This requires interoperation among various knowledge sources and services, such as biological databases, mathematical equations, data analysis tools, and so on. Semantic Web Services provide an infrastructure that allows a consistent representation of these knowledge sources as web-based information units, and enables discovery, composition, and execution of these units by associating machine-processable semantics description with them. In this paper, we show a method of using ontology alongside a semantic web services infrastructure to provide a knowledge standardisation framework in order to support modeling in systems biology. We demonstrate how ontologies are used to control the transformation of biological databases and data analysis methods into Web Services, and how ontology-based web services descriptions (OWL-S), are used to enable the composition between these services.

Keywords: Ontology, OWL, Semantic Web Services, OWL-S, Java EE

1. Motivation

Systems biology is an emergent discipline that involves integrating biological knowledge across scales and domains, in order to understand the dynamics of diverse and interacting biological processes as integral systems [1]. In the study of systems biology, one of the essential tasks is to couple experimental biologists' observations with scientific models. This task encompasses several collaboration processes involving experimenters and modelers: using experimental observations as the ground for constructing models of biological entities and the relations among them; qualitatively and quantitatively analysing the resulting models and then comparing the analyses against experimental data for model validation; providing instructive feedback in order to refine both the models and experimental protocols. The progress of systems biology relies on the success of these experimenter-modeler collaboration processes.

Experimenter-modeler collaboration processes present a number of challenges. Firstly, both the content and the format of knowledge that need to be shared among participants are diverse. For instance, this knowledge can be experimental data with descriptions of laboratory settings, or mathematical models that quantitatively represent relations among biological entities. The languages used to represent models may not be directly compatible. The computing environments used to store experimental data are usually heterogeneous.

Secondly, knowledge creation in systems biology relies on combining knowledge from many various and distributed sources, in order to achieve a system-level understanding. For instance, when studying complex biological systems such as human liver, researchers may have to integrate knowledge from gene regulation level up to intercellular communication level for investigating certain physiological phenomena. Models on different levels may be developed independently by several groups using various modeling paradigms and computational environments.

Thirdly, models and information from experiments are reused in many different settings. When specifying parameters for an equation in a biological model, modelers need to interact with various biological data sources such as literature, biological databases, or data embedded in existing models. Then they need to give justification on the selection of data sources according to the context of the model, make judgment on which analysis methods and parameterisation approaches need to be applied. All the above information can be crucial for model reuse. Newcomers may build a model based on the same rationale of an existing model, but want to use alternative data sources and parameterisation methods. In this case, the reasoning involved in the model construction could rely on the previous cases.

2. Approach

In order to tackle these challenges, a common means of formally representing diverse knowledge in computer-based format is required. Also, in the context of combining computer-based resources, it is required for the individual pieces of knowledge to have a 'descriptive interface' to support computer-based communication, so that knowledge can be easily integrated. Moreover, the collaboration processes that mediate between distributed knowledge representation are part of the knowledge of modeling and should be formally represented for future model reuse.

We use ontology and semantic web services [15] as the primary means to meet these requirements. Ontology is the theory of conceptualisation. In computing, ontology provides a means of formally representing the structure of objects and relations in an information system and associating meaning with them [2]. Ontology can provide

formal knowledge representation for distributed and heterogeneous computer-based biological information. Moreover, since ontologies explicitly define the content in information sources by formal semantics, they also enable the basis of interoperability between these sources. Further, as ontologies are able to separate domain knowledge from application-based knowledge, they can be used to define the collaboration processes among information-providing applications. Ontologies provide the benefits of reuse, sharing and portability of knowledge across platform [3].

