Centre f**o**r **M**athematics & **P**hysics in the **L**ife Sciences and
EXperimental Biology
University College London

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

Zhouyang Sun

Submitted for the degree of Doctor of Philosophy
At University College London
December 2008

Revised for the final submission 2009

I, Zhouyang Sun, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signature:

Date:

Abstract

This thesis addresses the problem of collaboration among experimental biologists and modelers in the study of systems biology by using ontology and Semantic Web Services techniques. Modeling in systems biology is concerned with using experimental information and mathematical methods to build quantitative models across different biological scales. This requires interoperation among various knowledge sources and services. Ontology and Semantic Web Services potentially provide an infrastructure to meet this requirement.

In our study, we propose an ontology-centered framework within the Semantic Web infrastructure that aims at standardizing various areas of knowledge involved in the biological modeling processes. In this framework, first we specify an ontology-based meta-model for building biological models. This meta-model supports using shared biological ontologies to annotate biological entities in the models, allows semantic queries and automatic discoveries, enables easy model reuse and composition, and serves as a basis to embed external knowledge. We also develop means of transforming biological data sources and data analysis methods into Web Services. These Web Services can then be composed together to perform parameterization in biological modeling. The knowledge of decision-making and workflow of parameterization processes are then recorded by the semantic descriptions of these Web Services, and embedded in model instances built on our proposed meta-model.

We use three cases of biological modeling to evaluate our framework. By examining our ontology-centered framework in practice, we conclude that by using ontology to represent biological models and using Semantic Web Services to standardize knowledge components in modeling processes, greater capabilities of knowledge sharing, reuse and collaboration can be achieved. We also conclude that ontology-based biological models with formal semantics are essential to standardize knowledge in compliance with the Semantic Web vision.

TABLE OF CONTENT

Table of Figures

Chapter 1 : Introduction

1.1 Background

Modeling in systems biology is concerned with using experimental information and mathematical methods to build quantitative models across different biological scales. This requires interoperation among various knowledge sources and services, such as biological databases, mathematical equations, data analysis tools, and so on. Ontology along with 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 to them. Therefore, there is an emerging need to adapt the modeling tasks in biological modeling to the Semantic Web vision.

1.2 Methods, Contributions, and Originality

Our method is concerned with using ontology alongside Semantic Web Services infrastructure to provide a knowledge standardization framework for supporting modeling in systems biology. We demonstrate how ontologies are used to build biological models and connect the transformation of biological databases and data analyzing methods into Web Services, and how ontology-based Web Services descriptions are used to enable the composition between these services.

We propose an ontology-based meta-model for building biological models in OWL (McGuinness and Harmelen, 2004) format. This meta-model provides capabilities beyond non-semantic models such as those in CellML (Cuellar et al., 2003) and SBML (Finney and Hucka, 2003) formats. We also develop means of transforming biological data sources and data analysis methods into Web Services. These Web Services can then be composed together to perform parameterization in biological modeling. The knowledge of decision-making and workflow of parameterization processes are then recorded by the semantic descriptions of these Web Services in OWL-S (W3C, 2004) format, an OWL-based ontology language, and embedded in model instances built on our proposed meta-model.

The main contribution of our work is the design of this ontology-centered computational framework that can facilitate modeling tasks in systems biology. This framework applies the latest advances of information technology, in order to enable the standardization and collaboration of knowledge components in biological modeling. This framework is organized in a unique fashion specifically to achieve our goal.

Currently there is no existing effort that uses formal ontology to represent biological models. The original contributions of this work includes the process of transforming knowledge components such as experimental databases and data analysis methods to Semantic Web Services, as well as the embedment of decision-making knowledge and workflow of parameterization in knowledge exchange format . Our work has been accepted and presented in a 2007 international conference on Web Information Systems Engineering. The result has been welcome in the biological modeling community (Sun et al., 2007).

1.3 Thesis Outline

The content of the thesis is arranged as follows: in Chapter 1 and 2 we will introduce what systems biology is, what modeling in systems biology involves, and what challenges we face motivating our study of using ontology and Semantic Web Services to support modeling in systems biology. Then, in Chapter 3 we will review the work and technologies related to our study within the Semantic Web for further discussion of the technologies needed to achieve our goal.

Following the introductory chapters, in Chapter 4 we will give several case studies in biological modeling, in order to provide real examples to support our analysis of the challenges in modeling in systems biology. These cases will also be used to validate the solution we propose.

Chapter 5 will investigate the knowledge involved in biological modeling and identify the Semantic Web technologies suitable to standardize all the modeling knowledge. We then propose our approach of an ontology-centered framework that formally represents biological models and connects the transformation of biological

databases, data analysis methods into Web Services, and semantic markup for these Web Services.

Chapter 6 then gives the step-by-step description of the construction of such framework in full detail. And in Chapter 7 the framework is evaluated by the case studies we introduced.

Finally, in Chapter 8 and 9 we will discuss the effectiveness of our approach, provide a roadmap for future work, and state our conclusions.

Chapter 2 : Motivation

2.1 What is Systems Biology?

Over the past decades, the development of molecular techniques has prominently reshaped our understanding of life sciences. From the intricacies and mechanistic details of genetic information transfer to the network of biochemical interactions within cells, tissues and organs, molecular biology has revealed to us the astonishing complexity of biological systems. In the process of achieving this revolution of understanding, molecular biology has accumulated a vast amount of information. With the blossoming of molecular technologies, information acquisition now proceeds too fast to analyze and interpret with available tools. It is urgent for us to start making sense of many rich areas of biological information. Besides the genome, which constitutes our knowledge about genes, we also need to work on the proteome, metabolome, and physiome, which embody the knowledge about proteins, metabolic processes, and physiology (Finkelstein et al., 2004). To understand how the various pieces interact to produce complicated biological activities, we must return to the study of whole biological systems: the heart, brain, and liver, which address an emerging discipline - Systems Biology.

As one of the most exciting scientific challenges of today, Systems Biology aims at the study of extracting knowledge from the increasing detailed data that run across scales, and integrating this into a comprehensive analytical description of biological system with predictive power. Modeling lies at the heart of Systems Biology. We can use experimental information to build models at different biological scales, integrating them to create a hierarchical model composition ranging from DNA and gene expression to intracellular networks, to cell-to-cell and transmembrane signals, and through to the organ level. Eventually, we should be able to construct such models at the organism level. “The resulting models can provide prediction that is used as a scaffold for our understanding of the data, identify gaps in our biological knowledge, and predict new behaviors that we can explore experimentally.” (Finkelstein et al., 2004)

2.2 What is Involved in Modeling in Systems Biology?

In the study of systems biology, one of the essential tasks is to couple experimental biologists' observations with scientific models. This task encompasses several collaborative 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 analyzing 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 parameterization 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 parameterization methods. In this case, the reasoning involved in the model construction could rely on the previous cases (Sun et al., 2007).

2.3 Semantic Web for Modeling in Systems Biology

In order to tackle these challenges, a common means of formally representing various 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 representation to have a descriptive interface for computer-based communication, so that knowledge can be easily integrated. Moreover, the collaboration processes 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 as the central means to meet these requirements. Ontology is the theory of conceptualization. In computing, ontology provides a means of formally representing the structure of objects and relations in an information system and associating meaning with them. Therefore, 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.

Semantic Web Services is 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. 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 allows unifying knowledge sources as Web Services and describing interacting and workflows among them by means of a semantic markup language, and therefore is able to fulfill the need of

modeling knowledge collaboration among systems biology practitioners. In this thesis, we will show how to help model construction by using ontology and ontology-based mapping of model components to Semantic Web Service architecture (Sun et al., 2007).

2.4 Chapter Summary

In this chapter we have introduced what systems biology is, what modeling in systems biology involves, and that the challenges that motivate our study of using ontology and Semantic Web Services to support modeling in systems biology. In the following chapter we will review the work and technologies related to our study within the Semantic Web infrastructure.

Chapter 3 : Review of Techniques and Related Work

In the previous chapter, we have identified the key problems that researchers are facing when carrying out modeling tasks in systems biology, i.e. how to formally represent various knowledge in computer-based format, how to access distributed knowledge representation including biological data, models and modeling methods, how to integrate the distributed knowledge together in a reusable way, and how to represent knowledge interoperation processes and their associated rationale for further investigation.

Success in tackling these problems depends on our ability to represent, share, query, integrate, and reuse knowledge sources. With the emergence of the Semantic Web vision as an extension to the Internet, useful tools have become available to achieve our goal. The Semantic Web infrastructure consists of many component technologies including ontology, web ontology languages, ontological databases, agent systems, Web Services, and associated web standards (Baker and Cheung, 2007). The combination of these technologies has the potential to present and provide access to complex and diverse knowledge in a standardized way while enabling collaboration among knowledge sources and middleware applications. Therefore, we use Semantic Web infrastructure as the foundation to facilitate the various tasks involved in modeling in systems biology.

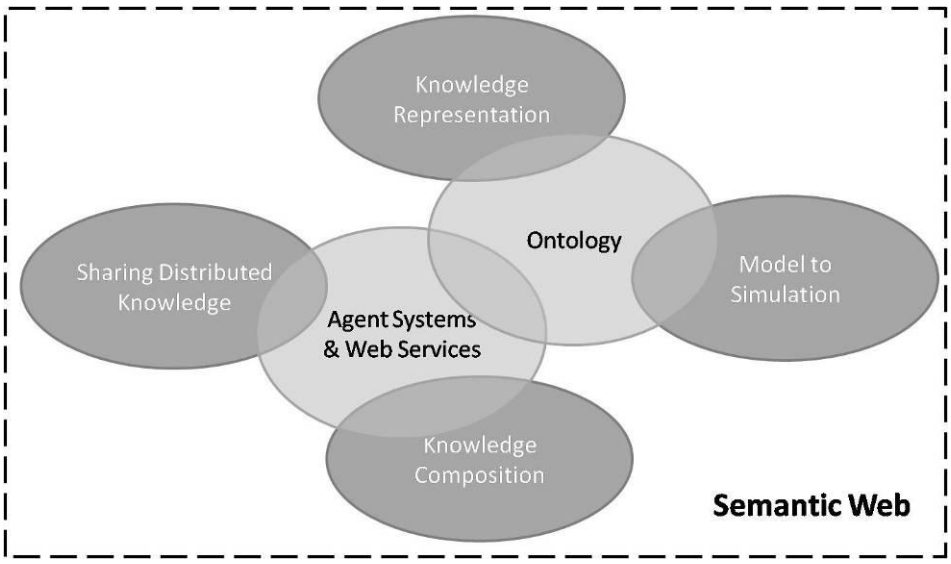


Figure 3.1 Problems of Biological Modeling and the Semantic Web

Figure 3.1 generally categorizes the problems involved in modelling in systems biology into four areas. As these areas are not necessarily independent from each other, this classification is a simplified guide. Its purpose is to help identifying the common ground of underlying technologies in the context of Semantic Web that can help us tackling the problems in biological modeling.

With the fast growing appreciation of the value that the Semantic Web offers, a considerable amount of effort has been made to develop or apply component technologies of Semantic Web for life sciences. Among these component technologies, ontology is at the heart of all the solutions that aim at using Semantic Web to facilitate knowledge sharing and reuse in the life science community. At the same time, agent-based systems and Web Services, which use ontology as their foundation technology for knowledge encoding, help improving the interoperation of distributed biological knowledge among researchers (see Figure 3.1). In the following chapter, we introduce both ontology and agent system & Web Service technology, and review the development status of existing efforts that apply these technologies in the study of life sciences in general.

3.1 Ontology

Ontology serves as the central component of the Semantic Web's knowledge representation infrastructure and the foundation of many other component technologies. Ontology is now a research discipline in its own right and interest in applications of ontology-based technologies is strong. In the following discussion, first we introduce the basic theory of ontology, ontological languages and formats. Then we discuss the current development and application of ontology in life science in more details.

3.1.1 What is Ontology?

Ontology is the theory of conceptualization. Intuitively, ontologies can be seen as defining the basic terms and relations of a domain of interest, as well as the rules for combining these terms and relations. In both computer science and information science, an ontology is a data model that represents a domain and is used to reason about the objects in that domain and the relations between them.

Common components of ontologies include:

- Individuals: instances or objects (the basic or "ground level" objects);
- Classes: sets, collections, concepts or types of objects;
- Attributes: properties, features, characteristics, or parameters that objects (and classes) can have;
- Relations: ways that classes and objects can be related to one another;
- Restrictions: formally stated descriptions of what must be true in order for some assertion to be accepted as input;
- Rules: statements in the form of an if-then (antecedent-consequent) sentence that describe the logical inferences that can be drawn from an assertion in a particular form;
- Axia: assertions (including rules) in a logical form that together comprise the overall theory that the ontology describes, for its domain of application;
- Events: the changing of attributes or relations (Wikipedia_ontology, 2008).

Among the above components, classes are the focus of most ontologies. Classes describe concepts in the domain. For example, in an ontology that models a news corporation, a class of employees represents all employees in the corporation. Specific employees are instances of this class. A class can have subclasses that represent concepts that are more specific than the superclass. For example, we can divide the class of all employees into reporters, columnists and editors. Attributes describe the properties of classes and instances, such as the class of editors can have the attribute ‘phone number’ whose value is numerical and the attribute ‘responsible for’ whose value is instances of the class reporters.

In practice, developing an ontology includes:

- defining classes in the ontology,
- arranging the classes in a taxonomic (subclass–superclass) hierarchy,
- defining attributes slots and describing allowed values for these slots,
- filling in the values for attribute slots for instances.

We can then create a knowledge base by defining individual instances of these classes filling in specific slot value information and additional slot restrictions with a certain form of logic (Noy and McGuinness, 2001).

There are many reasons to develop ontologies. Most importantly, ontologies can provide a formal means of representing heterogeneous information. Moreover, ontologies can be used to make the content in information sources explicit and serves as an index to a repository of information databases. This is the key to access and integrate distributed databases. Also, since ontologies explicitly define the content in information sources by formal semantics, they enable the basis of interoperability between knowledge-based 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. Finally, ontologies support automatic discovery and constraint reasoning by combining ontologies with formal logics. Ontologies provide the benefits of reuse, sharing and portability of knowledge across platform. Ontology technology therefore is competent to engage in the web-based distributed development of biological knowledge bases while providing the facilities of reasoning.

3.1.2 Ontology Representation Levels

Ontologies are generally constructed in frame-based knowledge representations and are commonly combined with either description logic or first-order logic. Also, an ontology together with a set of individual instances of classes constitutes a knowledge base, i.e. ontologies provide the basic structure around which knowledge bases can be built.

In knowledge representation, a frame is a data structure similar to the object-oriented paradigm, which represents classes (called frames) with certain properties called attributes or slots. Slots may contain values, refer to other frames (relations) or contain methods. Frames are thus a machine-usable formalization of concepts or schemata and are conceptual containers for meta-knowledge defining a given entity in the domain of discourse (Wikipedia_frame, 2007).

Class level and class properties level of ontologies (as the first two tiers shown in Figure 3.2) are the foundation of building higher levels of knowledge representation and can be developed relatively independently from other levels. Ontologies that only contain these two levels are called upper ontologies. An upper ontology (or foundation ontology) is a model of the common objects that are generally applicable across a wide range of domain ontologies. It contains a core glossary in whose terms

objects in a set of domains can be described. Such ontologies include Dublin Core (DCMI, 1995-2008), WordNet (Princeton, 2006), GO (OBO, 1999-2008), etc. This type of ontologies is constructed mainly as taxonomies and controlled vocabularies that are shared among users to build higher level of knowledge representations in a certain domain. Great effort has been made especially to develop biological upper ontologies in order to share consensus definitions of biological entities in life science community. Detailed discussion will be given later in section 3.1.5.

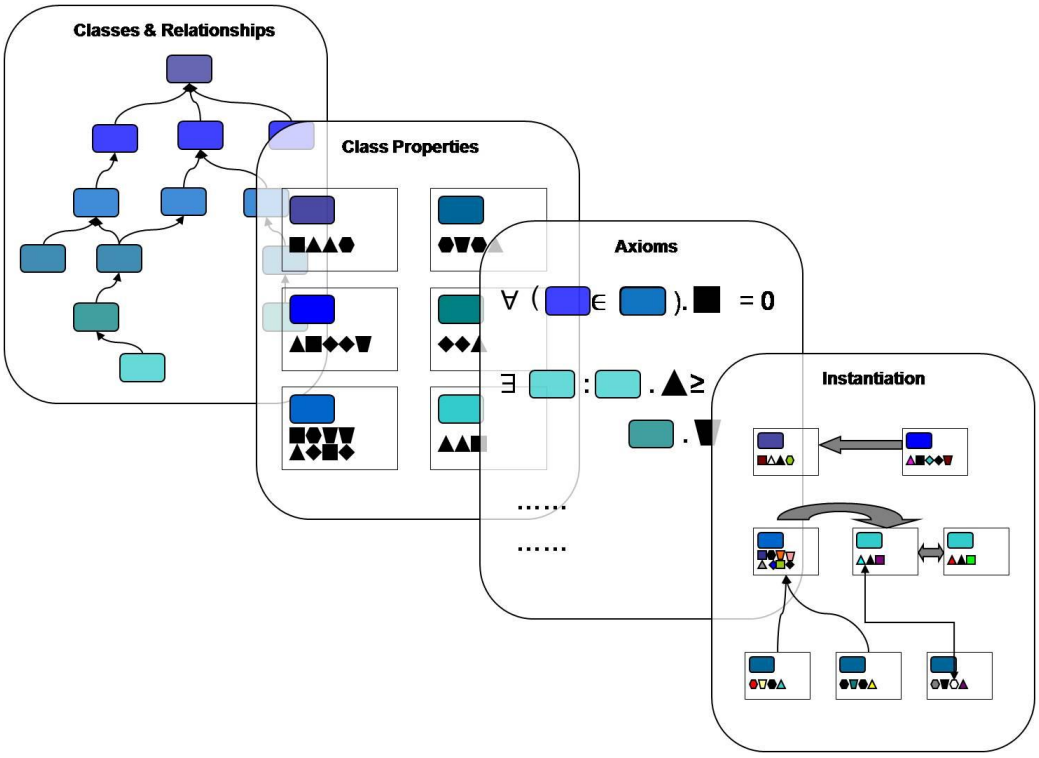


Figure 3.2 Ontology representation levels in frame-based knowledge representation

In order to model complex systems, knowledge bases are necessary when one builds a model to solve problems in the real world. This needs to define logical assertions as well as instantiations of classes in order to model the complete knowledge of a domain. For example, an ontology can be used to capture the important entities and relationships in a biological relational database, and then specify logical constraints on the entities and relationships. By doing so, one can make use of this specific ontology to detect inconsistent queries made on the biological database. More complicated knowledge bases can be constructed as ontologies in order to represent biological systems and mathematical models underlying these systems.

Ontologies with different knowledge representation (KR) levels have different requirements on the expressivity of the KR language. Also, ontology builders always need to make a decision on the trade-off between the expressivity of KR and the cost of computation. Therefore, many ontology languages have been developed for various needs. In the following section, we will discuss the development of ontology languages in detail.

3.1.3 Ontology Languages

Ontology languages are formal languages used to construct ontologies. They are designed to encode knowledge about specific domains and often include reasoning rules that support the processing of that knowledge. Ontology languages are mostly generalizations of frame languages, and are commonly based on formal logics, i.e. either first-order logic or on description logic.

In the early stage of ontology language developments, many standards have been proposed. Traditional ontology languages include Knowledge Interchange Format (KIF) as part of DARPA Knowledge Sharing Effort (Standford, 1994), Frame Logic (F-Logic) supporting predicate calculus (Kifer et al., 1995), CycL by Doug Lenat's Cyc artificial intelligence project (Lenat, 1996) which is a frame language based on first-order logic, and so on. These early ontology languages are all succeeded in certain aspects in ontology engineering, however a canonical standard of ontology format is lacking.

In order to fit the Semantic Web vision, current ontology languages have been proposed to encode ontologies based on the eXtensible Markup Language (XML). XML (W3C, 1996-2003) associates descriptive and hierarchically-structured tags with data values, in order to give semantic information to parse data in a meaningful way. With XML-based ontologies, information can be better understood by computer applications as well as human. Despite its machine processability, the nature of XML is syntactic and document-centric. This limits its ability to achieve the level of semantic interoperability required by the dynamic and integrated bioinformatics applications.