Semantic Web Services are the conjunction of Semantic Web and service-oriented computing. The Semantic Web is a framework for creating a universal medium for information exchange by associating semantics with documents on the World Wide Web [4]. Service-oriented computing is a software architecture that allows information resources to be presented as platform-independent, self-describing, modular software units. The combination of both provides a web-based infrastructure for general knowledge sharing and reuse. Semantic Web Services allow the bringing together of knowledge sources as Web Services and describing interaction and workflows among them by means of a semantic markup language, and therefore is able to meet the need of modeling knowledge collaboration among systems biology practitioners. Constructing biological models in a portable exchange format is challenging because it requires assembling knowledge from heterogeneous sources. Semantic web techniques can help in building such model by supporting meaningful descriptions of the elements drawn from these sources and thus enhancing the interoperability and consistency among them.

In this paper, we will show how to help model construction by using ontology and ontology-based mapping of model components to Semantic Web Service architecture. We will describe a simple case of how the model is built in practice. Then we will describe an ontology for modeling in systems biology and how it is used to automate the transformation of biological databases, data analysis methods into web services, as well as the transformation of mathematical equation to OWL-S, the semantic markup for web services, for interactive parameterisation processes. We will describe our approach to ontology-based modeling in systems biology and the procedure for using an ontology model to control the transformation of experimental data and mathematical methods into Semantic Web Services, as well as the composition of these services for parameterising equations in the biological models.

3. A Simple Case of Modeling in Systems Biology

We describe a case of model construction for explaining the details of our approach. The model used is developed by Hudspeth, A.J. and Lewis, R.S. [5], and has been

built to aid understanding electrical resonance in bull-frog hair cells. We chose this case because it encompasses all the major processes involved in the study of systems biology including data acquisition from experiments, abstract modeling, raw data analysis for parameterisation, and creation of simulations. Also, to construct this kind of models often requires integrating fairly complex sub-models of different types, which can demonstrate the complexity of the modeling tasks in systems biology.

To create the model, firstly an experiment is carried out using electrophysiological recording techniques. Data of cell responses to a series of electric stimuli are acquired and stored in a certain computer-based format. Meanwhile, abstract models are proposed which describe the system of interest as the entities and the relationships between them. In this case, the abstract models include a kinetic model for voltage-dependent calcium current, a diffusion model for the regulation of intracellular calcium ions, and a kinetic model for Ca²⁺-activated K⁺ current based on the kinetics of Ca²⁺-binding K⁺ channel. The relationships in these models are described by mathematical equations. For example, voltage-dependent calcium current is described by a third-order kinetic scheme [6],

$$I_{Ca} = \bar{g}_{Ca} m^3 (V_m - E_{Ca})$$

In which \bar{g}_{Ca} , m , and E_{Ca} are parameters; I_{Ca} and V_m are membrane current and potential variables.

In order to specify these models, experimental data are assessed for parameterisation. For example, to evaluate the conductance of voltage-dependent calcium channel, the data of tail currents are retrieved from datasheets and are fitted with a Boltzmann relation, $I = I_{max} / \{1 + e^{-(V_m - V_{1/2})/k}\}^3$, by a least-square-error criterion. Besides the parameters that are acquired by data analysis methods, some of the parameters are given from existing literature and some are given arbitrarily. No record will be kept on the data analysis and parameterisation processes. . Every time therefore, when the model needs to be modified or used for integration with other models, researchers have to go back to the data and analyse them again. This can be problematic, since models are shared among researchers but not everyone has access to the original data used for parameterisation, or the rationale for data source and mathematical methods selection. After the parameterisation, the models are fully instantiated. At this point, simulation can be created in certain computing languages such as Java etc.

4. A Simple Case of Modeling in Systems Biology

4.1 Implementation Details

For ontology model construction, we use Protégé 3.2.1 [7], an ontology editor by Stanford Medical Informatics. For generating web services, we use NetBeans 5.5 [8] by Sun Microsystems, a Java EE-enabled [9] integrated development environment (IDE) to generate Web Services. NetBeans 5.5 is an open-source IDE written entirely in Java featuring APIs including Java Persistence and JAX-WS [10], and bundles with Sun Java System Application Server Platform (SJSAP) 9.1 [11].

4.2 Create abstract biological models by using ontology

In our approach, we construct an abstract model of the biological systems of interest as the starting point. We use OWL DL [12] as the format of our ontology models, since OWL DL is a species of OWL [12] that provides the adequate expressiveness and has desirable computational properties for reasoning. By using ontology as the medium, the biological models we construct will be portable, integratable, and can be reasoned about using description logic.

We create an OWL abstract model as the meta-model for constructing biological models. In order to make it easy to understand, instead of using XML code we present it in a UML [13] diagram as bellow:

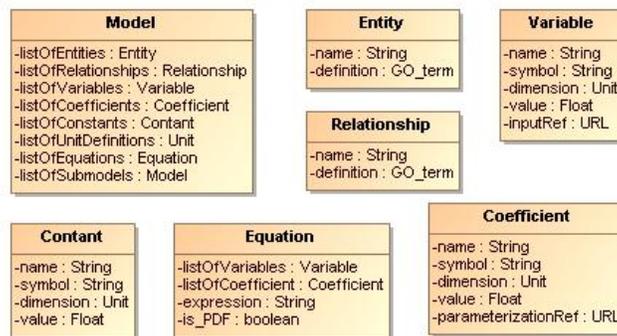


Fig. 1. UML Class diagram to represent the metadata of the OWL-based biological model

This model declares the information that needs to be assembled for constructing a valid biological model. The information includes the definitions of biological entities and biological processes, mathematical equations underlying the biological processes.

This model is then used to embed or refer to external sources of parameters, data sources and analysis methods used for parameterisation.

The following is a sample fragment of an OWL model instantiated by the case of Lewis & Hudspeth hair cell model:

```
<Model rdf:ID='`regulation_of_intracellular_calcium_ion`'>
  <listOfCoefficients
    rdf:resource='`#rate_constant_of_calcium_ion_migration`' />
  <listOfCoefficients
    rdf:resource='`#fraction_of_free_intracellular_calcium_ion`' />
  ..... <!--List of Coefficients -->
  <listOfVariables rdf:resource='`#concentration_of_calcium_ion`' />
  <listOfVariables rdf:resource='`#calcium_current`' />
  ..... <!--List of Variables -->
  <listofEntities>
    <GO_0005623 rdf:ID='`saccular_hair_cell_of_bull_frog`' />
  </listofEntities>
  ..... <!--List of Entities -->
  <listOfEquations
    rdf:resource='`#concentration_of_intracellular_calcium_ion`' />
  ..... <!--List of Equations -->
  <Equation rdf:ID='`concentration_of_intracellular_calcium_ion`'>
    <Variable rdf:ID='`concentration_of_calcium_ion`'>
      <symbol rdf:datatype='`http://www.w3.org/2001/XMLSchema#string`'
        >c</symbol>
      <value rdf:datatype='`http://www.w3.org/2001/XMLSchema#float`'
        >0.1</value>
      <unit rdf:resource='`#micromolar`' />
      <initialValue
rdf:datatype='`http://www.w3.org/2001/XMLSchema#float`'
        >0.0</initialValue>
    </Variable>
    .....
    <coefficient rdf:resource='`#fraction_of_calcium_accumulation`' />
    <coefficient rdf:resource='`#total_cell_volume`' />
    .....
    <is_PDE rdf:datatype='`http://www.w3.org/2001/XMLSchema#boolean`'
      >true</is_PDE>
    <expression rdf:datatype='`http://www.w3.org/2001/XMLSchema#string`'
      >c = c + dtime * (- 1.0 *Ui_Ca/(z*F*C_vol*xi)-K_S*c)
    </expression>
  </Equation>
</Model>
```

4.3 From experimental Data to Database Web Services

With the development of Java EE and its supporting APIs, we are able to transform experimental data stored in different formats into standard database web services. The general procedure of transformation is the following:

- Transform data source into relational database with a generic schema
- Generate Java Entity Classes from relational database

- Define generic operations to create Web services
- Deploy Web Services to generate WSDL
- Create semantic descriptions (OWL-S) for the generated web services

Experimental data exist in different formats such as data-containing documents or relational databases. Different kinds of data-containing documents can always be converted into text-based spreadsheets, which can be imported into relational database by using SQL statements [14]. These database web services are deployed in Java EE enabled application server and provide full flexibility of data retrieval, so that any analysis methods can be performed on them.