In order to improve the abilities of XML-based ontological language, the Resource Description Framework (RDF) has been developed. RDF(W3C, 1997-2001) offers a

more useful semantic model based on the directed acyclic graph structure. RDF essentially is a modeling language for defining statements about resources and relationships among them. Such resources and relationships are identified using the systems of Uniform Resource Identifiers (URIs)(W3C, 2001a). Each RDF statement is a triplet with a subject, property, and property value. Some biomedical ontologies such as the Gene Ontology(OBO, 1999-2008), UniProt (The UniProt, 2008), and the NCI thesaurus (Golbeck et al., 2003) have already been made available in RDF format. While RDF is a commonly-used Semantic Web standard, it is not expressive enough to support formal knowledge representation that is intended for processing by computers. Such a representation should consist of explicit objects and of assertions or claims about them, which enables computers to draw conclusions directly.

For this reason, more sophisticated ontological languages such as the Web Ontology Language (OWL)(McGuinness and Harmelen, 2004) have been developed. OWL is a vocabulary extension of the Resource Description Framework (RDF) and is derived from the DAML(DARPA, 2000)+OIL(Fensel et al., 2001) Web Ontology. OWL has three species, i.e. OWL Full, OWL DL, and OWL Lite. OWL Full is First-Order Logic (FOL)-based and emphasis on high expressivity of knowledge representation. On the other hand, OWL Lite and OWL DL are developed focused on minimizing the cost of computation. OWL DL is based on description logics (DL), which are a family of class-based (concept-based) knowledge representation formalisms. OWL was developed mainly because it has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S, and thus OWL goes beyond these languages in its ability to represent machine interpretable content on the web. In the life science domain, there are also some efforts applying OWL as ontological representation for encoding biological information. Such efforts include the biological pathway exchange standard called BioPAX(Hogue et al., 2002), pathway databases like HumanCyc(Romero et al., 2004) and Reactome(Vastrik et al., 2007).

In 2000, Tim Berners-Lee proposed the information architecture of Semantic Web (as shown in Figure 3.3) It indicates that ontology plays a role as the core of semantic information and the foundation of enabling reasoning services. In this vision, RDF and OWL based ontology languages are in the core of the architecture.

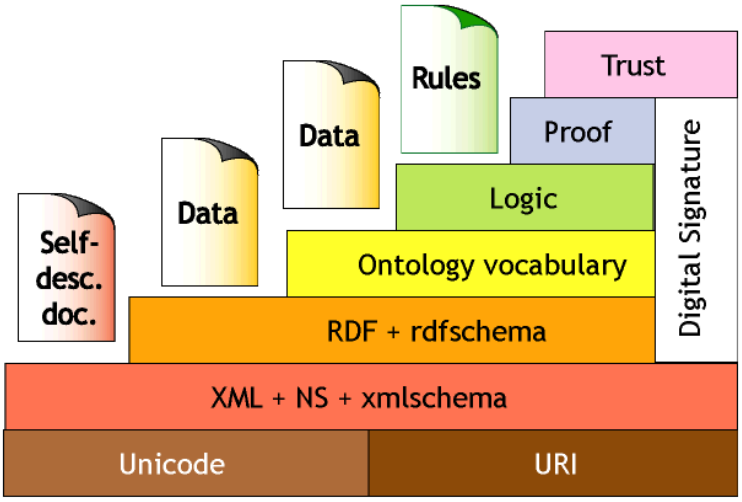


Figure 3.3 Berners-Lee's Architecture of Semantic Web on XML (Berners-Lee, 2000)

Besides the applications in knowledge representation, ontologies are also used in the discipline of software engineering. For example, Unified Modeling Language (UML) (OMG, 1997-2008) class diagrams can be seen as ontologies since they capture important classes, relationships and attributes in the application domain. Automated code generators can be used to create interfaces and implementation classes in object-oriented programming language, based on this kind of ontologies. The direct mapping between ontology and object models in programming is proven to be very useful for automated generation of biological simulations from ontology biological knowledge bases.

In the following section, we will review the existing efforts on using ontology-based technologies for the purpose of facilitating various tasks in life sciences, including knowledge representation and biological applications development.

3.1.4 Current Development of Ontology in Life Sciences

Ontology provides a shared framework of the common understanding of specific domains that can be communicated between people and application systems. Systems biology is a knowledge-based discipline, and its advance is dependent on access to distributed and heterogeneous knowledge. Therefore, ontological techniques have been extended to the development of systems biology.

Being a well-defined way of formal knowledge representation, ontology can provide a shared language for communicating biological information. It can help integrating

biological knowledge and enhancing experimenter-modeler collaboration. As the rigorous semantic descriptions of the entities and relationships between biological entities, ontologies can be used to formulate hypotheses about and navigate through the volumes of experimental data. Ontologies can also be used for rich annotation of data and the means to share and integrate data. In addition to the annotation of experimental data for search-and-retrieval in a collaborative computational environment, semantic modeling could also be applied to enhance biological problem solving and phenomenon discovery. In other words, when joined with well defined problem-solving methods, ontologies could provide convenient formalisms for modeling and for implementing solutions to application tasks in the research of systems biology.

In the following, we review the existing efforts that aim at developing ontologies for knowledge representation in life sciences as well as for building knowledge-based biological applications. On the knowledge representation aspect, we divide our review into two parts, ontology as shared biological vocabularies and ontology as biological knowledge base models.

Ontology as Shared Biological Vocabularies

In practice, biological ontologies have often started out as biological taxonomies and controlled vocabularies for biological knowledge. This allows the ontology builders to focus on the gathering of knowledge and the agreements upon definitions of biological terms and relationships.

Many projects have been launched in the past decade to develop shared ontologies across different biological domains, including the Gene Ontology (GO), SNOMED-Clinical Terms (NHS, 2007), the Unified Medical Language System (UMLS) (NLM, 1999-2008), the Foundational Model of Anatomy (FMA) (SIG, 2002-2008), the National Cancer Institute (NCI) Thesaurus (Golbeck et al., 2003), and so on. These projects are all part of a grand effort to create a unified biomedical informatics framework for integrating disparate biological information, which uses ontologies as its foundational layer. These shared ontologies are designed to meet the growing need for comprehensive and shared terminology, in order to provide a model of how biological information relates to each other. Among these efforts, the Gene Ontology (GO) provides the vocabulary for the description of many biological concepts such

as the annotation of the molecular function, biological process, and cellular component of gene products. Therefore, GO ontologies are used as a de facto standard for semantically annotating biological information including experimental data sources and biological models.

These developed biological ontologies for shared vocabularies are used for data source annotation, ontology-based search, data integration among biological databases, and community reference of general biological knowledge. Many biological data sources use ontologies for annotation of their data entries (e.g. BLAST2GO, GOFigure, Gotcha, etc.). Some systems have been constructed to use GO annotations to compute semantic similarities between entries in biological data sources (e.g. FUSSiMeG). There are also many GO annotation tools that interpret gene expression analysis on multiple genes (e.g. FatiGO, Onto-Compare and GOstat). Ontologies are also used in ontology-based search, where users can browse biological ontologies and the terms in such ontologies as query terms. For instance, GOFish, TAIR Keyword Browser, an MGI GO Browser use GO ontology to access and query various biological databases. Further, GO ontology is used to index PubMed, a literature archive for biomedical journals, in tools such as GOPubMed and MeSH for accessing literatures (Baker and Cheung, 2007). Using ontology as shared vocabularies in biological community is one of the most active areas in ontology engineering for the life sciences.

Ontologies as knowledge base models to represent biological and modeling knowledge

By using the latest ontological languages such as OWL, ontologies can be populated with logical assertions and instantiated classes in specific knowledge domain. The instantiated ontologies then become knowledge base models.

Currently, only a few ontological languages are developed for creating biological knowledge base models. Representatives of such efforts are the Systems Biology Markup Language (SBML) (Finney and Hucka, 2003) and the Cell Markup Language (CellML) (Cuellar et al., 2003). 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. SBML and

CellML models can be annotated by using shared ontologies such as GO, which then makes the content of these models explicit and can be indexed to model repositories. Both SBML and CellML have gained support from systems biology community and hundreds of software systems have been developed applying these standards.

Despite the fact that both CellML and SBML can be seen as ontology-based models for representing biological modeling knowledge, neither specifications contain information about model parameterization or associated rationale, nor do they define a way to connect modeling information with experimental data sources. Further, although both specifications are developed in XML and share characteristics with ontology models, they don't possess the facility of representing formal logic between biological entities and therefore don't fit in the vision of semantic web. Therefore, in our work we propose using OWL as the format to represent biological models to replace CellML and SBML as the representation standard for creating knowledge bases of biological modeling.

Ontology for automated transformation from biological knowledge-based models to simulations

Unlike data models, the fundamental asset of ontologies is their independence from any particular applications, i.e. an ontology consists of generic knowledge that can be reused among different users and computer software. For this reason, knowledge-based ontology models can be constructed independently from knowledge-centered applications. For example, simulation software that is built on a certain biological model can be modularized into a general computational engine and a biological knowledge-based model. In this way, the knowledge-based model can be modified without rewriting the computational engine and they can be used to generate the updated simulation software directly. This independence from application provided by ontology gives great advantage to the development of biological applications. The interface between applications and ontology is often achieved by wrapping or transforming knowledge-based ontology models into formats that are compatible with the targeted simulation software. The wrapping or transformation can be achieved by using XML Stylesheet Transformation (XSLT) (W3C, 1999).

Due to the lack of a fully-fledged ontology language constructed for biological knowledge bases, the effort related to this area is minimal. Only a few projects are

involved in software engineering for automated code generation from ontology-based biological models to computational simulations. One good example of such effort is Cell Electrophysiology Simulation Environment (CESE) (Missan and McDonald, 2005). In CESE project, attempt has been made on the direct transformation from CellML-based biological models to JavaBeans programs that can be used by CESE computing platform.

3.2 Agent-based Systems and Web Service Infrastructure

Systems biology research community is a highly distributed environment, in which information exists as individual pieces of personal knowledge with various versions and configurations. Different versions of knowledge may contain various errors, inaccuracies, ambiguities, inconsistencies, and redundancies; hence personal knowledge need to be brought together for debate and ultimately raise consensual knowledge accepted by the whole community. There are many kinds of ontologies that represent different levels and aspects of biological knowledge. Many existing efforts have been focused on creating consensus ontologies of biological terminology. This kind of ontologies can be seen as ‘accepted by the whole community’ in a way that these ontologies can be used as common building blocks to create or index knowledge presentation with more complexity. The shared ground level ontologies, such as Gene Ontology, support the possibility of entity evolving through time. Outdated entities are marked as ‘obsolete’ and remain in the repositories as references.

While originally computers and web-based systems were mostly used to store and retrieve information from biological databases, in systems biology information systems are desired to be able to compose and carry out complex workflows across multiple distributed biological resources through Internet. This means the knowledge sharing infrastructure is required to perform a certain level of cognitive processes and has reasoning capabilities based on domain-specific knowledge. This in turn requires intelligent and distributed software to facilitate such processes. The candidates that meet such requirements in the context of the Semantic Web vision are agent-based systems and Web Service infrastructure. In this section we introduce key concepts of agent and Web Service technologies, specifically those that can play

roles in facilitating modeling in systems biology. We then review the existing effort that uses agent-based technologies and Web Services infrastructure in life sciences.

3.2.1 Agent-based Systems

As described in Agent-oriented Programming, which is a programming paradigm promoting a social view of computing, an agent is any entity whose state is viewed as consisting of mental components (e.g. beliefs, capabilities, choices, and commitments). Agent-hood is in the mind of the programmer and anything can be viewed as having mental states, anything can be viewed as an agent. In the context of computing, an agent can be defined as a relatively independent application or computational function set that can deal with certain problems (for example automated inferencing, data visualization, responding to users' inquiry, and so on) in a certain knowledge domain (Shoham, 1994).

Agent technology emphasises the use of autonomous software entities with the ability to interoperate with other such software entities, in a uniform and standardised way. Because semantic heterogeneity is a fundamental part of interoperability, agent systems used ontologies from the beginning. For agents to be able to work together, they must communicate with each other. In a distributed environment, agents use ontologies to establish communication at the knowledge level using specific languages and protocols. The knowledge stored in agents can be encoded with ontology language such as OWL or KIF (Standford, 1994), and then enveloped with communication parameters (such as info of senders) as agent communication language (ACL) (FIPA, 2000). Ontologies are explicit representations of the agents' commitments to a model of the relevant world; hence they enable knowledge sharing and reuse. In the context of multi-agent systems, ontology is a computer-readable description of knowledge about the resources in an enterprise's network. The software agents become intelligent because they can make use of the knowledge contained in ontology to use in the process of negotiation and decision-making.

Knowledge is attributed to agents by observing their actions; an agent "knows" something if it acts as if it had the information and is acting rationally to achieve its goals. The "actions" of agents, including knowledge base servers and knowledge-based systems, can be seen through a 'tell' and 'ask' functional interface, where a client interacts with an agent by making logical assertions (tell), and posing queries

(ask). The ability to exchange information with other agents enables them to work together, share their knowledge, and achieve goals that no single agent could solve. Systems with cooperating agents are called multi-agent systems (MAS). A MAS is naturally distributed. Multi-agent systems offer location transparency by providing facilities for service discovery and brokering. In addition, a high-level communication language enables flexible and advanced communication between distributed agents.

Life science research community is a highly distributed environment, which is composed of diversified members in terms of their distinct knowledge domains and applied approaches. By regarding the experts (both human experts and computer-based experts systems) as agents, the multidisciplinary study for life sciences can be viewed as multi-agent collaboration. It could be beneficial to take an agent-oriented view, if we intent to model the behaviors of the practitioners in a unified manner (Baker and Cheung, 2007).

Agent-based Systems in the Life Sciences

Agents have been designed with the intention of information exchange between data sources. Sharing information is a major part of system integration, and thus agents naturally cover the fundamental aspects of integration. In the life sciences, a number of agent-based systems have been targeting the data integration problem.

One of the early efforts in using agent-based system in biomedical research took place as part of the BioMAS project (Decker et al., 2002). BioMAS is a genomic annotation and information gathering system, which uses the RETSINA multi-agent organization (Sycara and Pannu, 1998) and includes information extraction agents, task agents, and interface agents. Example information extraction agents include wrappers for BLAST services at Genbank (Madden et al., 1996), access to the human-annotated part of SwissProt at the EBI (Bairoch, 2000). The agent-based system of BioMAS proved useful with respect to dealing with dynamic information over time. While systems such as BioMAS are more oriented towards the integration of biological databases, other agent systems have been developed for simulation of biological processes. For example, Infectious Disease Epidemic Simulation System (IDESS) has been developed to study the outbreak of an infectious disease in any geographic region (Yergens et al., 2006).

There are other agent systems that integrate bioinformatics resources, such as GeneWeaver (Bryson et al., 2000), and myGrid project (Gibson et al., 2007) has many agent-like features. OntoFusion (Perez-Rey et al., 2006) is also an example of using multi-agent system for biological database integration, where database agents that act as wrappers are used to hide the actual database access procedures from the rest of the system. Such wrapper agents were created for public Web-based DBs, as well as for private DBs that are accessible through ODBC or JDBC. The mediator module is able to divide and propagate user queries through the agent society. It collects and merges the results of these queries and sends them back to the user interface.

Agent technology has been successfully applied in the past to system integration. However, in bioinformatics systems it has mainly been used for enhanced automation and thus so far only a couple of bioinformatics integration systems are based on agent technology. In recent development, attention has been shifted from agent systems to service-oriented approaches, which will be discussed in the following section.

3.2.2 Web Service Infrastructure in the Life Sciences

In the Semantic Web, service-oriented computing is a software architecture that allows information resources to be presented as platform-independent, self-describing, modular software units. Web Services technology therefore provides a web-based infrastructure for general knowledge sharing and reuse (Baker and Cheung, 2007).

Web Services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web Services can therefore:

- Transform personal knowledge into light-weight, flexible, and easy-accessible units;
- Composite new functionality through the use of loosely coupled reusable software components;
- Decompose and distribute large-scale tasks into component tasks executed simultaneously across many platforms

Web Service definition encompasses many different systems, but in common usage the term refers to clients and servers that communicate using XML messages that

follow the SOAP standard (W3C, 2000). Common in both the field and the terminology is the assumption that there is also a machine readable description of the operations supported by the server written in the Web Services Description Language (WSDL) (W3C, 2001b). The latter is not a requirement of a SOAP endpoint, but it is a prerequisite for automated client-side code generation in many Java and .NET SOAP frameworks. Some industry organizations, such as the WS-I, mandate both SOAP and WSDL in their definition of a Web Service.

As defined by W3C, a Web Service is an abstract notion that must be implemented by a concrete agent. The agent is the concrete piece of software or hardware that sends and receives messages, while the service is the resource characterized by the abstract set of functionality that is provided. To illustrate this distinction, one might implement a particular Web Service using one agent and a different agent with the same functionality. Although the agent may have changed, the Web Service remains the same. Agents encompass notions of communication negotiation, argumentation, auctions commitment, coalitions, communities evolving behavior and adaptation autonomous behavior. On the other hand, Web Service technology is focused on the functionality aspects rather than achieving software behavioral autonomy.

Using ontologies to annotate services has also been addressed by many initiatives, including WSDL-S (W3C, 2005), SAWSDL (W3C, 2006) and Semantic Markup for Web Services (OWL-S) (W3C, 2004) under the W3C recommendation. This is intended to provide reuse, discovery, and composition abilities to Web Services.

Web Service infrastructure in the Life Sciences

Web Services provide a standard way of publishing applications and data sources over the internet, enabling mass dissemination of knowledge. In the life sciences, the web-service approach is seen as being a road to standardizing the multitude of tools available from different providers.

There are existing 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 858 databases and 166 web servers available in molecular biology. 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 (Baker and Cheung, 2007).

In the life sciences domain, the increasing number of bioinformatics Web Services requires a common framework that allows researcher to search for Web Services according to the required functionality, input and output, pre and post conditions and to combine these Web Services into integrated processes. There are also efforts that aim at building such computing frameworks to bring biological Web Services together. For example, the MOBY-S system by BioMoby project defines an ontology-based messaging standard through which a client will be able to automatically discover and interact with task-appropriate biological data and analytical service providers. BioMOBY describes bioinformatics services as Web Services and provides a language to describe biological services in terms of their inputs and outputs, as well as a central registry, called 'MOBY Central,' to enable service registration and discovery (Nagarajan et al., 2006).

myGrid is another good example of such effort, which provides similar functionality and is based on the Open Grid Services Architecture (OGSA) (Goble, 2005). myGrid is an open source project developing a suite of software components that application developers and scientists can use for building and running *in silico* experiments. Over 3000 distributed services including remote Web Services, local scientist-specific Java applications, and simple scripts from over the internet are accessible through myGrid's software suite. myGrid uses Taverna workbench (myGrid, 2005) workflow engine to allow users to find and run workflows among biological Web Services. In addition, it offers the users the ability to use or create their own workflows. A workflow is a sequence of services executed in the correct order that combined can provide higher-level services. Workflows can be thought of as pre-generated static plans, as opposed to the (query) plans generated dynamically by mediators (Goble, 2005).

In latest development, Web Services have been combined with formal semantics that allow describing interaction and workflows among them by means of a semantic

markup language, and therefore are able to meet the need of modeling knowledge collaboration among systems biology practitioners.

3.3 Summary

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 to them.

The Semantic Web vision promises transparent search, manipulation, and integration of information to researchers in systems biology by an interconnected set of technologies. Among these technologies, ontology plays a key role to adapt the Semantic Web infrastructure to facilitate biomedical research and has been proved very useful to facilitate most of the processes in modeling in systems biology.

As discussed above, the current application of the Semantic Web in biological modeling is still preliminary. We have found out that the main application for ontologies in the life sciences is currently focused on sharing biology terminologies. The Web Services used in the biological studies are limited and unorganized. We believe that there are much more benefits to be had by using ontology along with Semantic Web Services to supporting modeling tasks in systems biology.

In the following chapters, we will introduce several case studies of biological modeling and identify the knowledge components involved in the modeling processes that can be standardized to the Semantic Web infrastructure.

Chapter 4 : Cases of Biological Modeling

In this chapter, we describe three cases of modeling in biological research. The first one is the model for describing how action potentials in neurons are initiated and propagated in the giant nerve fibre of a squid developed by Hodgkin and Huxley. The second one is the model for electrical resonance phenomenon in hair cells in vertebrate hearing systems developed by Lewis and Hudspeth. The third one is a composite model for describing calcium oscillation induced by hormones.

We chose these cases because they encompass all the major processes involved in the study of systems biology including data acquisition from experiments, abstract modeling, raw data analysis for parameterization, and creation of simulations. To construct this kind of model requires reusing modeling knowledge (the second case reuses the scheme of the first case), and integrating fairly complex sub-models of different types, e.g. in both the second and the third case. This will be sufficient to demonstrate the complexity of the modeling tasks in systems biology. These three cases will also be used to validate the usefulness of our proposed semantic web modeling framework in the succeeding chapter.

We choose Hodgkin and Huxley as the first case (Hodgkin and Huxley, 1952). It has been used as the basis of ionic current models, which includes the Lewis and Hudspeth modeling case. We will describe its biological background and the modeling processes.

In the second case study, we will go into more detail of the whole workflow of biological modeling. We will introduce Lewis & Hudspeth case (Hudspeth and Lewis, 1988), which describes the biological background of the electrical resonance phenomenon in hair cells in vertebrate hearing systems. Then we will describe the approaches used for acquiring experimental data for quantitative investigation. We describe the model construction, which was developed by Hudspeth and Lewis for understanding electrical resonance in bull-frog hair cells. And then we give an example of computational simulation for such model.

In the third case study, we will introduce the model for describing calcium oscillation induced by hormones. This model is composed from several stand-alone modeling

cases including hormone-induced G-protein activation (Riccobene et al., 1999), G-protein initiated $InsP_3$ release (Sneyd et al., 2004), kinetics of $InsP_3$ receptor on Endoplasmic Reticulum (Sneyd and Dufour, 2002), and a minimum model for cytosolic calcium oscillation (Hofer, 1999). This case is important to investigate the process of model integration.

4.1 Hodgkin & Huxley Case

In this section, we introduce the effort of biological modeling by AL Hodgkin and AF Huxley. In 1952, Hodgkin and Huxley published a quantitative model for describing how action potentials in neurons are initiated and propagated. This model was developed with the results of a series of experiments in which Hodgkin and Huxley investigated the flow of electric current through the surface membrane of the giant nerve fibre of a squid (Hodgkin and Huxley, 1952). In this model, a mathematical description for the ion conductances and excitation of the neuron fibre was given. The Hodgkin and Huxley model is regarded as one of the great achievements of 20th-century biophysics. They therefore were awarded the Nobel Prize in 1963 in Physiology or Medicine for this work. The developed quantitative model has then been used as the basis for almost all other ionic current models of excitable tissues followed.

4.1.1 Biological background

Neurons (also known as nerve cells) are electrically excitable cells in the nervous system processing and transmitting information to sensory organs, such as brain and spinal cord. Neurons are typically composed of a cell body, dendritic trees and an axon. The majority of vertebrate neurons receives input on the cell body and dendritic tree, and transmits output via the axon. Neurons communicate by chemical and electrical synapses, in a process called synaptic transmission. The fundamental process that triggers synaptic transmission is called the action potential, an electrical signal that is generated by the excitation of the neuron cell body and then propagates along the axon.

Neurons, like all cells, maintain different concentrations of certain ions across their cell membranes. The neuronal membrane contains specialized proteins called ion

channels, which form pores in the membrane that are selectively permeable to particular ions. Sodium channels allow sodium ions through the membrane, while potassium channels allow potassium ions through. Under resting conditions, the potassium channel is more permeable to potassium ions than the sodium channel is to sodium ions. Hence there is a high concentration of sodium ions present outside the neuron, and a low concentration of potassium ions inside. The membrane has a charge on the inside face that is negative relative to the outside, as more positively charged ions flow out of the neuron than flow in. The resting voltage across the membrane is typically −70 millivolts (mV).

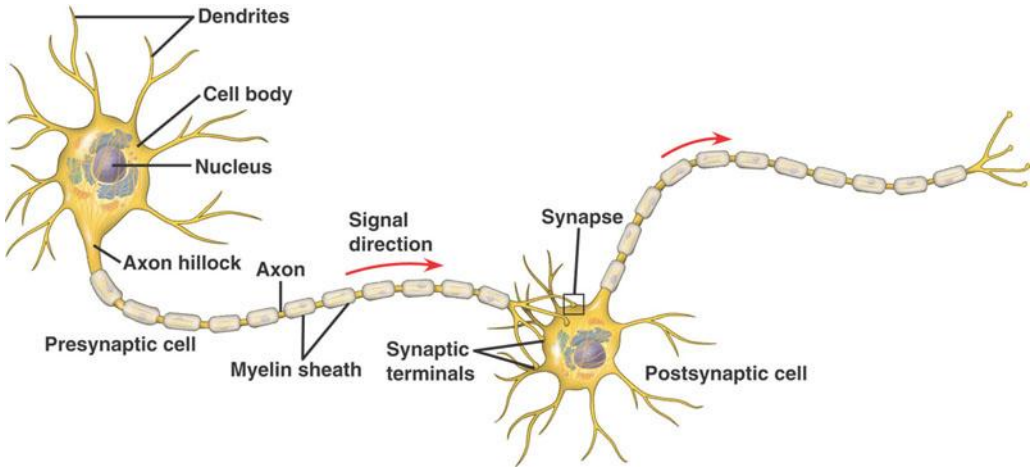


Figure 4.1 Structure of vertebrate neurons

In Figure 4.1 neurons are sheathed in myelin, which is formed by either of two types of glial cells: Schwann cells ensheathing peripheral neurons and oligodendrocytes insulating those of the central nervous system. Along myelinated nerve fibers, gaps in the sheath known as nodes of Ranvier occur at evenly-spaced intervals, enabling an especially rapid mode of electrical impulse propagation called saltation. (Image source: http://kvhs.nbed.nb.ca/)

In neurophysiology, an action potential (also known as a nerve impulse) is a collection of membrane potential changes which occur during nerve impulse propagation that travels along several neurons. The propagation of a nerve impulse along an axon begins when the neuron synapses receives neurotransmitters from nerve endings nearby. The neuron then increases its internal potential, setting off a chain of events as the nerve impulse runs down the axon. The action potential moves rapidly down the axon, with a conduction velocity as high as about 100 meters/second. Because they are able to transmit information so fast, the flow of

action potentials is a very efficient way of information transmission, considering that each neuron the signal passes through can be up to a meter in length. Action potential also appears in other types of excitable cells, such as muscle cells and plant cells.

Action potential can be divided into the following phases:

Depolarization: Initially the membrane potential is resting at around -70 mV. When the neuron synapses receives neurotransmitters from nerve endings, the cell becomes excited. Voltage-gated sodium (Na^+) channels open when the membrane potential rises about 20 mV above the rest potential to -50 mV. This potential is called the "threshold". The membrane permeability to Na^+ is suddenly increased. Na^+ then rushes into the cell (about 1 millisecond). This will cause the membrane to "fire", initiating a positive feedback that causes the voltage inside the axon to suddenly and rapidly become more positive. The membrane potential continues to rise above the “threshold” and keeps the channels open until the inside of membrane becomes positively charged till about +40 mV. Because of the positive feedback, an action potential is all-or-none; there are no partial action potentials.

Repolarization: During this rapid depolarization, a large influx of Na^+ causes the immediate opening of voltage-activated Potassium (K^+) Channels and leads to a rapid efflux of K^+ from the neuron. At the same time, Na^+ channels responsible for the initial inward current are inactivated. The Na^+ channels close when the voltage peaks. The membrane voltage is then restored to its resting value by the combination of these two effects until undershoots (hyperpolarization) slightly and persists until the membrane K^+ permeability returns to its usual value.

Refractory Period: The Na^+/ K^+ Channels actively pump Na^+ out of the neuron and K^+ into the neuron. This re-establishes the initial ion distribution of the resting neuron. During this time (~ 1 millisecond), the neuron firing “threshold” is much higher than normal, making the Na^+ and K^+ Channels more difficult to open, and thus inhibiting another action potential at the same spot. Under this condition, an axon is said to be refractory.

The cycle of depolarization and repolarization is extremely rapid, taking only about 2 milliseconds and thus allows neurons to fire action potentials in rapid bursts, a common feature in neuronal communication. The action potential "travels" along the

axon without fading out because the signal is regenerated at each patch of cell membrane. This happens because an action potential at one patch raises the voltage at nearby patches, depolarizing them and provoking a new action potential there. In unmyelinated neurons, the patches are adjacent, but in myelinated neurons, the action potential "jumps" between distant patches, making the process faster and more efficient (Hodgkin and Huxley, 1952).

4.1.2 Mathematical modeling

Mathematical and computational models are essential for understanding the action potential and allowing simulations or even predictions to test against experimental data. Therefore Hodgkin and Huxley developed accurate model to describe the voltages and currents of the action potential in all of its phases.

In Hodgkin and Huxley model, they developed an electric circuit network to match how the squid axon carries an action potential across a patch of membrane (See Figure 4.2). The current flow across the cell membrane depends on the capacitance of the membrane and the resistance/conductance (conductance is the inverse of resistance) of the ion channels. The total ionic current is represented by the sum of the sodium current, potassium current and a small leakage current. The leakage current represents the collective contribution of ions such as chloride and bicarbonate etc. The lipid bi-layer membrane is represented as a capacitance (C_m). Voltage-gated ion channels are represented by nonlinear electrical conductances (g_n, where n is the specific ion channel), i.e. the conductance is both voltage and time-dependent. This was then mediated by voltage-gated cation channel proteins, each of which has an open probability that is voltage-dependent. Leak channels are represented by linear conductances (g_L). The electrochemical gradients driving the flow of ions are represented by batteries (E_n and E_{Leak}), the values of which are determined from the Nernst potential of the ionic species of interest. Finally, ion pumps are represented by current sources (I_p).

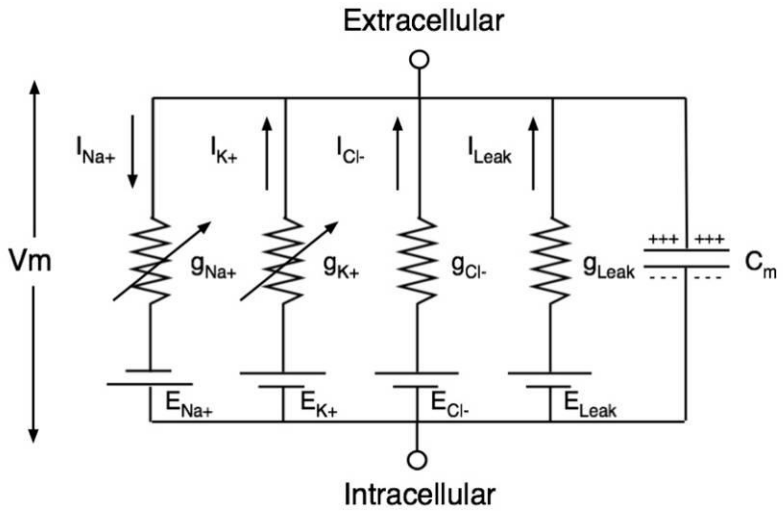


Figure 4.2 Equivalent electrical circuit for the Hodgkin–Huxley model

Figure 4.2 illustrates an equivalent electrical circuit for the Hodgkin–Huxley model of the action potential. I_m represents the current through a small patch of membrane and V_m represents the voltage across it. The C_m represents the capacitance of the membrane patch, whereas the four g's represent the conductances of four types of ions. The two conductances on the left, for potassium (K^+) and sodium (Na^+), are shown with arrows to indicate that they can vary with the applied voltage, corresponding to the voltage-sensitive ion channels. The two conductances on the right help determine the resting membrane potential. Chloride conductance is normally included in the leak conductance in Hodgkin and Huxley model.

Within the circuit network, current flowing through the membrane can be carried via the charging and discharging of a capacitor or via ions flowing through variable resistances in parallel with the capacitor. Each of the resistances corresponds to charge being carried by different components. In the nerve cell these components are Na^+ and K^+ and a small leakage current that is associated with the movement of other ions. Each current (I_{Na}, I_K, and I_{Leak}) can be determined by a driving force which is represented by a voltage difference and a permeability coefficient, which is represented by a conductance in the circuit diagram. These equations can easily be derived using Ohm's law $V=IR$:

$$I_{\mathrm{Na}} = g_{Na}(E - E_{Na}) \tag{1}$$

$$I_K = g_K(E - E_K) \tag{2}$$

$$I_{Leak} = g_{Leak}(E - E_{Leak}) \tag{3}$$

g_{Na} and g_K are both functions of time and membrane potential E. E_{Na}, E_K, E_{Leak}, Cm and g_{Leak} are all constants that are determined via experimentation.

To build their mathematical model that describes how the membrane current works during the voltage clamp experiment, Hodgkin and Huxley used the basic circuit equation

$$I = C_m dV_m / dt + I_i \tag{4}$$

where I is the total membrane current density (inward current positive), I_i is the ionic current density (inward current positive), V is the displacement of membrane potential (depolarization is negative) , C_m is the membrane capacitance, t is time. During the experiment Hodgkin and Huxley found that the ionic current when the derivative was set to zero and the capacity current when the ionic current is set to zero were similar. Therefore, they write the capacity current and ionic current in a linear relationship:

$$I = I_{Na} + I_K + I_{Leak} \tag{5}$$

where I_{Na} is the sodium current, I_K is the potassium current and I_{Leak} is the leakage current. We can further expand on this model by adding the following relationships:

$$I_{\mathrm{Na}} = g_{Na}\,(V - V_{Na}) = g_{Na}(E_{Na} - E_R) \tag{6}$$

$$I_K = g_K\,(V - V_K) = g_K(E_K - E_R) \tag{7}$$

$$I_{Leak} = g_{Leak}(V - V_{Na}) = g_{Leak}(E_{Leak} - E_R) \tag{8}$$

$$V = E - E_R \tag{9}$$

where E_R is the absolute value of resting potential, whose value is decided experimentally. V, V_{Na}, V_K, V_{Leak} can then be measured directly as displacements from the resting potential.

When examining the graph of the potassium current versus the potassium potential difference, Hodgkin and Huxley found that if g_K is used as a variable the end of the record can be fitted by a first-order equation, but a third- or fourth-order equation is needed to describe the beginning. Therefore, a simplification is made by supposing that g_K is proportional to the fourth power of a variable which obeys a first-order equation. The formal assumptions used to describe the potassium conductance are:

$$g_K = \bar{g}_K n^4 \tag{10}$$

$$\frac{dn}{dt} = \alpha_n (1-n) - \beta_n n \tag{11}$$

where $\bar{g}_K$ is a constant, and n is a dimensionless variable that varies from *0* to *1*. It is the proportion of ion channels that are open. α_n is the rate of closing of the channels and β_n is the rate of opening. Together, they give the total rate of change in the channels during an action potential.

In order to find functions connecting α_n and β_n with membrane potential, all the measurements are plotted against *V*. Empirical expressions for α_n and β_n are then used to fit the experiment data and get:

$$\alpha_n = 0.01(\mathrm{V}+10)/(\mathrm{e}^{(\mathrm{V}+10)/10} - 1) \tag{12}$$

$$\beta_n = 0.125\mathrm{e}^{\mathrm{V}/80} \tag{13}$$

The sodium conductance is described to be determined by two variables, each of which obeys a first-order equation.

$$g_{Na} = m^3 h \bar{g}_{Na} \tag{14}$$

where $\bar{g}_{Na}$ is a constant and m is the proportion of activated carrier molecules (ion channels) and *h* is the proportion of inactivated carrier molecules (ion channels). *m* and *h* can be further described by

$$\frac{dm}{dt} = \alpha_m(1-m) - \beta_m m \tag{15}$$

$$\frac{dh}{dt} = \alpha_h(1-h) - \beta_h h \tag{16}$$

where α_m and β_m, α_h and β_h are again rate constants that are similar to the rate constants for the potassium conductance. Similarly empirical expressions are fitted to the experimental data:

$$\alpha_m = 0.1(\mathrm{V}+25)/(\mathrm{e}^{(\mathrm{V}+25)/10}-1) \tag{17}$$

$$\beta_m = 4\mathrm{e}^{\mathrm{V}/18} \tag{18}$$

$$\alpha_h = 0.07\mathrm{e}^{\mathrm{V}/20} \tag{19}$$

$$\beta_h = 1/(\mathrm{e}^{(\mathrm{V}+30)/10}-1) \tag{20}$$

Finally, the Hodgkin and Huxley model for the action potential in squid giant axon can be summarized as equation (11)-(13) (15)-(20) and

$$I = C_m dV_m/dt + \bar{g}_K n^4(V-V_K) + m^3 h\bar{g}_{Na}(V-V_{Na}) + \bar{g}_{Leak}(V-V_{Leak}) \tag{21}$$

4.2 Lewis & Hudspeth Case

4.2.1 Biological Background

Ears have evolved in parallel in vertebrates since a common divergence 300 million years ago. Hearing organs in different species differ in the range of sound frequencies to which they can respond. The sensory cell on which all designs rely is the hair cell: a polarized neuroepithelial cell with sensory transduction channels at the apical end and the primary sensory synapse at the basal end. In birds, reptiles and amphibia, the hearing organ is strictly referred to as a papilla. The more familiar cochlea is the hearing organ of mammals and consists of a coiled cavity within the temporal bone on either side of the head. It contains the structures that separate the components of a complex sound by frequency and intensity. The cochlea acts as the

sensory organ for the auditory system, signalling the information along the auditory nerve to the central nervous system. The length of the cochlea is divided into three regions by a pair of membranes. The central region, termed the scala media, is separated from the scala tympani by the basilar membrane and from the scala vestibuli by Reissner's membrane. The scala media narrows toward the apical end of the cochlea, terminating just short of the tip. There is then a small opening connecting the scala vestibule and scala timpani. Figure 4.3 is a cross section through the cochlea showing the two membranes along with the organ of Corti which contains the sensory hair cells (Ashmore and Gale, 2000).

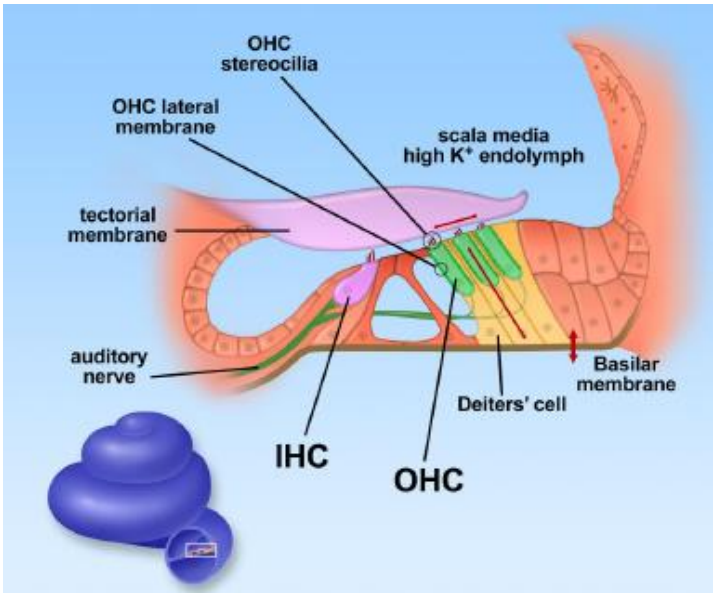


Figure 4.3 Organ of Corti

The movements of the basilar membrane (BM) and tectorial membrane (TM) up and down result in stimulation of the hair cells in the Organ of Corti. The tectorial membrane is located above the hair cells and their associated stereocilia. The stereocilia are actin-based hair-like fibers that project from the hair cell. Movement of the stereocilia causes the hair cell to be stimulated. Hair cells then transduce the movement of their hair bundles into electrical signals. Such process involves the opening of mechano-transduction channels, hair cell depolarization, and basolateral synaptic release, resulting in a modulation of the membrane potential.