Since most of the procedure in this case has been simplified by Java EE platform, the remaining tasks are mainly about defining the meta-models of the database structure, the description of the web service competence, and the semantics for advertising the generated web services. These all can be modeled by using ontology. Moreover, as the operations of the database web services are dependent on the database structure, and the profiles of the web services are based on the description of the entities in the database, a single ontology model can be used to control the generation of the above information all at the same time.

In the case of Lewis & Hudspeth model, data are saved in a spreadsheet which contains both the specifications of the experiment and the recorded electrophysiological data. The specifications of the experiment may include the profile of the electric stimulus performed on the cells, the definitions of signals, the meaning of the columns in the tables of data, etc. The results of experiment are stored in a table whose columns are specified with names of entities and rows are sequences of recorded data.

In order to transform this kind of data sources into database web services, we defined an OWL model that describes the properties included in the settings, the entities that specify the columns of the data table, and the operations supported by the web services to be generated. By using the OWL model, database schema, semantic description of web services are automatically generated.

4.4 From Analysing Methods to Web Services

We use JAXWS API [10] in Java EE to transform mathematical methods into Web Services. It is a simple process. Any programming implementation of mathematical methods, such as calculation or best fitting methods etc., is transformed by using annotations as specified in A Metadata Facility for the Java Programming Language

(JSR 175) and Web Services Metadata for the Java Platform (JSR 181), as well as additional annotations defined by the JAX-WS 2.0 specification. For example, the following is the implementation of the Boltzmann relation in our example, as a web service ($I = I_{\max} / \{1 + e^{-(V_m - V_{1/2})/k}\}^3$).

```

import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;

@WebService
public class BoltzmannRel {
    public BoltzmannRel () {}

    @WebMethod(operationName= ``Boltzmann Relation ``)
    public float BoltzmannRel (@WebParam(name = ``Membrane
Potential``) float var1, @WebParam(name = ``Peak Current``) float var2,
@WebParam(name = ``Steady State Potential``) float var3, @WebParam(name
= ``slope factor``) float var4) {
        float result = var2/math.pow((1+math.exp(-1*(var1-
var3)/var4)),3) ;
        return result;
    }
}

```

Similar to the generation of database web services, we also define data analysis web services by using ontology models. The meta-model consists of the description of the inputs, outputs, preconditions, and post-conditions of the mathematical methods. The instantiations of this meta-model are then translated into web services implementation automatically by using an XML transformation stylesheet.

4.5 Use OWL-S to specify Parameterisation in Computational Models

After experimental data and mathematical methods are transformed into web services and annotated with semantics, we orchestrate them together by using the web service composer developed by MINDSWAP. Web service composer is a software interface developed by Maryland Information and Network Dynamics Lab Semantic Web Agents Project (MINDSWAP) [16]. It can be used to guide users in the dynamic composition of web services by supporting OWL-S [17] standard. Using the composer one can generate a workflow of web services. The composition is done in a semi-automatic fashion where composer discovers web services by reasoning on their semantic descriptions, then presents the available web services at each step to a human controller to make the selection. The generated composition can be directly executable through the WSDL grounding of the services. Compositions generated by the user can also be saved as a new service which can be further used in other compositions.

In this work, we use OWL-S to specify the parameterisation process in model construction: we translate mathematical equations embedded in the OWL-based biological models into OWL-S files as template web service composition profiles.