In general, the bending of stereocilia allows K^+ to flow into the hair cells, which are thus depolarized. This opens voltage-gated Ca^{2+} channels. Calcium is involved in

neurotransmitter (glutamate) release and also in a K^+ active exit mechanism. A simple representation of how this happens is schematized in Figure 4.4.

A biologically important sound can contain many different frequency components and hearing organs are designed to extract information selectively about frequencies. Although the basic design of hair cell mechanoelectrical transduction is conserved across species, it is clear that different mechanisms of frequency selectivity have evolved. The hair cells of the vertebrate inner ear transduce stimuli derived from sound or acceleration into receptor potentials.

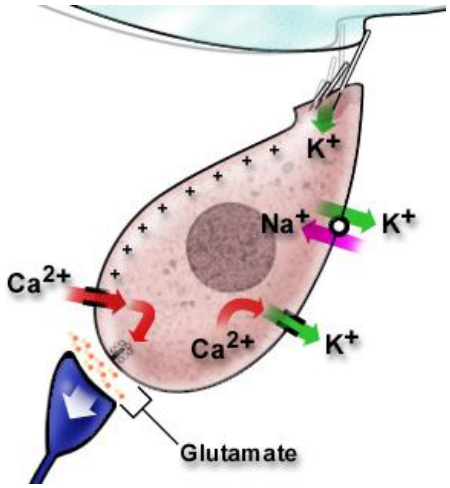


Figure 4.4 Mechanoelectrical transduction of hair cells

In some cases, the responses of hair cells are made selective for stimuli or particular frequencies through a tuning mechanisms based upon electrical resonance in their membranes. In non-mammalian species, frequency selectivity depends upon an intrinsic resonance mechanism in each hair cell. Electrical resonances in hair cells were observed first in the basilar papilla of the turtle (Crawford and Fettiplace, 1981). Then they were found in hair cells of amphibian papillae (auditory sensors) and sacculi (seismic sensors) in rain frogs (Hudspeth, 1983, Hudspeth, 1985, Pitchford and Ashmore, 1987). In each case, the resonant frequencies were within or very close to the known range of acoustical sensitivities of the sensor in question. In the turtle basilar papilla and the sacculus of the bullfrog (Rana catesbeiana), the resonances have been shown to result from an interplay between voltage-sensitive calcium channels and calcium-sensitive potassium channels in the hair cell membrane

(Hudspeth, 1983, Art and Fettiplace, 1987). The bullfrog sacculus is extraordinarily responsive to substrate vibration (seismic stimuli). In the absence of such stimuli, its afferent axons typically exhibit random spontaneous spike activity at mean rates between 10 and 40 spikes/s. Crawford and Fettiplace found that afferent axons from the turtle basilar papilla exhibited spike-interval histograms with conspicuous periodicities.

These functions of transduction, tuning, and synaptic transmission are mediated to a large extent through the activity of ion channels. The transduction channel, a mechanically gated, non-selective cation channel associated with the hair bundle, has been extensively characterized. Studies have been carried out by Lewis and Hudspeth for the voltage- and ion-dependent channels responsible for the membrane conductances: a voltage-dependent Ca^{2+} conductance, a transient, voltage-dependent, A-type K^+ conductance, and a Ca^{2+}-sensitive K^+ conductance (Hudspeth and Lewis, 1988).

4.2.2 Experimental Data Acquisition

In the case of Lewis & Hudspeth, all three types of ion channels were first studied in hair cells using tight-seal patch-clamp recording techniques on cells dissociated from the bullfrog sacculus (Hudspeth and Lewis, 1988). Much of what we know about the properties of ion channels in cell membranes has come from experiments using voltage and current clamp. In general, the method allows ion flow across a cell membrane to be measured as electric current, whilst the membrane voltage is held under experimental control with a feedback amplifier. The voltage clamp technique allows an experimenter to "clamp" the cell potential at a chosen value. This makes it possible to measure how much ionic current crosses a cell's membrane at any given voltage. This is important because many of the ion channels in the membrane of a neuron are voltage gated ion channels, which open only when the membrane voltage is within a certain range. Voltage clamp measurements of current are made possible by the near-simultaneous digital subtraction of transient capacitive currents that pass as the recording electrode and cell membrane are charged to alter the cell's potential. The method was first developed by Cole (1949) and Hodgkin *et al*. (1952) for use with the squid giant axon. Since then, many variants of the technique have evolved and voltage clamp analysis has been extended to a wide range of tissues. For

example, patch clamp recording is one of the variations. It takes place with a glass micropipette that has an open tip diameter of about one micrometre. This type of electrode is distinct from the sharp microelectrode used to impale cells in traditional intracellular recordings.

In Lewis and Hudspeth case, whole-cell recording technique is used. Whole-cell recording is a kind of patch clamp recording where the microelectrode is placed next to a cell, and gentle suction is applied through the microelectrode to draw a piece of the cell membrane into the microelectrode tip. The glass tip forms a high resistance 'seal' with the cell membrane. Then more suction is applied to rupture the portion of the cell's membrane that is inside the electrode, thus providing access to the intracellular space of the cell. The larger opening at the tip of the patch clamp electrode provides lower resistance and thus better electrical access to the inside of the cell.

By using whole-cell recording techniques, experimental 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 the Lewis & Hudspeth case, the columns are defined as the following:

SignalsExported=I_{memb},V_m

Signals=‘‘I_{memb}*”, ‘‘*V_m*”, ‘‘*I_{memb}*”, ‘‘*V_m*”, ‘‘*I_{memb}*”, ‘‘*V_m*” .....*

‘‘Time (s)”; ‘‘Trace #1 (pA)” ‘‘Trace #1 (mV)”; ‘‘Trace #2 (pA)” ‘‘Trace #2 (mV)”; ‘‘Trace #3 (pA) ” ‘‘Trace #3 (mV) ”;....

In which ‘‘*Trace#*” is used to generalize the table structure so that any signals can be stored by the table. Each ‘‘*Trace#*” is then specified according to the signal, such as membrane current (‘‘*Imemb*”) and membrane potential (‘‘V_m”).

4.2.3 Mathematical Modeling

By analyzing the acquired data, the following mathematical description of kinetics of voltage- and current-dependent conductances in hair cells has been built. The ionic

conductance model is entirely based on the Hodgkin and Huxley case (Hodgkin and Huxley, 1952).

The involvement of Ca^{2+} and Ca^{2+}-activated K^{+} conductances in electrical resonance was tested by the hair cell's expected membrane-potential changes in response to extrinsic current pulses. The total membrane current is at all times equal to the sum of capacitive and ionic currents,

$$I_m = I_{cap} + I_{Ca} + I_{K(Ca)} + I_{Leak} \tag{1}$$

in which I_m is the total membrane current, I_{cap} is capacitive current, and I_{ca}, $I_{K(Ca)}$ and I_{Leak} are respectively the Ca^{2+} and Ca^{2+}-activated K^{+}, and leakage currents. Each of these terms is a function of time. The model therefore consists of sub-models that describe Ca^{2+}current, Ca^{2+}-activated K^{+} current, and leakage currents respectively. The leakage current is relatively simple. Here we give the Ca^{2+}current and Ca^{2+}-activated K^{+} current sub-models as the following:

Ca^{2+} current (Ica) sub-model

Voltage-dependent I_{ca} activation is described by a third-order kinetic scheme (Hodgkin & Huxley, 1952),

$$I_{ca} = \bar{g}_{ca} m^3 (V_m - E_{ca}) \tag{2}$$

in which $\bar{g}_{ca}$ is the limiting value of Ca^{2+} current conductance when all Ca^{2+} channels are open, m is the time-dependent value of the activation parameter, V_m is membrane potential, and E_{ca} is the Ca^{2+} equilibrium potential. The activation parameter, m, varies between zero and unity with time,

$$\frac{dm}{dt} = \beta_m (1 - m) - \alpha_m m \tag{3}$$

in which α_m and β_m are respectively the m gate's closing and opening rate constants. For perturbation by a step change in membrane potential, equation (3) can be solved for m, yielding

$$m(t) = m_0 + (m_\infty - m_0)[1 - \exp(-\frac{t}{\tau_m})] \tag{4}$$

in which m_0 is the initial value of m before a potential change and τ_m is the time constant of m's exponential approach to m_∞, its equilibrium value at the new potential.

If $m_0=0$ at holding potentials below -60 mV, τ_m and m_∞ can be expressed in terms of α_m and β_m:

$$\tau_m = 1/(\alpha_m + \beta_m) \tag{5}$$

$$m_\infty = \beta_m /(\alpha_m + \beta_m) \tag{6}$$

α_m and β_m can, in turn, be described as empirical functions of potential:

$$\alpha_m = \alpha_0 \exp[-(V_m + V_0 / V_A)] + K_A \tag{7}$$

$$\beta_m = \beta_0 \exp[(V_m + V_0) / V_B] + K_B \tag{8}$$

in which α_0, β_0, K_A, K_B are rate constants and V_0, V_A, V_B are potentials to be determined experimentally.

Regulation of intracellular Ca^{2+}: A relatively simple scheme has been adopted here which was used to describe the inactivation of I_{ca} caused by Ca^{2+} accumulation in insect skeletal muscle fibres. This Ca^{2+}-regulation scheme involves some basic assumptions. First, Ca^{2+} entering the cell binds fraction, U, of total Ca^{2+} remains free at any given time. Secondly, the binding of that Ca^{2+} accumulates next to the membrane in a small fraction, ξ of the cell's total volume. Thirdly, Ca^{2+} leaves this submembrane compartment at a rate proportional to its free concentration there; the rate constant for this process is Ks, This scheme predicts that, at any given time, the submembrane Ca^{2+} concentration, $[Ca^{2+}]_i$ changes at a rate

$$\frac{d[Ca^{2+}]_i}{dt} = UI_{ca} /(zFC_{vol}\xi) - K_s[Ca^{2+}]_i \tag{9}$$

Ca^{2+}-activated K^+ current ($I_{K(Ca)}$) sub-model

First the gating of the Ca^{2+}-activated K^+ channels in the hair cell is described by a linear, five-state kinetic scheme adapted from earlier models by Magleby & Pallotta, 1983; Moczydlowski & Latorre, 1983. Each transition, other than the open-closed transition, is characterized by a Ca^{2+}-dissociation constant, K_j. The voltage

dependence of these transitions derives from the assumption that Ca^{2+} binds to a site within the transmembrane electric field, making the effective dissociation constant a function of membrane potential,

$$K_j(V_m) = K_j(0)\exp[\delta_j zFV_m/(RT)] \tag{10}$$

$K_j(0)$ is the dissociation constant of the jth binding site, and z, F, R and T have their usual meanings. These dissociation constants are related to the forward and reverse rate constants k_j and k_{-j} by

$$K_j = k_{-j}/k_j \tag{11}$$

The closing rate constant α_c is also expressed as a function of membrane potential

$$\alpha_c = \alpha_c(0)\exp(-V_m/V_A) \tag{12}$$

$\alpha_c(0)$ is the closing rate constant at 0 mV membrane potential, V_m is membrane potential, and V_A is a potential used to express the voltage dependent of α_c.

If we assume that the mass-action principle applies, the change in occupancy of each state with time is described by a family of differential equations:

$$\frac{dC_0}{dt} = k_{-1}C_1 - k_1[Ca^{2+}]_i C_o \tag{13}$$

$$\frac{dC_1}{dt} = k_1[Ca^{2+}]_i C_o + k_{-2}C_2 - (k_{-1} + k_2[Ca^{2+}]_i)C_1 \tag{14}$$

$$\frac{dC_2}{dt} = k_2[Ca^{2+}]_i C_1 + \alpha_c O_2 - (k_{-2} + \beta_c)C_2 \tag{15}$$

$$\frac{dO_2}{dt} = \beta_c C_2 + k_{-3}O_3 - (\alpha_c + k_3[Ca^{2+}]_i)O_2 \tag{16}$$

$$\frac{dO_3}{dt} = k_3[Ca^{2+}]_i O_2 - k_{-3}O_3 \tag{17}$$

in which C_i and O_i are the time-dependent probabilities that a channel with i Ca^{2+} ions bound is respectively in the closed or the open state. The summed occupancy of

the two open states at any given time the probability that the channel is open; therefore

$$g_{K(Ca)} = \bar{g}_{K(Ca)}(O_2 + O_3) \tag{18}$$

The current through open Ca^{2+}-activated K^+ channels is assumed to be ohmic, yielding

$$I_{K(Ca)} = g_{K(Ca)}(V_m - E_K) \tag{19}$$

in which E_K is the equilibrium potential for the Ca^{2+}-activated K^+ current.

Composite Model

The two sub-models are then composed together. During current-clamp experiments, the membrane current is forced to follow a command current, I_{com}. The capacitive current can be expressed as the product of the membrane capacitance, C_m, and the rate of change of the membrane potential, dV_m/dt; each ionic current is the product of conductance and driving-force terms. Substituting these expressions in equation (1) yields

$$I_{com} = C_m dV_m / dt + g_{Ca}(V_m - E_{Ca}) + g_{K(Ca)}(V_m - E_K) + g_{Leak}(V_m - E_{Leak}) \tag{20}$$

in which g_{Ca}, $g_{K(Ca)}$, and g_L are respectively the Ca^{2+} and Ca^{2+}-activated K^+, and leakage conductances, and E_{ca}, $E_{K(Ca)}$, and E_L are the corresponding equilibrium potentials. Unlike g_{Ca} and $g_{K(Ca)}$, g_L is time- and voltage-independent. Equation (20) can be rearranged to give

$$\frac{dV_m}{dt} = -[g_{Ca}(V_m - E_{Ca}) + g_{K(Ca)}(V_m - E_K) + g_{Leak}(V_m - E_{Leak}) - I_{com}]/C_m \tag{21}$$

4.2.4 Computational Simulation

After building the mathematical descriptions of the biological system, computational simulation can be created accordingly. The creation of such simulations requires specific programming skills and understanding of computational mathematical methods. In general, the means of building simulation is to translate simultaneous mathematical equations into certain implementation in advanced programming language and combine the implementation with parameter inputs and pass the

parameterized equations to analyzing methods. The analyzing methods can be pre-developed scientific application packages. An example of such package can be the ODE toolbox in MATLAB (The MathWorks, 1994-2008). The following is a simple simulation built for Lewis & Hudspeth model under current-clamping condition in MATLAB, where function 'hudspeth' contains the equation set and all parameter information. The main program evokes the function and passes it as an input to a built-in ODE solver 'ode15s' and then render the simulation result as computer images by plotting tools.

```
Function dy = hudspeth(t,y,v)

%constants
g_ca=4.14*10^-9;Eca=100*10^-3;
alpha0=22800;beta0=0.97;
V0=70*10^-3;% holding potential
VA=8.01*10^-3;VB=6.17*10^-3;
KA=510;KB=940;
U=0.02;xi=3.4*10^-5;z=2;F=96485;C_vol=1.25*10^-12;Ks=2800;R=8.3145;T=310;
% parameters of 52ntracellular calcium concentration
K10=6*10^-6;K20=45*10^-6;K30=20*10^-6;
D1=0.2;D2=0;D3=0.2;
k1r=300;k2r=5000;k3r=1500;
alpha_c0=450;beta_c=1000;Va=33*10^-3;
```

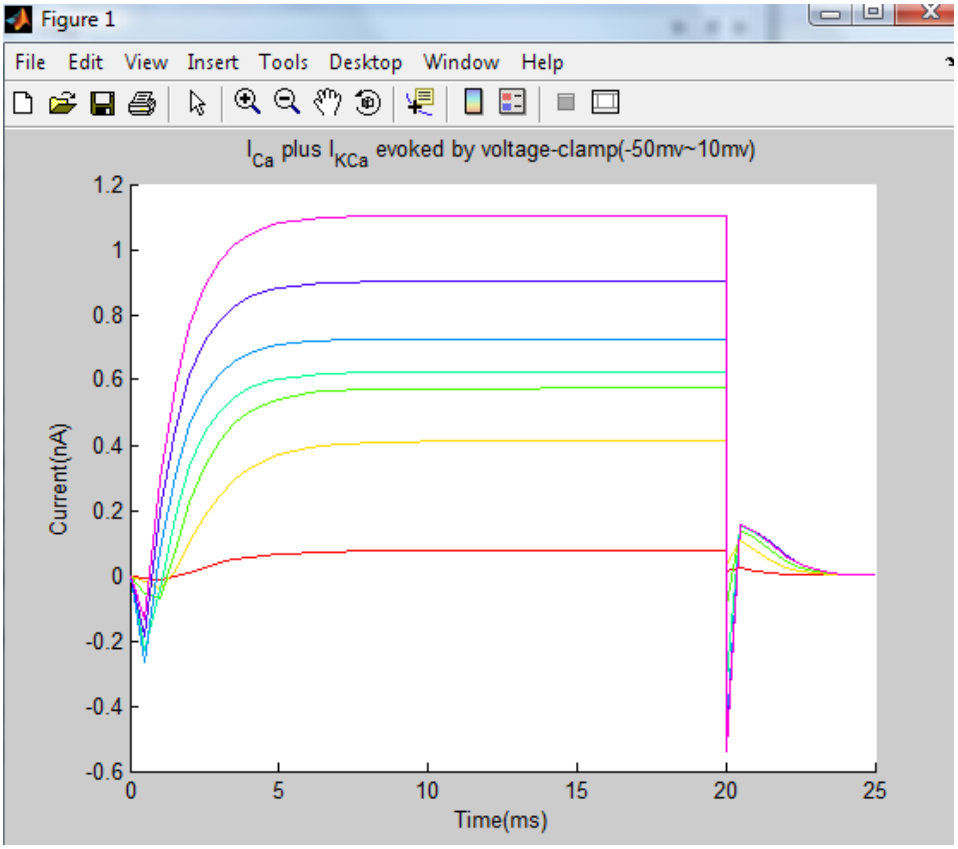


Figure 4.5 MATLAB simulation result of current-clamping in hair cell

```
% constant calculation
alpha_m=alpha0*exp(-(v+V0)/VA)+KA;
beta_m=beta0*exp((v+V0)/VB)+KB;
coef1=-1*g_ca*(v-Eca)*U/(z*F*C_vol*xi);
k1c=k1r/(K10*exp(D1*z*F*v/(R*T)));
k2c=k2r/(K20*exp(D2*z*F*v/(R*T)));
k3c=k3r/(K30*exp(D3*z*F*v/(R*T)));
alpha_c=alpha_c0*exp(-v/Va);


% ODEs
dy = zeros(7,1); % a column vector
dy(1) = beta_m*(1-y(1))-alpha_m*y(1); %m
dy(2) = coef1*y(1)^3-Ks*y(2); %[Ca++]
dy(3) = k1r*y(4)-k1c*y(2)*y(3); %C0
dy(4) = k1c*y(2)*y(3)+k2r*y(5)-(k1r+k2c*y(2))*y(4); %C1
dy(5) = k2c*y(2)*y(4)+alpha_c*y(6)-(k2r+beta_c)*y(5); %C2
dy(6) = beta_c*y(5)+k3r*y(7)-(alpha_c+k3c*y(2))*y(6); %O2
dy(7) = k3c*y(2)*y(6)-k3r*y(7); %O3

main.m  %the main program
g_ca=4.14*10^-9;Eca=100*10^-3;
g_kca=16.8*10^-9;Ekca=-80*10^-3;
V0=-70*10^-3;
v=1e-3*(-50:10:10);
tspan1 = 0:0.0005:0.02; tspan2 = 0.02:0.0005:0.025;
y0 = [0; 10^-7; 1; 0; 0; 0; 0];
result = [tspan1';tspan2'];

colorm=hsv(length(v));
figure
hold

for i=1:length(v)
[T,Y] = ode15s(@hudspeth,tspan1,[0 10^-7 1 0 0 0 0],[],v(i));
[T1,Y1]=ode15s(@hudspeth,tspan2,Y(length(Y),:),[],V0);
vm = [v(i)*ones(length(T),1); V0*ones(length(T1),1)]; T2 = [T;T1];
Y2= [g_ca*(v(i)-Eca)*Y(:,1).^3;g_ca*(V0-Eca)*Y1(:,1).^3];
Y3= [g_kca*(v(i)-Ekca)*(Y(:,6)+Y(:,7));g_kca*(V0-Ekca)*(Y1(:,6)+Y1(:,7))];
plot(1E3*T2,1E9*(Y2+Y3),'-','Color',colorm(i,:));
xlabel('Time(ms)');ylabel('Current(nA)');
title('I_{Ca} plus I_{KCa} evoked by voltage-clamp(-50mv~10mv)');
result = [result vm Y2+Y3];
end
```

The above computational simulation appears to be simple since the most computing algorithms have been hidden by the built-in ODE solver in MATLAB. If we were to build it in other advanced programming languages such as C++ or PASCAL, the implementation can sometimes be very complicated. This kind of hand-crafted computational simulation is common to researchers with quantitative background, but it will be difficult to many biologists. Even to mathematicians, the kind of simulation creation is error-prone. The simulation models created have rather low

reusability, i.e. any change of experimental data, or parameters, or details of equation, will need rework of the programming codes.

4.3 Case of Hormone-induced Calcium Oscillation Composite Model

In this section, we will introduce the model for describing calcium oscillation induced by hormones. This model is composed from several stand-alone modeling cases including hormone-induced G-protein activation (Riccobene et al., 1999), G-protein initiated $InsP_3$ release (Sneyd et al., 2004), kinetics of $InsP_3$ receptor on Endoplasmic Reticulum (Sneyd and Dufour, 2002), and a minimum model for cytosolic calcium oscillation (Hofer, 1999). This case is important to investigate the process of model integration.

4.3.1 Background

Intracellular and intercellular calcium signalling is one of the crucial methods of cellular coordination and control. Calcium ions (Ca^{2+}) are part of an information-processing system in animal and plant cells and play an essential role in regulating a variety of cellular processes such as secretion, reproduction, cell movement, cell growth and so on. The regulation mechanism is achieved by changes in the concentration of the free cytosolic calcium ions in response to external signals.

Under normal conditions, the concentration of free calcium in cells is maintained at very low levels in the cytosol (about 10^4 mMol) because of the presence of calcium pumps in the plasma membrane and the endoplasmic transports calcium out of the cell, whereas the calcium pump in the ER sequesters calcium ions in the lumen of the ER (Becker et al., 2000). In contrast, the Ca^{2+} concentration in the extracellular fluid and the blood is about 1-2 mMol, while in ER the sequestered calcium concentration is around 0.5 mMol, about 10^4 times as high as that of the cytosol. The cytosol, with its very low concentration of free Ca^{2+}, is located between two very calcium-rich environments. This results in the cytosol being a major site of transient elevation of calcium concentration which is induced by hormones and neurotransmitters and described as ‘calcium signals’(Dupont et al., 2000). Instead of switching between stationary and pulsatile regimes, very often this transient elevation exhibits a high

spatiotemporal organization characterized as an oscillatory behavior (Becker et al., 2000).

It has been shown that the calcium signals in response to agonist stimulation consist of a series of spikes in Ca^{2+} concentration with a period of a few seconds to a few minutes (Woods et al., 1987). It also appears that each spike is organized spatially. The Ca^{2+} concentration first increases locally, then the increase propagates in the whole cell as a wave, traveling at a speed of 10-20$\mu m\ s^{-1}$(Dupont et al., 2000). Moreover, it has been observed in a variety of systems that calcium signals can also propagate from one cell to another and thereby serve as a means by which a group of cells can communicate with each other, and coordinate a multicellular response to a local event (Hofer, 1999). One of such examples is glycogenolysis - the process of liver releasing glucose from glycogen. In the liver hepatocyte, agonists such as glucagon binding to their receptors will cause the increase of intracellular free calcium. The free calcium ions then serve as second messenger molecules to release glucose into the blood stream.

The phenomenon of calcium oscillation and wave propagation is observed in different kinds of cells and has been studied extensively. Calcium spikes had been known for a long time in periodically contracting muscle cells (e.g. heart cells) and neurons, before they were discovered in the mid-1980s in nonexcitable cells, notably in oocytes upon fertilization and in hepatocytes subject to hormone stimulation (Schuster et al., 2002). In excitable cells such as neurons, an increase in intracellular Ca^{2+} is brought about by the Ca^{2+} channels in the plasma membrane when these channels open during depolarization, permitting extracellular Ca^{2+} to enter the cell. By contrast, in non-excitable cells Ca^{2+} is supplied mainly by internal stores and thus involves much more complicated processes (Chay et al., 1995).

4.3.2 Understand Intracellular Calcium Oscillation by Model Integration

In this section, we give detailed description of several models that consist of the composite model of intracellular calcium oscillation. The composite model consists of the following models: Agonist-stimulated G-protein activation (Riccobene et al., 1999), G-protein initiated $InsP_3$ release (Sneyd et al., 2004), kinetics of $InsP_3$ receptor on Endoplasmic Reticulum (Sneyd and Dufour, 2002), and a minimum model for cytosolic calcium oscillation (Hofer, 1999).

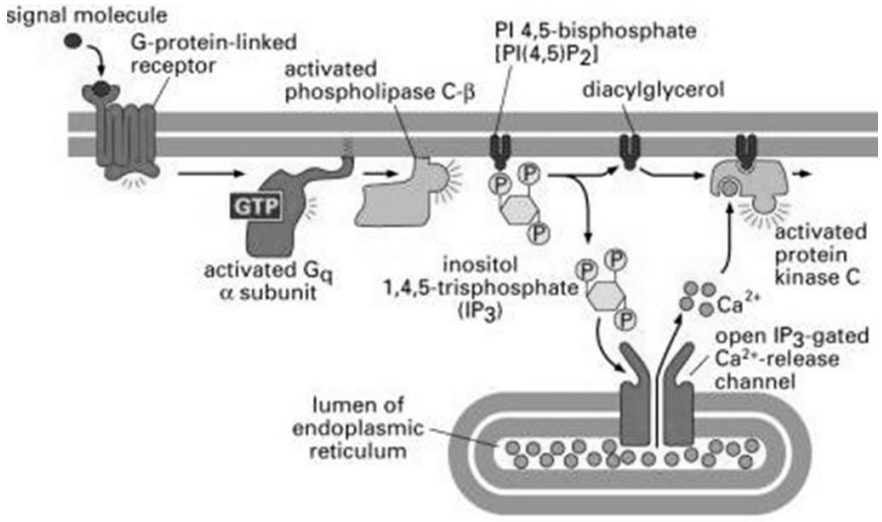


Figure 4.6Phospholipid-inositol-calcium signalling pathway (Alberts et al., 1994)

Figure 4.6 illustrates the calcium elevation in hormone-stimulated non-excitable cells mediated by the phospholipid-inositol-calcium signalling pathway. In this pathway hormonal stimuli lead to the activation of G-proteins as an effecter system. This initiates Phospholipase C (*PLC*) activation and the subsequent formation of diacylglycerol (*DAG*) and inositol-1,4,5-triphosphate ($InsP_3$) which finally leads to the release of calcium stored in the ER through the stimulation of $InsP_3$-sensitive calcium ion channels.

Agonist-stimulated G- protein Activation

Signalling through G protein-linked receptors is one of the most prevalent and important methods of transmitting information from the outside to the inside of cells (Riccobene et al., 1999). The G protein-linked receptor family is so named because ligand binding causes a change in receptor conformation that activates a particular G protein (an abbreviation for guanine-nucleotide binding protein). The G protein-linked receptors go across plasma membrane seven times with the N-terminus exposed to the extracellular environment and the C-terminus resides in the cytosol. The inactive form of the G-protein consists of α, β, and γ subunits with a molecule of GDP bound to the α-subunit. After the binding of an agonist to the extracellular side of a G protein-linked receptor, it catalyses the exchange of GDP for GTP and activates the G-protein. The G-protein is then released from the receptor and it dissociates into separate β-γ and α-GTP (active) subunits. Depending on the G protein and the cell type, either the free GTP-G_α subunit or the $G_{\beta\gamma}$ complex can then initiate signal transduction events in the cell. The activation of the G protein persists

only as long as the G_α is bound to GTP and the subunits remain separated. Active G-proteins are returned to their inactive state upon the hydrolysis of GTP to GDP by the intrinsic GTPase activity of the α-subunit and the α-GTP and β-γ subunits can recombine (Becker et al., 2000).

G-protein Initiated $InsP_3$ Release

After the activation of G-protein, different isoforms of phosphoinositide-specific phospholipase *C* (1-phosphatidylinositol-4,5-bisphosphate phosphodiesterase, *PLC*) are then activated. After that, PLC cleaves phosphatidylinositol-4,5-bisphosphate (PIP_2), a relatively uncommon membrane phospholipid, into two molecules: diacylglycerol (*DAG*) and inositol-1,4,5-triphosphate ($InsP_3$). Both $InsP_3$ and *DAG* have been shown to be second messengers in a variety of regulated cell functions. $InsP_3$ is a small water soluble molecule and can quickly diffuse through the cytosol, binding to a ligand-gated calcium channel known as the $InsP_3$ receptor channel in the ER membrane. On the other hand, the *DAG* generated by *PLC* activity remains in the membrane, where it activates the enzyme protein kinase *C* (*PKC*) (Becker et al., 2000).

Calcium-induced Calcium Release Mediated by the Kinetics of $InsP_3$ Receptor on Endoplasmic Reticulum

One of the most important mechanisms underlying the complex dynamic behaviour of calcium oscillations and waves is the dynamics of inositol-triphosphate receptor (*IPR*), which also functions as a Ca^{2+} channel. The binding of *InsP3* to *InsP3*-sensitive receptors in the ER membrane leads to the opening of calcium channels, which results in a massive flux of calcium ions from intracellular stores into the cytoplasm. The concentration of free cytosolic calcium then increase tenfold (around 10^{-3} mMol) from its resting level (10^{-4} mMol). After the initial rise in concentration of calcium ions in the cytoplasm, calcium itself also stimulates the release of additional calcium ions. This latter mechanism is called calcium-induced calcium release (*CICR*) (Berridge, 1993).

The equilibrium open probability of the *IPR* presents a bell-shaped dependence on cytosolic calcium concentration. It was also shown that at high calcium concentrations the IP_3 receptors can be inhibited (Kummer et al., 2000). The

reversible calcium-induced inhibition of calcium release observed at high calcium levels develops more slowly than the activation. The decrease of free cytosolic calcium concentration is due to the activity of the ATPases (SERCA pumps), which actively transport calcium back to the ER lumen (Dupont et al., 2000). The cooperation of positive and negative feedback of calcium release to *IPR* dynamics is believed to be essential for the occurrence of intracellular calcium oscillations. Many mathematical models have been proposed to argue that Ca^{2+}–regulated IPRs and Ca^{2+} ATPases together are sufficient to generate intracellular calcium oscillatory behaviors.

Model Integration for Intracellular Calcium Signalling

Finally, with the above detailed descriptions of the specific pathways involved in the calcium signalling system, we can summarize the whole process in the following general scheme. A number of theoretical models have been proposed to explain the signalling cascade and the scheme is rather well established now: After the binding of ligands to the extracellular side of G protein-linked receptors, the α-subunit of receptor-coupled G protein is activated. The subunit in turn stimulates a phospholipase C, which catalyzes the hydrolysis of the membrane phospholipid PIP_2 to form $InsP_3$ and *DAG* as second messengers. With $InsP_3$ binding to the $InsP_3$-sensitive receptor in the ER membrane, calcium release from the ER lumen store is ensured by the *IPR* activation. By the coordination of calcium-induced calcium release and the further inhibition of *IPR*, the calcium concentration in the cytosol then displays oscillatory behavior with the form of repetitive, sharp spikes, pacemaker-like elevation.

4.3.3. Mathematical Modelling

In this section, we introduce the mathematical models of hormone-induced calcium oscillation.

Fundamental Scheme of calcium oscillation

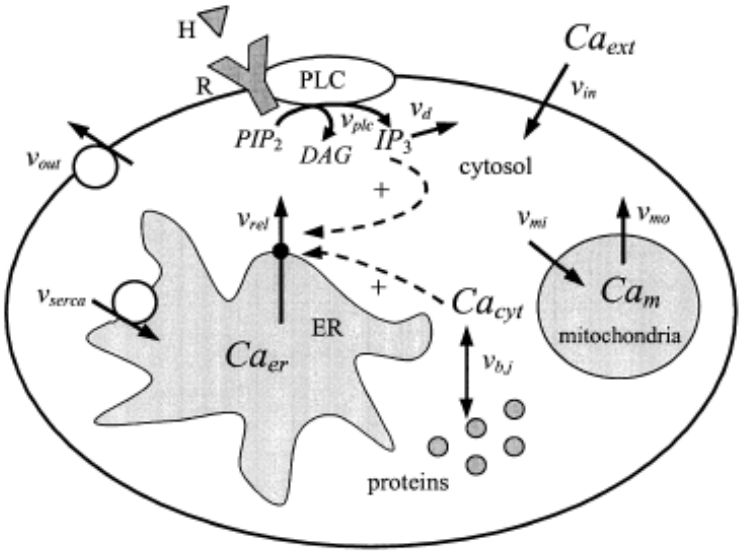


Figure 4.7 The fluxes involved in intracellular calcium oscillation

Fig 4.7 Illustrates the fluxes involved in intracellular calcium oscillation. The meaning of the symbols for reaction rates are as follows:

v_{in}, influx of Ca^{2+} across plasma membrane channels;

v_{out}, transport of Ca^{2+} out of the cell by plasma membrane Ca^{2+} ATPase;

v_{mi}, Ca^{2+} uptake into mitochondria; v_{mo}, release of $Ca2+$ from mitochondria;

v_{rel}, Ca^{2+} release from the ER through $InsP_3$-sensitive channels;

v_{serca}, transport of Ca^{2+} into the ER by sarco-/endoplasmic reticulum Ca^{2+} ATPase (SERCA);

v_{plc}, formation of $InsP_3$ and DAG catalyzed by phospholipase C (PLC);

v_d, degradation of $InsP_3$ (performed mainly by hydrolysis to inositol-1,4-bisphosphate or phosphorylation to inositol-1,3,4,5-tetrakisphosphate)

$v_{b,j}$, net rate of binding of Ca^{2+} to the j-th class of Ca^{2+} buffer (e.g. protein);

The concentrations of inositol-1,4,5-triphosphate ($InsP_3$), cytosolic calcium (Ca_{cyt}), ER calcium (Ca_{er}), mitochondrial calcium (Ca_m), and occupied calcium binding sites of the buffer species j in the cytosol (B_j) are:

$$\frac{d}{dt}IP_3 = v_{plc} - v_d \tag{1}$$

$$\frac{d}{dt}Ca_{cyt} = v_{in} - v_{out} + v_{rel} - v_{serca} + v_{mo} - v_{mi} - \sum_{j=1}^{n} v_{b,j} \tag{2}$$

$$\frac{d}{dt}Ca_{er} = \rho_{er}(v_{serca} - v_{rel}) \quad (3)$$

$$\frac{d}{dt}Ca_{m} = \rho_{mit}(v_{mi} - v_{mo}) \quad (4)$$

$$\frac{d}{dt}B_{j} = v_{b,j} \text{ (For each } j\text{=1, 2... } n\text{)} \quad (5)$$

Where ρ_{er} and ρ_{mit} are the cytosol/ER and cytosol/mitochondria volume ratios (Schuster et al., 2002).

Minimal model

A mathematical model of calcium dynamics by using minimal set of variables is developed that satisfactorily accounts for the properties of agonist-evoked calcium oscillations in an "average" hepatocyte in the work of Hofer, T. (Hofer, 1999). The G-protein coupled receptor activation, the phospholipase pathway and the $InsP_3$-sensitive channel dynamics are simplified into an algebraic chain relation with which the concentration of $InsP_3$ is taken as proportional to the dose of agonist. In the minimal mathematical model, spatial homogeneity is assumed. The model computes the change in free cytosolic calcium concentration by summing the relative flux of Ca^{2+} from the ER, plasma membrane and through Ca^{2+} pumps. A constant background Ca^{2+} influx across plasma membrane is assumed to be controlled by a first order Michaelis-Menten equation and the efflux by a second order Michaelis-Menten equation dependent on the Ca^{2+} concentration. The model is based on the scheme described in section 4.3.1 but an alternative variable is utilized to measure the total free calcium content of the whole cell:

$$Ca_{w} = Ca_{cyt} + \frac{C_{er}}{C_{c}}Ca_{er} \quad (6)$$

where C_{er} and C_c stand for the effective volume of ER and cytosol.

The balance equations read:

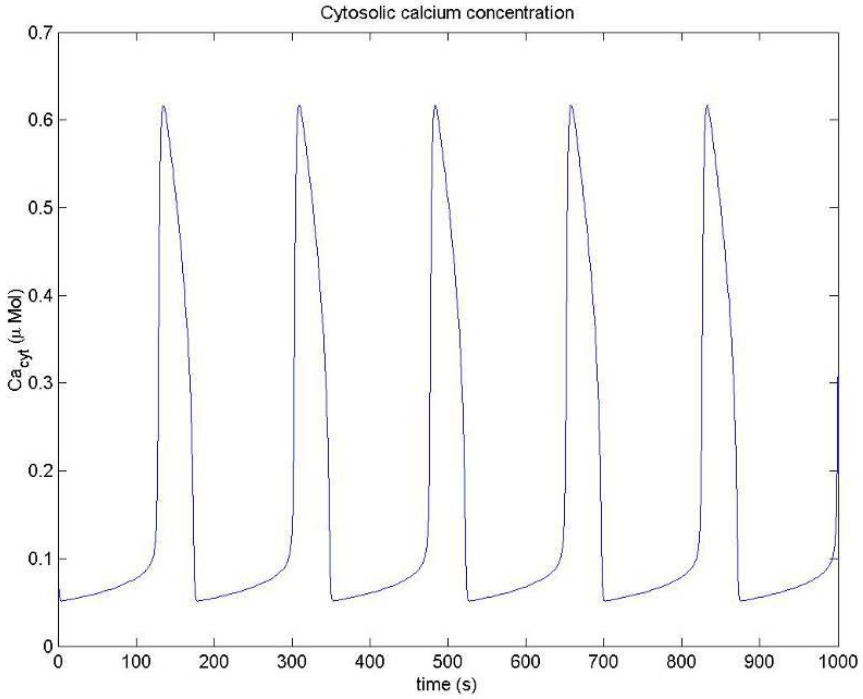
$$\frac{d}{dt}Ca_{cyt} = \rho[v_0 + v_c\frac{IP_3}{K_0 + IP_3} - v_4\frac{Ca_{cyt}^{\ 2}}{K_4^{\ 2} + Ca_{cyt}^{\ 2}} + \alpha(k_r(Ca_{cyt}, IP_3)\frac{Ca_w - (1+\beta)Ca_w}{\beta} - v_3\frac{Ca_{cyt}^{\ 2}}{K_3^{\ 2} + Ca_{cyt}^{\ 2}})] \quad (7)$$

$$\frac{d}{dt}Ca_{w} = \rho(v_0 + v_c\frac{IP_3}{K_0 + IP_3} - v_4\frac{Ca_{cyt}^{\ 2}}{K_4^{\ 2} + Ca_{cyt}^{\ 2}}) \quad (8)$$

With the IPR release sub-function

$$k_r(Ca_{cyt}, IP_3) = k_1 \frac{d_2 \frac{d_1 + IP_3}{d_3 + IP_3} IP_3 \cdot Ca_{cyt}}{(d_p + IP_3)^3 (d_a + Ca_{cyt})^3 (d_2 \frac{d_1 + IP_3}{d_3 + IP_3} IP_3 \cdot Ca_{cyt})^3} + k_2 \quad (9)$$

Based on the above mathematical model, a simulation is implemented in MATLAB. The initial values of the concentrations and the parameters are chosen as in the Appendixes Table 1. The estimation of the structural parameters is based on calcium diffusion in *Xenopus* oocytes, which is also discussed in Hofer's work. The computation result shows in Figure 4.8. We can see that the characteristics of intracellular free calcium are successfully reflected by our model and simulation.