For example, when we parameterise equation $\alpha_m = \alpha_0 e^{-(V_m + V_0)/k_B} + K_A$ in which α_0, K_A, K_B are parameters need to be specified, we create an OWL-S file that defines web service composition for the parameterisation of this equation. This OWL-S file can be handled by the composer and an interactive interface is then. To parameterise α_0 , this interface searches all the available databases web services and analyser web services and enables orchestrating a selection of these services together. A new composition can be specified on what data to retrieve from database web services and then pass to analyser web services in order to obtain the value of α_0 . After all the parameters are obtained, the values and the equation will be passed to an equation parser service to generate an instantiated equation (looks like $\alpha_m = 22800e^{-(V_m + V_0)/33} + 510$). When all the equations in a model are parameterised, they can then be passed to simulation generation service.

After the parameterisation process, the information of databases and analysis methods used is stored in OWL-S files and published on the web as new web services. We embedded the links to these OWL-S back to the OWL-based biological models, so that when users want to reuse these models, they are able to retrieve the parameterisation processes and modify them themselves.

4.6 Result

By these means we constructed a fully specified ontology model for the biological system of interest, i.e. the model of hair cell electrophysiology. All parameters that are specified interactively are annotated with external links to OWL-S files that represent the composition of database web services and data analysis web services used for parameterisation. This OWL-based biological model consists of all the essential information for generating a computational simulation.

The meta-model we proposed for modeling in systems biology contains the information specified by the XML schemas of SBML and CellML. Therefore, a subset of our model can be transformed into either SBML or CellML. This gives us the advantage of using any SBML or CellML-enabled software. For example, we have successfully transformed our OWL-based Lewis&Hudspeth model into CellML format and use it in Cell Electrophysiological Simulation Environment (CESE) [18]. A simulation is then generated automatically by transforming CellML model to JavaBeans programs.

5. Related Work

There are a number of XML-based specifications for modeling in systems biology. Representatives of such efforts are the Systems Biology Markup Language (SBML) and the Cell Markup Language (CellML). Both SBML and CellML are XML-based exchange formats that provide formal representation of main modeling components including biological entities, parameter definitions and the equations of the underlying biological processes such as reaction mechanisms, etc [19, 20]. These efforts have however, principally been focused on improving the exchanging models between simulation environments [21]. Both SBML and CellML do not contain information on parameterisation or associated rationale. Our model provides a novel way to combine semantic web services infrastructure, so that data sources and mathematical methods can be standardised in web-based units and then integrated together to achieve interoperation.

There are also attempts at creating biological Web services to enable e-Science in systems biology. An increasing number of tools and databases in molecular biology and bioinformatics are now available as Web Services. For example, *Nucleic Acids Research* describes 968 databases and 166 web servers available in molecular biology [3, 22]. Web service composition frameworks have also been developed for biological studies, including BioMoby [23] and myGrid [24], which support workflow design and execution, data management, and provenance collection among distributed biological web services. Almost all the existing efforts are however focused on publishing genomic data, such as DNA sequence, protein sequence, nucleotide sequence, and so on, and web services are mainly for sequence alignment or looking up definitions of biological terms. As far as we are aware, there is no web service available for physiological level simulation. Furthermore, our approach brings a collection of the latest semantic web techniques in a way that no others have proposed before.

6. Discussion and Future Work

In this work, we described an ontology-centered framework that uses a formal ontology language (OWL) to construct biological models in a portable exchange format, controls the transformation from biological data sources and data analysis methods into semantic web services, and defines the composition of these services in an OWL-based semantic description of web services (OWL-S). The main contribution of our work is the design of this novel framework. Currently no similar effort exists. This framework applies the latest advancement of information technology and is organised in a unique fashion specifically for the purpose of tackling the challenges of

biological modeling. This framework allows users to build biological models from an abstract view, so that the biological entities and relationships underlying biological processes are based on the same shared vocabulary. The semantics in these models also enhance the interoperability of biological models, so that they can be easily integrated. Moreover, this framework directly connects with semantic web services infrastructure through ontology models, so that biological data sources and data analysis tools are published as web services. The transformation processes from the knowledge sources to web services are directly controlled by the ontology models. With the advantages of ontology, the automation of transformation can be achieved and consistency between different components can be maintained. Furthermore, this ontology-centered annotation/transformation framework allows users to control the parameterisation processes by semantically defining the workflow between these services. The parameterisation processes are saved in OWL-S models and can be retrieved and dynamically modified. This mechanism allows model specifications and the rationale of modeling processes to be associated with the representation of models, and therefore provides solid ground for further model reuse. The biological models written in OWL can also be translated into simulation models directly. This allows users to build models in abstraction and valid these models in real-time.

This framework provides a basis for combining future development of semantic web techniques to support more sophisticated modeling tasks. For example, by constructing biological models in OWL format, reasoning engines that support description logic can be used to infer the relations between entities across different models. This can help the automatic discovery and matching of distributed models and control the consistency of model integration. Moreover, the OWL-S models that describe parameterisation processes can be combined with case-based reasoning engines so that modeling tasks can be solved automatically by reusing existing cases. Further, the semi-automatic transformation from abstract model to simulation provided by the framework can allow users to construct model heuristically and validate them against experiments in real time. All of these can be very beneficial to modeling in systems biology. We require further experience to be confident of the broader applicability of our work though clearly this has been an important design goal of the framework.

So far our approach has only been tested on a relatively simple example; therefore, the scalability of the framework has not been verified. Also, we have focused the modeling tasks on constructing electrophysiological models. It still needs to be determined whether this framework is sufficiently generic to handle other kinds of

biological models. In future, we will focus on examining the framework with more complicated cases and different model types.

Reference

1. Finkelstein, A., et al.: Computational Challenges of Systems Biology. IEEE Computer, Vol. 37 No. 5 (2004) 26-33
2. Noy, N.F., and McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology - what is an ontology and why we need it. <http://protege.stanford.edu/publications/>
3. Baker, C.J.O., Cheung, K-H.,: Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences. Springer Science and Business Media, LLC 1st Edition (2007)
4. Berners-Lee, Semantic Web on XML - Keynote presentation for XML 2000, <http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide1-0.html>
5. Hudspeth, A.J and Lewis, R.S.: Kinetic Analysis of Voltage and Ion-dependent Conductances in Saccular Hair Cells of the Bull-frog, *Rana Catesbeiana*. J Physiol. 400 (1998) 237-274
6. Hodgkin, A.L. and Huxley, A.F: A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve. J Physiol. 117 (1952)500-544
7. Protégé, <http://protege.stanford.edu/>
8. Netbeans, <http://www.netbeans.org/index.html>
9. Java EE, <http://java.sun.com/javaee/>
10. JAX-WS, <https://jax-ws.dev.java.net/>
11. Application Server, <http://java.sun.com/javaee/community/glassfish/>
12. OWL, <http://www.w3.org/TR/owl-features/>
13. UML, <http://www.uml.org/>
14. SQL, <http://en.wikipedia.org/wiki/SQL>
15. Semantic Web Services, http://en.wikipedia.org/wiki/Semantic_Web_Services
16. Web Service composer, <http://www.mindswap.org/2005/composer/>
17. OWL-S, <http://www.daml.org/services/owl-s/1.1/>
18. CESE, <http://cese.sourceforge.net/>
19. SBML, <http://www.sbw-sbml.org/>
20. CellML, <http://www.cellml.org/>
21. Uhrmacher, A.M., et al.,: Towards Reusing Model Component in Systems Biology. Danos and V. Schachter (Eds.): CMSB 2004, LNBI 3082, Springer-Verlag Berlin Heidelberg (2005) 192–206
22. Nucleic Acids Research, <http://nar.oxfordjournals.org/>
23. BioMoby, <http://biomoby.org/>
24. myGrid and Taverna services, <http://mygrid.org.uk/>