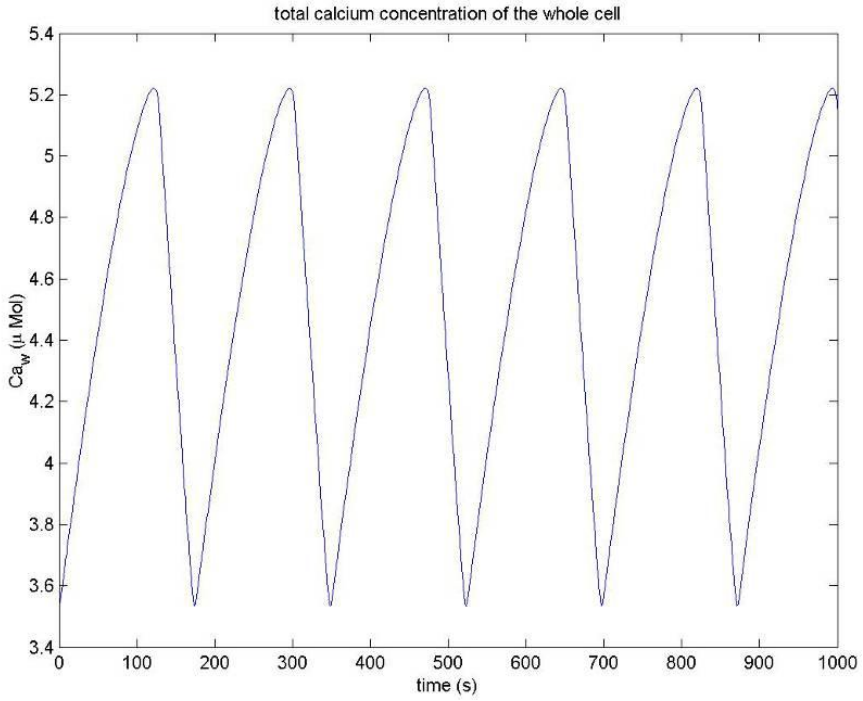


Figure 4.8 Intracellular calcium oscillations in the Hofer's model

G-protein coupled receptor activity

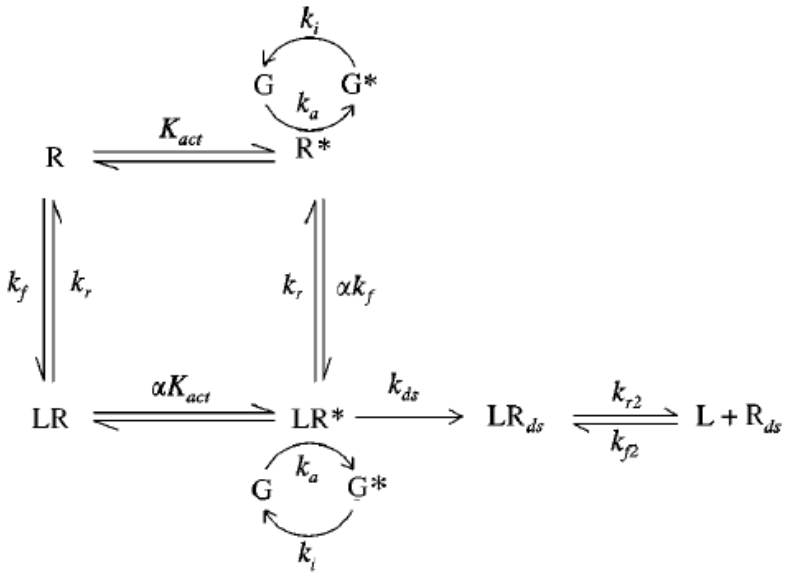


Figure 4.9 Scheme of G protein coupled receptor signaling

Fig 4.9 illustrates G protein coupled receptor signalling including G protein activation and receptor desensitization. Reproduced from (Riccobene et al., 1999). R is the inactive form of the receptor, *R** is the active form of the receptor, *LR* is the inactive ligand/receptor complex, *LR** is the active ligand/receptor complex, LR_{ds} is the desensitized ligand/receptor complex,

G is inactive G-protein, G^* is activated G-protein, L is free ligand, and R_{ds} is the desensitized receptor. Parameter values and definitions are given in Appendixes Table 2.

A mathematical model for G protein coupled receptor signaling is introduced based on the work of Riccobene, et al. in 1999. In their model, G protein activation and receptor desensitization are included and prediction can be made on how activation and desensitization would change if either the conformational selectivity (the effect of ligand binding on the distribution of active and inactive receptor states) or the desensitization rate constant was ligand-specific (Riccobene et al., 1999). The balance equations are the following:

$$\frac{d}{dt}R = (k_r LR + \frac{k_{fR}}{K_{act}}R^*) - (k_f L \cdot R + k_{fR}R) \tag{10}$$

$$\frac{d}{dt}R^* = (k_r LR^* + k_{fR}R) - (\alpha k_f L \cdot R^* + \frac{k_{fR}}{K_{act}}R^*) \tag{11}$$

$$\frac{d}{dt}LR = (k_f L \cdot R + \frac{k_{fR}}{\alpha K_{act}}LR^*) - (k_r LR + k_{fR}LR) \tag{12}$$

$$\frac{d}{dt}LR^* = (\alpha k_f L \cdot R^* + k_{fR}LR) - (k_{ds}LR^* + \frac{k_{fR}}{K_{act}}LR^* + k_r LR^*) \tag{13}$$

$$\frac{d}{dt}LR_{ds} = (k_{ds}LR^* + k_{f2}L \cdot R_{ds}) - k_{r2}LR_{ds} \tag{14}$$

$$\frac{d}{dt}R_{ds} = k_{r2}LR_{ds} - k_{f2}L \cdot R_{ds} \tag{15}$$

$$\frac{d}{dt}G = k_i G^* - k_a G \cdot (LR^* + R^*) \tag{16}$$

$$\frac{d}{dt}G^* = k_a G \cdot (LR^* + R^*) - k_i G^* \tag{17}$$

The activation and desensitization parameters, ligand binding rate constants and total species concentrations used in Riccobene's model are the following:

Kinetics of $InsP_3$ receptor on Endoplasmic Reticulum

A model that describes the Ca^{2+} binding to the receptor using saturating, not mass-action kinetics has been developed (Sneyd and Dufour, 2002). The model assumes that the binding of IP_3 and Ca^{2+} is sequential, not independent, so Ca^{2+} can bind to the activating site only after IP_3 has bound.

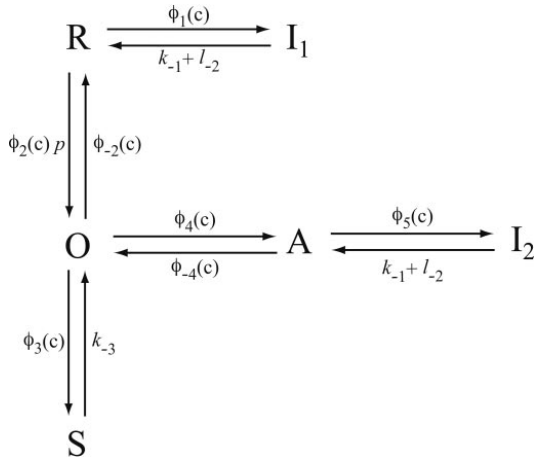


Figure 4.10 Scheme of IP3 Kinetics on Endoplasmic Reticulum

The basic scheme is the following: an $InsP_3$ receptor, R, can bind Ca^{2+} and inactivate to state I_1, or it can bind IP_3 and open to state O. State O can then shut (state S) or bind Ca^{2+} and activate to state A. State A can then bind Ca^{2+} and inactivate to state I_2. The balance equations are the following:

$$\frac{dR}{dt} = \Phi_{-2}O - \Phi_2 pR + (k_{-1} + l_{-2})I_1 - \Phi_1 R \tag{18}$$

$$\frac{dO}{dt} = \Phi_2 pR - (\Phi_{-2} + \Phi_4 + \Phi_3)O + \Phi_{-4}A + k_{-3}S \tag{19}$$

$$\frac{dA}{dt} = \Phi_4 O - \Phi_{-4}A - \Phi_5 A + (k_{-1} + l_{-2})I_2 \tag{20}$$

$$\frac{dI_1}{dt} = \Phi_1 R - (k_{-1} + l_{-2})I_1 \tag{21}$$

$$\frac{dI_2}{dt} = \Phi_5 A - (k_{-1} + l_{-2})I_2 \tag{22}$$

Where

$$\Phi_1(c) = \frac{(k_1 L_1 + l_2)c}{L_1 + c(1 + L_1 / L_3)} \tag{23}$$

$$\Phi_2(c) = \frac{k_2 L_3 + l_4 c}{L_3 + c(1 + L_3 / L_1)} \tag{24}$$

$$\Phi_{-2}(c) = \frac{k_{-2} + l_{-4}c}{1 + c / L_5}$$

(25)

$$\Phi_3(c) = \frac{k_3 l_5}{L_5 + c} \tag{26}$$

$$\Phi_4(c) = \frac{(k_4 L_5 + l_6)c}{L_5 + c} \tag{27}$$

$$\Phi_{-4}(c) = \frac{L_1(k_{-4} + l_{-6})}{L_1 + c} \tag{28}$$

$$\Phi_5(c) = \frac{(k_1 L_1 + l_2)c}{L_1 + c} \tag{29}$$

Based on the experimental data, the best-fit parameter values are:

$k_1 = 0.64 s^{-1} \cdot \mu M^{-1}$, $k_{-1} = 0.04 s^{-1}$, $k_2 = 37.4 s^{-1} \cdot \mu M^{-1}$, $k_{-2} = 1.4 s^{-1}$, $k_3 = 0.11 s^{-1} \cdot \mu M^{-1}$, $k_{-3} = 29.8 s^{-1}$, $k_4 = 4 s^{-1} \cdot \mu M^{-1}$, $k_{-4} = 0.54 s^{-1}$, $L_1 = 0.12 \mu M$, $L_3 = 0.025 \mu M$, $L_5 = 54.7 \mu M$, $l_2 = 1.7 s^{-1}$, $l_4 = 1.7 s^{-1} \cdot \mu M^{-1}$, $l_6 = 4707 s^{-1}$, $l_{-2} = 0.8 s^{-1}$, $l_{-4} = 2.5 s^{-1} \cdot \mu M^{-1}$, $l_{-6} = 11.7 s^{-1}$.

It is assumed that the *InsP*$_3$ receptor consists of four independent and identical subunits and allows Ca^{2+} current when all four subunits are in state *O*, or all four are in state *A*, or some intermediate combination (for instance, when three are in state *O*, and one is in state *A*). Further assumption is also made that the more subunits there are in state *A*, the greater the open probability of the receptor. With these assumptions, the open probability of the receptor is most conveniently written as $(0.1O + 0.9A)^4$ (The numbers 0.1 and 0.9 are not crucial) (Sneyd et al., 2004) and can then be applied to specify the flux term v_{rel} in fundamental balance equation (2):

$$v_{rel} = [k_f (0.1O + 0.9A)^4 + g_1](Ca_{er} - Ca_{cyt}) \tag{30}$$

Where g_1 models a constant background leak from the ER at zero *IP*$_3$ concentration (Sneyd and Dufour, 2002).

4.4 Chapter Summary

In this chapter we have introduced three case studies of biological modelling. These three cases are of biological significance and exemplary in the problem domain of biological modelling. They cover a reasonable range of models and are comprehensive enough to represent the major challenges faced in the day-to-day modelling effort. In the following chapter, we will examine the main knowledge

components involved in the modelling processes. Then, we will propose our solution within Semantic Web infrastructure to help biological modelling by using ontology and Semantic Web Services.

Chapter 5 : Using Semantic Web Technologies to Support Biological Modeling

After the introduction of biological and mathematical modeling background for these case studies, we will then identify different types of knowledge and information flow among participants during the modeling processes. In order to do so, we will first summarize all the modeling tasks of Lewis & Hudspeth case into an agent dialogue, which describe a sequence of events by defining actors, information flows, and actions. We will then give the typology of knowledge in biological modeling based on this summary. After that by examining the knowledge typology of biological modeling we will discuss how semantic web technologies are able to help in these modeling processes. We will propose a systematic solution of using ontology and Semantic Web Service to support modeling in systems biology.

5.1 Workflow of Modeling Processes

Biological modeling are usually motivated by new discoveries of biological phenomena or biological entities and then focused on gaining further understanding on their underlying mechanisms. The efforts generally require defining the scope of the problem domain, identifying all the significant factors which account for the phenomenon, understanding the relationships between these factors, and then quantitatively describing the factors and the relationships between them.

In this context, the initial actions are usually taken by experimenters who make discoveries by conducting experiments and defining the scope of interests. The rest of the efforts rely on the collaboration among experiment specialists, modeling specialists, and computing specialists. In practice, modelers and computing specialists are the same actors since nowadays mathematicians who do quantitative analysis for modeling systems very often have certain amount of computational skills and perform these computing tasks on their own. We distinguish these two actors for the purpose of highlighting the different knowledge sets required for these two different roles to conduct actions.

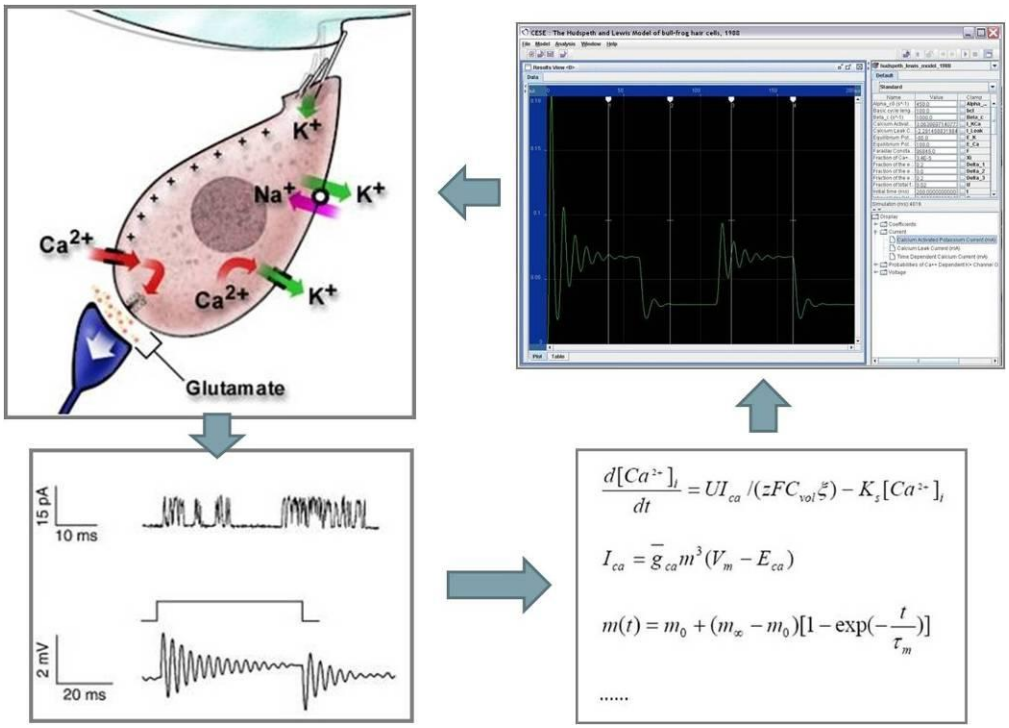


Figure 5.1 Processes involved in the study of systems biology

I have observed Experimentalist, Modeler and Computer Specialists in general by closely observing a few biological modeling projects, such as the team of Beacon project in UCL CoMPLEX working on modeling for liver tissue, and the postgraduate students in UCL Ear Institute working on electrophysiology of cochlear hair cells. I have then carried out a modeling exercise on the Lewis and Hudspeth case of electrical resonance in hair cells, and derived the workflow by considering the distinctive skill set required to perform various experimental and modeling tasks.

In order to demonstrate this collaboration more conveniently, we describe the roles of these specialists and the communication activities between them by an agent dialogue scheme. In the description, we will use E to stand for experimenter, M for modeler, and C for computing specialist. We also define the direction of information flow as follows:

- Experimenter→Modeler: Experimenter passes knowledge to Modeler
- Modeler→Experimenter: Modeler passes knowledge to Experimenter
- Modeler→Computer Specialist: Modeler passes knowledge to Computing Specialist
- Computer Specialist→Modeler: Computing Specialist passes knowledge to

Modeler

- Computer Specialist→Experimenter: Computing Specialist passes knowledge to Experimenter
- Experimenter → Computer Specialist: Experimenter passes knowledge to Computing Specialist

The actual activities are explained by a sequence of events as the following:

Stage I: Identify the factors of the phenomenon and relationships among them

Experimenter: Conducts current-clamp experiment on single saccular hair cell of the bull-frog;

Experimenter: Records correlated data of membrane potential against total membrane current;

Experimenter→Computer Specialist: E tells the data to C;

Computer Specialist: Interprets the data into graph with current as X-axis and voltage as Y-axis which shows that the voltage oscillates against current pulse;

Computer Specialist→Experimenter: C tells the graph to E;

Experimenter: Uses the graph as evidence and asserts the existence of electrical resonance phenomenon in saccular hair cells of bull-frog;

Experimenter→Modeler: E tells the assertion and its associated annotation to M;

Modeler: Searches for all the existing models that describe intracellular oscillatory behaviors;

Modeler: Recalls an electrical analog circuit model (LRC model) describing the subthreshold oscillatory behavior of the squid's giant axon;

Modeler→Experimenter: Tells the model description to E;

Experimenter: Asserts the LRC model is related to current case;

Experimenter→Modeler: E asks qualitative information on the LRC model;

Modeler→Experimenter: M tells E the answer;

Experimenter: By looking at the answer, E asserts that the membrane's capacitance and ionic conductances are the factors of the model;

Experimenter: Searches for the existing knowledge on cellular ionic conductances;

Experimenter: Recalls the presence of voltage- and ion-dependent channels responsible for membrane conductances including voltage-dependent Ca^{2+} conductance, transient voltage-dependent A-type K^+ conductance, and Ca^{2+}-sensitive K^+ conductance;

Experimenter→Modeler: E asks the relationship between time, membrane potential, currents, capacitance, and conductances;

Modeler: Recalls the following mathematical descriptions and the annotations for each term:

$$I_m = I_{cap} + \sum I_{ion}$$

$$I_{Cap} = C_m \cdot \frac{dV_m}{dt}$$

$$\text{any } I_{ion} = g \cdot (V_m - E)$$

I_m: Total membrane current; I_{cap}: Capacitive current; I_{ion}: Ionic current; C_m: Membrane capacitance; g: Conductance; V_m: Membrane potential; E: Equilibrium potential

Modeler: Translates the mathematical descriptions into assertion set A1:

- The total membrane current is at all times equal to the sum of capacitive and ionic currents;
- The capacitive current is the product of the membrane capacitance and the rate of change of membrane potential;
- The ionic current is at all times the product of conductance and the value of membrane potential subtracting equilibrium potential.

Modeler→Experimenter: M tells the assertion set A1 to E;

Experimenter: Asserts that during current-clamp experiments the total membrane current is at all times equal to the command current; and link it to assertion set A1;

Experimenter: Asserts that the ionic currents consist of Ca^{2+}-current and Ca^{2+}-sensitive K^{+}-current; and link it to assertion set A1;

Experimenter→Modeler: E tells updated assertion set A1 to M;

Modeler: Translates assertion set A1 to mathematical description:

$$I_m = I_{cap} + \sum I_{ion}$$

$$I_{Cap} = C_m \cdot \frac{dV_m}{dt}$$

$$I_m = I_{com}$$

$$\sum I_{ion} = I_{Ca} + I_{K(Ca)}$$

$$I_{Ca} = g_{Ca} \cdot (V_m - E_{Ca})$$

$$I_{K(Ca)} = g_{K(Ca)} \cdot (V_m - E_{K(Ca)})$$

Modeler: Manipulate the mathematical descriptions to:

$$\frac{dV_m}{dt} = -[g_{Ca}(V_m - E_{Ca}) + g_{K(Ca)}(V_m - E_{K(Ca)}) - I_{com}] / C_m$$

Modeler: Translates the mathematical descriptions into assertion a1;

Modeler→Experimenter: M tells assertion a1 to E;

Experimenter: Asserts that the factors required for quantitatively describing the electrical resonance mechanism are the membrane capacitance, the Ca^{2+} conductance, the Ca^{2+}-sensitive K^{+} conductance; the equilibrium potential of Ca^{2+}-current, and the equilibrium potential of Ca^{2+}-sensitive K^{+}-current.

Experimenter: Asserts that by applying pharmaceutical agent tetraethylammonium (TEA) the Ca^{2+}-sensitive K^{+}-current can be inactivated;

Experimenter: Infers that the Ca^{2+} conductance and the equilibrium potential of Ca^{2+}-current can be investigated separately from the Ca^{2+}-sensitive K^{+} conductance and Ca^{2+}-sensitive K^{+}-current.

Experimenter→Modeler: E tells M the inference;

Modeler: Manipulate the mathematical descriptions into sub-components:

$$I_{Ca} = g_{Ca} \cdot (V_m - E_{Ca})$$

$$I_{K(Ca)} = g_{K(Ca)} \cdot (V_m - E_{K(Ca)})$$

$$\frac{dV_m}{dt} = -[I_{Ca} + I_{K(Ca)} - I_{com}] / C_m$$

Stage II: produce valid mathematical descriptions for individual sub-component

Modeler: Recalls existing models for Ca^{2+}-current;

Modeler: Asserts that the activation time course of Ca^{2+}-current is adequately described by a third-order kinetic scheme without inactivation;

Modeler: Recalls the scheme which can be described as:

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

$$m(t) = m_0 + (m_\infty - m_0)[1 - \exp(-\frac{t}{\tau_m})] \qquad (*)$$

$\bar{g}_{ca}$: the limiting value of Ca^{2+} conductance when all Ca^{2+} channels are open;

m : the time-dependent value of the activation parameter, varies between zero and unity with time;

m_0: the initial value of *m* before a potential change;

m_∞, the equilibrium value of m_0 at the new potential

τ_m: the time constant of m's exponential approach to

Modeler→Experimenter: To parameterize the scheme, M asks E for measurements;

Experimenter: Conducts a series of voltage-clamp experiments on single saccular hair cell of the bull-frog under the condition of TEA;

Experimenter: For each voltage-clamping, records Ca^{2+} current against time;

Experimenter→Modeler: E tells M all the data on the series of voltage-clamp experiments;

Modeler: Estimates tail-current amplitudes from single-exponential extrapolations back to the end of the pulse;

Modeler: Fits a Boltzmann relation $I = I_{\max} /[1 + e^{-(V_m - V_{1/2})/\kappa}]^3$ to the estimated tail-current data by a least-square-error criterion (in which $I_{\max}$ is the peak tail current with all Ca^{2+} channels open, $V_{1/2}$ is the potential at which m_∞ =0.5, and κ is a slope factor describing the voltage dependence of activation); gains the following parameter values: $I_{\max}$ =-0.5nA, $V_{1/2}$ =-36.5mV; and κ =9.4mV, $\bar{g}_{ca}$ =4.14nS, and generates the data sets on m_∞ against V_m and τ_m against V_m;

Modeler: Manipulates equation (*):

Since m varies between zero and unity with time, M assumes $\frac{dm}{dt} = \beta_m(1-m) - \alpha_m m$

α_m : the *m* gate's closing rate constant;

β_m : the *m* gate's opening rate constants.

Also assuming $m_0 = 0$ at holding potentials below -60 mV, M express τ_m and m_∞ as:

$$\tau_m = 1/(\alpha_m + \beta_m);$$

$$m_\infty = \beta_m /(\alpha_m + \beta_m).$$

Modeler: Calculate the data sets of α_m against V_m and β_m against V_m from the data set on m_∞ against V_m and τ_m against V_m;

Modeler: Recalls empirical functions of α_m and β_m against V_m:

$$\alpha_m = \alpha_0 \exp[-(V_m + V_0 / V_A)] + K_A \qquad \text{(a)}$$

$$\beta_m = \beta_0 \exp[(V_m + V_0)/V_B] + K_B \quad \text{(b)}$$

$\alpha_0, \beta_0, K_A, K_B$ are rate constants to be determined and V_0, V_A and V_B are potentials to be determined;

Modeler: Fits function (a) and (b) to the data sets of α_m against V_m and β_m against V_m, gains parameter $\alpha_0, \beta_0, K_A, K_B, V_0, V_A$ and V_B;

Modeler: Asserts that sub-component 1 is fully parameterized.

Modeler→Computer Specialist: M tells all the mathematical descriptions and parameter values to C;

Computer Specialist: Translates the mathematical descriptions and parameter values into computational programs;

Computer Specialist: Generates the computational results;

Computer Specialist: Interprets data to graphs;

Computer Specialist→Experimenter: C tells the graph to E;

Experimenter: By comparing to experimental data, asserts the mathematical description is valid.

Stage III: Integrate mathematical description of individual sub-components

Modeler: Links the two mathematical descriptions together;

Modeler→Computer Specialist: M tells the integrated mathematical descriptions and according parameter values to C;

Computer Specialist: Translates the mathematical descriptions and parameter values into computational programs;

Computer Specialist: Generates the computational results;

Computer Specialist: Interprets data to graphs;

Computer Specialist→Experimenter: C tells the graph to E;

Experimenter: Compare the graphs from C with the graphs from experimental results, asserts that they are inconsistent;

Experimenter→Modeler: E tells M that the mathematical descriptions for the electrical resonance are invalid;

Modeler: Add term leak current as an ionic current and assign empirical parameters, gains updated description:

$$I_{Ca} = g_{Ca} \cdot (V_m - E_{Ca})$$

$$I_{K(Ca)} = g_{K(Ca)} \cdot (V_m - E_{K(Ca)})$$

$$I_L = g_L \cdot (V_m - E_L)$$

$$\frac{dV_m}{dt} = -[I_{Ca} + I_{K(Ca)} + I_L - I_{com}] / C_m$$

Modeler→Computer Specialist: M tells the updated mathematical descriptions and according parameter values to C.

Computer Specialist: Translates the mathematical descriptions and parameter values into computational programs;

Computer Specialist: Generates the computational results;

Computer Specialist: Interprets data to graphs;

Computer Specialist→Experimenter: C passes the graph to E;

Experimenter: By comparing to experimental data, asserts the mathematical description is valid.

End of workflow.

The above workflow mainly describes the collaboration processes among experimenters and modelers for building a quantitative model based on experimental data. It demonstrates the complexity of such effort, the spectrum of knowledge involved to establish a solid modeling scheme, and reuse part of the existing

modeling knowledge to compose with other compound models to form a systematic description of a biological system.

In the following sections, we will classify the knowledge involved in the modeling processes, identify the relationships among these processes, and discuss how the Semantic Web technology can possibly fit in to facilitate the modeling processes and enhance the reusability of models and collaboration among participants of such modeling tasks.

5.2 Typology of Modeling Knowledge

By examining the dialogue-like summary of Lewis & Hudspeth case in previous section, we can identify the main knowledge components involved in biological modeling tasks. They are the following:

Entity and relationship: Identifying the subjects of interest is always a prerequisite when doing any kind of study. During biological modeling, it is required to define the scope of the biological system to be investigated and the elements to be described in such system. The definitions of biological entities are the starting point of biological modeling. The relationships that connect these entities are also crucial knowledge. In fact, the task of modeling is to understand the intrinsic relationships among biological entities in a qualitative or quantitative way. The definitions of biological entities and the relationships among them are the foundation of higher levels of knowledge to be discussed.

The biological entities and relationships are the building blocks of conceptual modeling for biological systems. They consist of the basic semantics in formal knowledge representation of the biological models. They are also used to define the elements in the schema of biological relational databases. Moreover, biological entities are used to annotate the terms of mathematical equations while these equations describe the relationships among these entities.

The definitions of biological entities and relationships are normally given in literature separately from the models. The correspondence between these definitions and the model components are loose. In recent modeling efforts such as CellML and SBML, biological entities and relationships are defined. However, CellML and

SBML do not allow models to be shared across authors by using consensus ontologies as their building blocks. Although CellML and SBML -enabled software, such as COPASI and the Systems Biology Workbench, allow model sharing by file exchange, both CellML and SBML do not have the element that can include ontological information for checking terminology consistencies among models.

Experimental data: Data are acquired from experiments by using various biological technologies. These data are obtained for many purposes. First, data are used by mathematicians to formulate quantitative relationships among biological entities. Second, data are used to parameterize these formulated equations. Third, data are used to initiate computational simulations. Fourth, data are used to compare with simulation outputs to verify the validity of quantitative models.

The data acquired from experiments however are not always accessible to others. They may be saved as un-digitized figured, put in literature and unsuitable for computer-based processing. They may also exist as different types of datasheets and databases. In order to make experimental data in various formats accessible to researchers, unified and formally represented data storage is required. To adapt the Semantic web vision, this storage needs a universal interface for communication that is able to provide computer-processable description of the storage content. The storage also needs facilities for data retrieval and data updating by external users, and provides enough security to the internal data to prevent unwanted access and manipulation.

Mathematics: many kinds of mathematical knowledge are required for constructing quantitative models for biological studies. These include methods for analyzing raw data to identify quantitative relationships among biological entities, formulating mathematical equations to represent these relationships, various fitting techniques for equation parameterization, and simulation methods for scientific computing.

Mathematical knowledge is one of the most important parts of biological modeling. Only with well-formed mathematics can produce meaningful quantitative models. Building mathematical models often requires using different sets of experimental data to formulate. It is therefore also essential for model representation to have a standard communication interface, in order to connect with different data storage. This interface again will need semantic descriptions to allow computers to

understand the content of the model as well as its input and output value types and conditions. More importantly, mathematical models should be represented in a format that can be easily decomposed into granular model parts for further reuse.

Decision-making and workflow: Knowledge of decision-making in biological modeling includes the rationale of parameterization, the selection of data sources, the selection of computational methods and their precision, and so on. Workflow among different modeling components defines the communication activities and interfacing protocols. These two types of knowledge are crucial for model reuse and model composition. They are however almost always lacking in current knowledge representation of biological models.

In the above summary, we classified the main knowledge components of biological models and identified the difficulties of formally representing these components in compliance with the Semantic Web vision. In the following section, we will revisit the Semantic Web technologies and establish connections between these technologies and the problem domain of biological modeling.

5.3 From Modeling Knowledge to Semantic Web Components

Having examined the various knowledge involved in the real cases of biological modeling above, now we recall that our goal is to meet the challenges of formally representing all these kinds of knowledge in computer-based exchange formats, so that researchers are able to share and reuse them with the full power of the toolset provided by Semantic Web Infrastructure as discussed in Chapter 2. Having also investigated the reach and scope of the Semantic Web in its current state with respect to life sciences in Chapter 3, we are now ready to formulate our approach to utilize ontology and Semantic Web Services into our scenarios.

Currently, only a few ontological languages are developed for representing knowledge of biological modeling, namely CellML (Cuellar et al., 2003) and SBML (Finney and Hucka, 2003). Both SBML and CellML are able to provide representation of knowledge components including biological entities, parameter specifications and the mathematical equations of the underlying biological. These formats, however, don’t possess formal mechanism of using shared terminology

across model instances, nor are they able to embed representation of model parameterization or associated rationale in the model to be built.

As discussed previously, in biological modeling, one of the major difficulties in knowledge reuse is to allow the discovery of distributed knowledge. This requires knowledge to be represented in formal exchange format and share the same terminology. This shared terminology has to be general enough to be used across all knowledge representation but specific enough to make meaningful definitions possible.

Moreover, it is also crucial to represent biological models with formal logics so that semantic queries can be made across the model repository. For example, if one researcher built a biological model which contains an entity annotated with the term "sodium channel", while another research wants to query a range of models that contains the entity "cation channel", although human researchers are able to deduce that sodium channel is a type of cation channel and therefore satisfies the queries, without a layer of formal logic that describe such relationship, computers won't be able to return the proper result.

For the above reasons, we propose an upgrade of knowledge representation format from CellML and SBML specifications to OWL ontology-based models. The proposed ontology format will not only contain all the knowledge components represented in these formats, but also provide a mechanism for using shared biological ontologies and allows embedding references of external experimental data sources and profiles of parameterization processes.

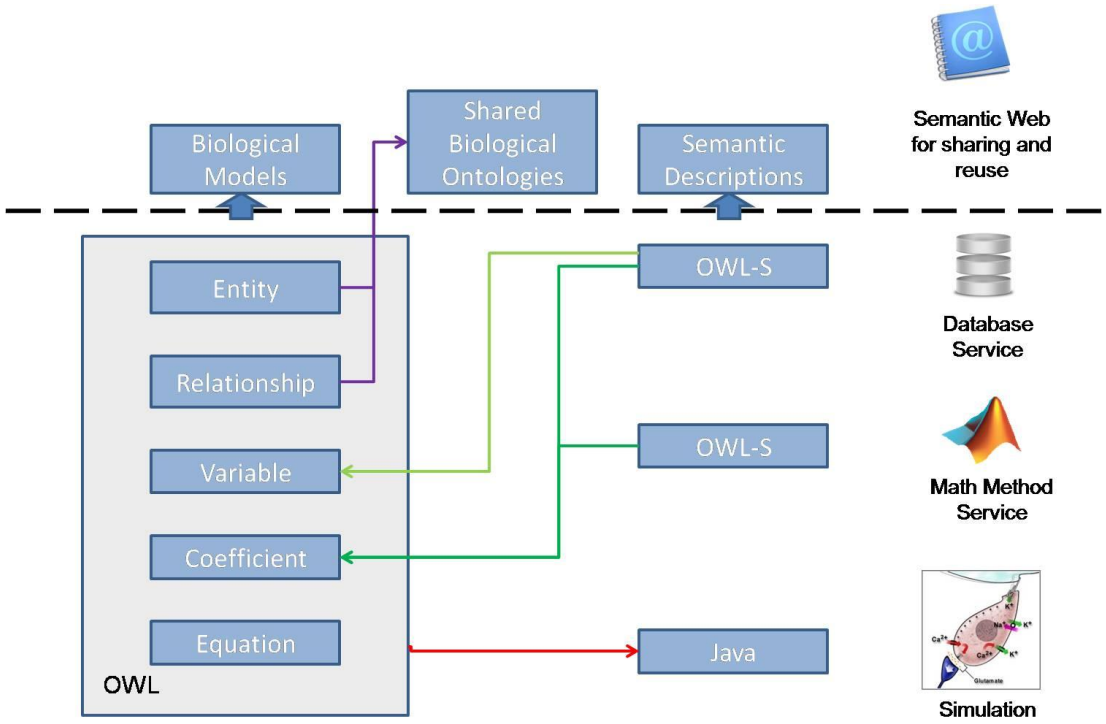


Figure 5.2 A scheme to illustrate the content of the meta-model and its relationship with external knowledge sources

Figure 5.2 demonstrates what knowledge components the proposed ontology format contains, and how it links to shared biological ontologies and allows embedding references of external experimental data sources, mathematical methods used for parameterization processes and possibility of transformation from the model to simulation.

With the upgraded knowledge representation for biological models, the remaining issues are to standardize the access to the biological databases, mathematical methods used to analyze the data and the workflow among these data and methods. The standardization of these forms of information still needs to be carried out within the Semantic Web infrastructure. And very importantly, it needs to be done in the way that the knowledge standardization can be easily reused and integrated.

The conventional means to standardize biological data is to convert different data format into relational databases. The content of these databases are then defined by the database schemas. The retrieval and update of the data are achieved by inputting database queries commands such as SQL. There are a few possibilities to transform conventional databases into units in compliance with Semantic Web. One is by using agent-based framework. Agent framework provides communicative protocols that allow distributed autonomous software entities to interoperate with other software

entities in a uniform manner. Therefore, one can develop ontologies that describe the biological elements of the databases schemas as part of the internal knowledge of these software agents. These agents can then use relational databases as their back-end agents with full control of their data content and communicate with other agents in Semantic Web infrastructure in a standardized way.

Another possibility is to use Web Services standards instead of agent framework. Web Services infrastructure is able to provide functionality to deploy fault-tolerant, distributed, multi-tier Java software, based largely on modular components running on application servers. Web Services standards in Semantic Web in fact is very similar to agent framework, only that the agent framework is more focused on providing autonomous capabilities to software units. Web Services, while also having standard communication protocols such as WSDL (W3C, 2001b), are oriented to enable orchestration and choreography. Since knowledge composition is crucial in modeling in systems biology, we believe that Web Service infrastructure is more fitted to our need.

The breakthrough of using Web Service to access relational databases is very recent. Only in May, 2006 the final version of the Java Persistence API was released as part of the Enterprise JavaBeans 3.0 (JSR 220) specification, which defines a unified Java querying interface to relational databases. In October, 2006, Sun Microsystems Inc. released the programming environment NetBeans 5.5 version that allows automated generating Java entity classes directly from database schemas by Java Persistence API and XSL stylesheets. Within Java EE platform, we then propose a way of converting Java entity classes for databases into Web Services by using JAXWS API.

Transforming mathematical methods used in analyzing biological data into Web Services is relatively easy by JAXWS API. The remaining issue is then describing both the database Web Services and Mathematical methods Web Services with formal semantics. As discussed in Chapter 3, using ontologies to annotate services has been addressed by many initiatives, including WSDL-S, SAWSDL and Semantic Markup for Web Services (OWL-S) under the W3C recommendation. Since we have proposed a knowledge representation upgrade to OWL ontology, it is natural to choose OWL-based semantic markup OWL-S as our format to describe the semantic layer of Web Services. During our investigation, we discovered a way to generate

OWL-S for Web Services from their WSDL profiles by using simple ontologies, which can be contained inside our OWL-based meta-model for biological modeling.

Finally, we have an ontology-centred framework that uses OWL models to formally represent fundamental modelling knowledge and allow embedment of references to external data sources and profiles of decision-making. The profiles of decision-making represent the knowledge of selection of experimental data and mathematical methods used to analyze these data. These profiles are stored in OWL-S ontology models and can be used to reconfigure the decision-making for further studies.

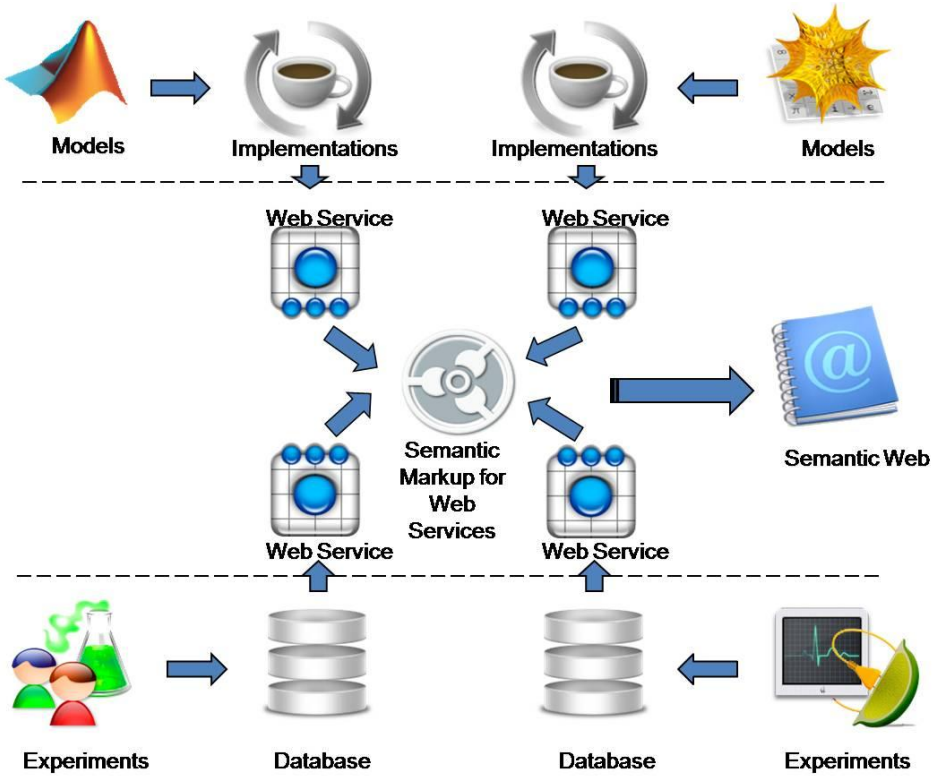


Figure 5.3 Diagram of knowledge standardization in the Semantic Web

Figure 5.3 illustrates the standardization of experimental data and mathematical methods to Web Services and the orchestration of these services by semantic markup.

5.4 Our Approach

In this section, we will go into more details of how to use ontology and Semantic Web Services to construct a knowledge standardization framework to achieve our goal.

5.4.1 Create abstract biological models by using ontology

In our approach, first we construct an abstract model of the biological systems of interest as the starting point. We use OWL DL as the format of our ontology models, since OWL DL is a species of OWL 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 using description logic.

We create an OWL abstract model as the meta-model for constructing biological models. 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.

5.4.2 From experimental Data to Database Web Services

With the help of Java EE and its supporting APIs, especially Java Persistence API, 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. 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. Both the generic schema that transforms data into relational databases, and the semantic description of Web Services can be automatically generated from the instances of the meta-model.

5.4.3 From Analysing Methods to Web Services

We use JAX-WS API in Java EE to transform mathematical methods into Web Services. 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. We then define the inputs, outputs, preconditions, and post-conditions of data analysis Web Services by using OWL-S ontology models.

5.4.4 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). It can be used to guide users in the dynamic composition of Web Services by supporting OWL-S 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. 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.

5.4.5 Outcome

By these means we constructed a fully specified ontology model for the biological system of interest. 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). A simulation is then generated automatically by transforming CellML model to JavaBeans programs.

5.5 Chapter Summary

In this chapter we have examined the main knowledge components involved in the modelling processes. Based on this we propose our solution within Semantic Web infrastructure to help biological modelling by using ontology and Semantic Web Services. Our ontology-centred framework is aimed at providing means to fulfil the requirements of biological modelling. The centrepiece is an ontology-based knowledge representation adapted for semantic web infrastructure, which not only contains modelling knowledge we identified but also allows discovering, reasoning and combining through formal semantics. Built on top of this ontology-based biological model representation, the rest of the framework is designed to link distributed data sources, mathematics, and decision-making knowledge that are involved in the modelling processes.

Chapter 6 : Description of the Framework for Biological Modeling

In this chapter, we give the step-by-step description of the proposed framework. First, we will propose an OWL-based meta-model for building biological models. This meta-model serves as a template for model creation in systems biology as well as a basis to guide the transformation from experimental data to Semantic Web Services, and it enables linking biological models with external shared ontologies for biological term annotation, references of external data sources, data analysis methods, and parameterization profiles. Then, we introduce the process of transforming biological data sources into Semantic Web Services that can be deployed in a Java-based application server-enabled programming environment. After that, we describe the transformation from data analysis methods into data analyzer Web Services. Furthermore, we introduce how to use OWL-S ontologies to defining the workflow among related semantic Web Services, in order to describe the parameterization processes for biological modeling. Finally, we introduce the automatic transformation from instances of the proposed OWL-based biological models to computational simulations.

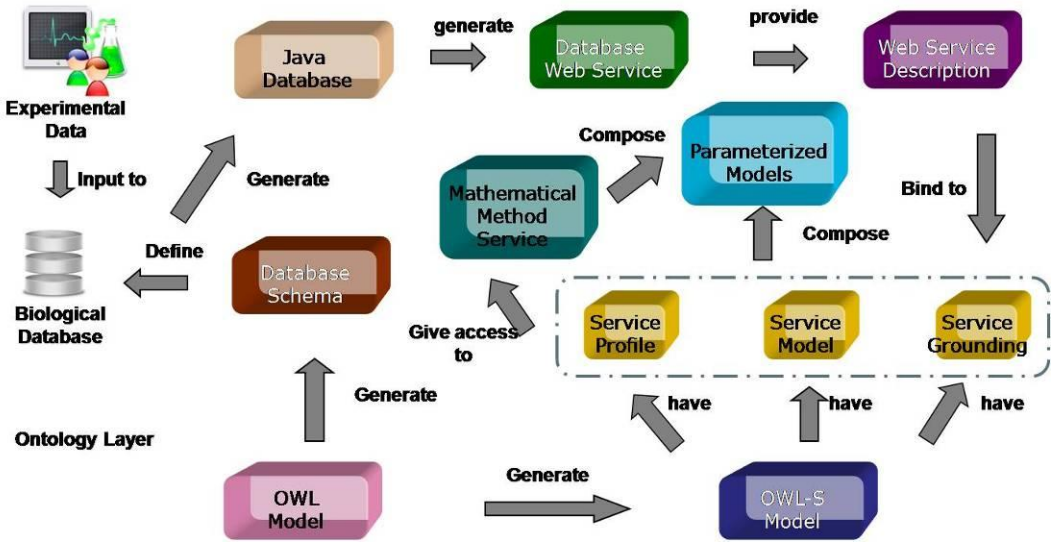


Figure 6.1 Elements and Processes of the proposed framework

6.1 Build Biological Models in OWL

In this section we describe a meta-model for building biological models. This meta-model serves as the foundation of our computational framework. It is not only used as the formal knowledge representation to store all the interested aspects of a quantitative biological model, but it also plays a role to guide the automated generation of many framework components including semantic Web Services, parameterization profiles and computational simulations from relevant modeling elements. Our proposed meta-model has the following distinctive characteristics:

- OWL-based;
- Covers all the crucial model components that are represented in both CellML and SBML specifications;
- Uses direct reference to shared biological ontologies for instantiating the property of the classes in the model;
- Enables linking to external sources including URL of experimental databases, parameterization workflow profiles, etc;
- Includes the elements for automated generation of Semantic Web Services for the proposed framework;
- Can be easily transformed to computational simulations.

The meta-model is designed to describe the elements of quantitative biological models. It is used to help users categorize, organize, and retrieve models and model parts. This meta-model is intended for both computer programs accessing the model so that they can search for models and model components based on user queries, and for human users who might want to view portions of the meta-model to evaluate whether a particular model meets their requirements.

6.1.1 Using OWL format for the Meta-model

Using formal ontology language to represent biological models is of primary importance for the life science domain and for the adoption of Semantic Web technologies by the life science community. Ontology-based knowledge representation enables the use of multiple tools without rewriting models for each tool, enables models to be shared and published in a form other researchers can use even in a different software environment, and ensures the survival beyond the lifetime of the software used to create them.

In our approach, in contrast to those existing standards such as CellML and SBML which are mainly XML specifications without formal logic, we use OWL DL as the format of our biological models. *OWL DL* is a species of OWL that provides the adequate expressiveness and has desirable computational properties for reasoning. By using OWL-based ontology as the medium, the biological models will be portable, integratable, and can be reasoned with using description logic. This OWL-based format offers several advantages over those pure XML-based standards:

Semantics. A major advantage of OWL is its well defined semantics. This prevents any ambiguity and misunderstanding in the biomedical vocabulary. The logical semantics provides the grounding of powerful reasoning services supporting ontology engineering. It also allows easy importing of existing shared ontologies, which can keep user-defined biological models relatively light-weight.

For example, in our meta-model, we import a few external shared ontologies by default, including the Geno Ontology, a formal units ontology, and an ontology that describes experimental technology in cell physiology. The ontology importing can be easily achieved by the following lines:

```
<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.geneontology.org/owl"/>
  <owl:imports rdf:resource="http://www.ucl.ac.uk/CoMPLEX/EMCollab/cell-physiology"/>
  <owl:imports rdf:resource="http://www.foodontology.nl/units"/>
</owl:Ontology>
```

By doing so, classes defined in those existing ontologies can be used directly by the current meta-model as super- and sub-classes or value of properties.

Interoperability: OWL-based ontologies explicitly define the content in information sources by formal semantics, they enable the basis of interoperability between knowledge-based sources.Interoperability of Web ontologies is crucial for shared use across different biomedical domains and fosters integration of and interoperability between Web biomedical ontologies. Once represented to OWL, biological models are easier to integrate with each other. A simply way of integration can be achieved by importing as well since all biological models are created as formal ontology. For example, one can import several OWL-based biological models sharing the same meta-model into a new ontology model. Then the definition and properties of existing entities can be reused directly.

Tools and services: OWL provides access to an expanding range of tools and services that facilitate both development and deployment of biomedical models. Automated tools are essential to check the consistency of large ontologies. OWL DL reasoners, e.g. RacerPro, Pellet and Fact++, enable the automatic detection of inconsistencies caused by possible modelling errors, which may otherwise be difficult to identify. Moreover, the OWL-based models also enable semantic queries that allow these models to be discovered online. OWL also supports tools for ontology debugging and modularity. Debugging tools allow for tracing the reasons for inconsistencies identified by reasoners. Modularity tools allow for integrating and extracting different modules, e.g., extracting a sub-model of Lewis & Hudspeth model that describes the kinetic dynamics of calcium channel, for reusing it in other models. Consistency checking, debugging and modularity services are of primary importance for the development of models for systems biology.

Easy editing: By proposing a meta-model in OWL, we don't need to develop a specific model creation editor. We can use any OWL editor and open the meta-model as a model building template. Users create their own biological models by instantiating the pre-defined classes and specifying the values of pre-defined properties.

The ontology editor we recommend is protégé developed by Stanford University. Protégé is a free, open-source platform that provides a suite of tools to construct domain models and knowledge-based applications with ontologies. Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. The following figure illustrates a working environment of modeling creation by using our proposed meta-model for biological models.

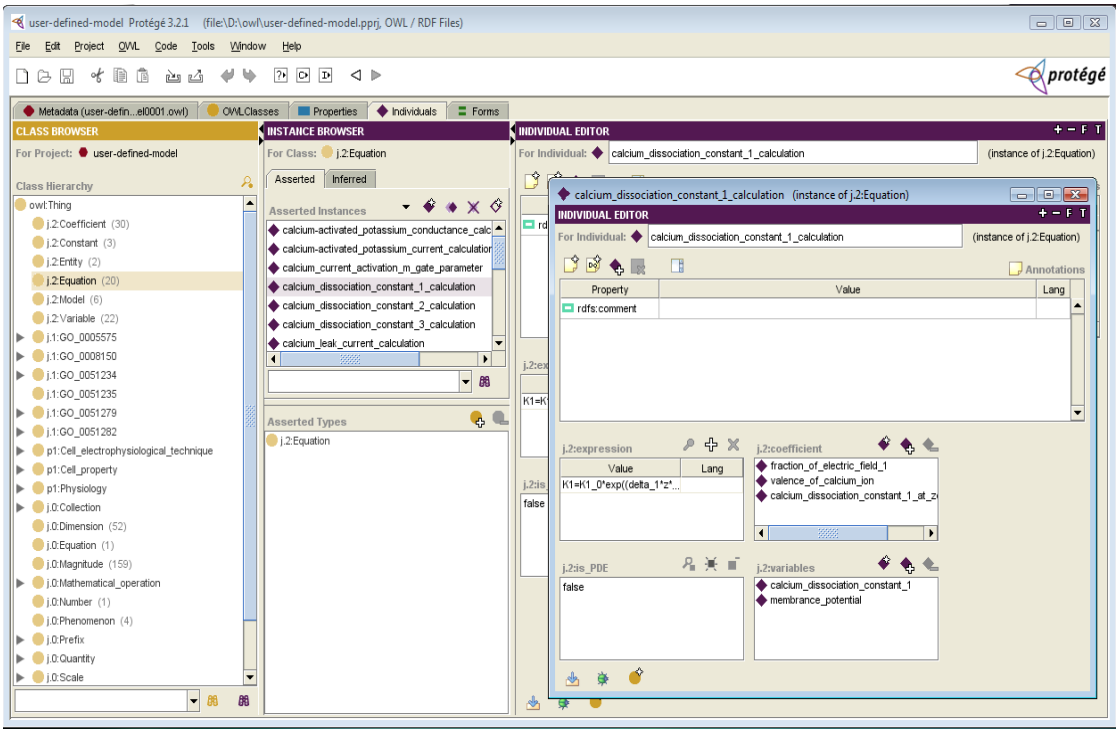


Figure 6.2 Using Protégé Ontology editor to construct biological models with the pre-defined meta-model

6.1.2 Meta-model for the Crucial Modeling Components

We design the meta-model to include the crucial modeling components described in SBML and CellML ontology. In this way, a subset of our model can be transformed into either SBML or CellML, in order to ensure the compatibility of our model with other SBML- or CellML-based scientific applications. 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.

SBML is a machine-readable language, based on XML, for representing models of biochemical reaction networks. SBML can represent metabolic networks, cell-signaling pathways, regulatory networks, and other kinds of systems studied in systems biology. It contains:

- **Function definition**: A named mathematical function that may be used throughout the rest of a model.

- **Unit definition**: A named definition of a new unit of measure, or a redefinition of an existing SBML default unit. Named units can be used in the expression of quantities in a model.
- **Compartment Type**: A type of location where reacting entities such as chemical substances may be located.
- **Species type**: A type of entity that can participate in reactions. Examples of species types include ions such as Ca^{2}, molecules such as glucose or ATP, binding sites on a protein, and more.
- **Compartment**: A well-stirred container of a particular type and finite size where species may be located. A model may contain multiple compartments of the same compartment type. Every species in a model must be located in a compartment.
- **Species**: A pool of entities of the same species type located in a specific compartment.
- **Parameter**: A quantity with a symbolic name. In SBML, the term parameter is used in a generic sense to refer to named quantities regardless of whether they are constants or variables in a model. SBML Level 2 Version 2 provides the ability to define parameters that are global to a model as well as parameters that are local to a single reaction.
- **Initial Assignment**: A mathematical expression used to determine the initial conditions of a model. This type of structure can only be used to define how the value of a variable can be calculated from other values and variables at the start of simulated time.
- **Rule**: A mathematical expression used in combination with the differential equations constructed based on the set of reactions in a model. It can be used to define how a variable's value can be calculated from other variables, or used to define the rate of change of a variable. The set of rules in a model can be used with the reaction rate equations to determine the behavior of the model with respect to time. The set of rules constrains the model for the entire duration of simulated time.
- **Constraint**: A mathematical expression that defines a constraint on the values of model variables. The constraint applies at all instants of simulated time. The set of constraints in model should not be used to determine the behavior of the model with respect to time.

- **Reaction**: A statement describing some transformation, transport or binding process that can change the amount of one or more species. For example, a reaction may describe how certain entities (reactants) are transformed into certain other entities (products). Reactions have associated kinetic rate expressions describing how quickly they take place.
- **Event**: A statement describing an instantaneous, discontinuous change in a set of variables of any type (species concentration, compartment size or parameter value) when a triggering condition is satisfied.

On the other hand, CellML specification consists of a very similar set of concepts, including unit definition, mathematical relations, parameters, etc. The difference between SBML and CellML is that CellML models use a class called ‘model’ as the root, and then encapsulate other elements in the class ‘component’. Each class ‘component’ contains a number of class ‘variables’, which must be declared by placing a variable element inside the component. Mathematical relationships between variables are expressed within ‘components’, using MathML. MathML is then used to make declarative expressions (as opposed to procedural statements as in a computer programming language).

By including the modelling components of SBML and CellML, the meta-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.

We will give the XML-view meta-model in full in appendix. Here in order to make it easy to understand, we present it in a UML diagram as below instead of using XML code:

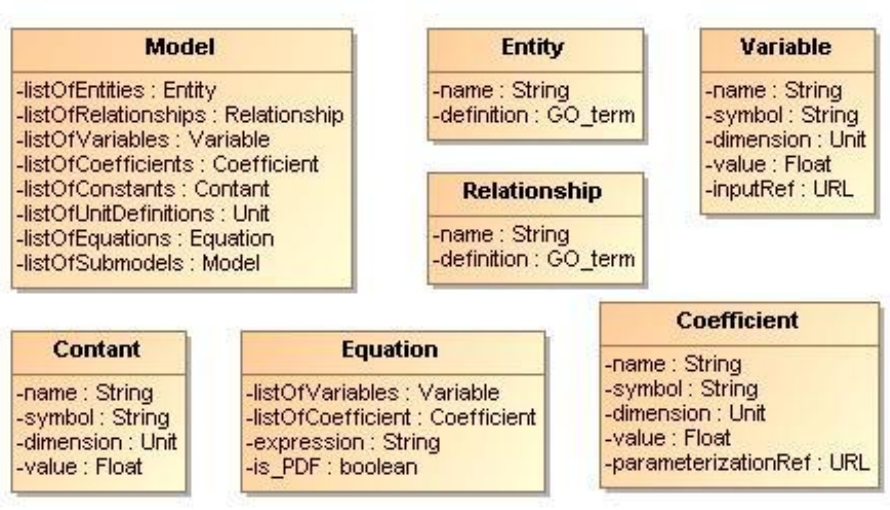


Figure 6.3 Class diagram to represent the metadata of the OWL-based biological model

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_intracelluar_calcium_ion’‘/>
      ..... <!--List of Equations -->
   <Equation rdf:ID=‘‘concentration_of_intracelluar_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 *U*i_Ca/(z*F*C_vol*xi)-K_S*c)
      </expression>
   </Equation>
</Model>
```

After the creation of this meta-model, researchers can then build biological models by loading the meta-model in ontology editors and instantiate its ontology entities with creator’s own knowledge.

We don’t attempt to define a universal language for representing quantitative models. A more realistic alternative is to acknowledge the diversity of approaches and methods being explored in systems biology, and seek a common intermediate format that enables communication of the most essential aspects of the models.

6.1.3 Meta-model Uses Shared Biological Ontologies for Instantiation

As we mentioned in the previous section, our meta-model allows importing of existing shared ontologies and then using the content of these imported ontologies to the current model creation as super- and sub-classes or value of properties. There is a shared ontology that is very helpful for defining biological entities and biological processes, i.e. the Gene Ontology (GO). GO provides the vocabulary for the description of many biological concepts such as the annotation of the molecular functions, biological processes, and cellular components of gene products. Therefore, GO ontologies are used as a de facto standard for semantically annotating biological information in the biological models.

For example, the following lines define the class ‘listOfEntities’ and define the value of any ‘Entities’ have to be in the range of biological entities defined by GO term GO:005575.

```
<owl:ObjectProperty rdf:ID="listofEntities">
          <rdfs:range
       rdf:resource="http://www.geneontology.org/owl#GO_0005575"/>
          <rdfs:subPropertyOf rdf:resource="#modelComponents"/>
          <rdfs:domain rdf:resource="#Model"/>
 </owl:ObjectProperty>
```

Then when instantiating the meta-model, a entity for describing voltage-gated calcium channel can be specified as the following:

```
<j.1:GO_0005891 rdf:ID="voltage-gated_calcium_channel_complex">
  <rdfs:comment xml:lang="en">lewis_hudspeth_model_20070202</rdfs:comment>
</j.1:GO_0005891>
<j.2:Entity rdf:ID="Voltage-gated_Calcium_Channel">
  <j.2:GO_term rdf:resource="#voltage-gated_calcium_channel_complex"/>
  <j.2:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Voltage-gated Calcium Channel</j.2:name>
</j.2:Entity>
```

The original GO definitions exist in online databases and can be retrieved by any users. For example, the GO term used above is defined as the following:

GO:0005891 voltage-gated calcium channel complex Back

A protein complex that forms a transmembrane channel through which calcium ions may pass in response to changes in membrane potential.

Term Information | Ancestor chart | Ancestor table | Child Terms | Protein Annotation | Statistics

ID	GO:0005891
Name	voltage-gated calcium channel complex
Definition	A protein complex that forms a transmembrane channel through which calcium ions may pass in response to changes in membrane potential.
Comment	

Synonyms

Type	Synonym
exact	voltage gated calcium channel complex
exact	voltage-sensitive calcium channel complex
exact	voltage-dependent calcium channel complex

XRefs

Database	ID
INTERPRO	IPR002077
GO	GO:0005891

Figure 6.4 An example of GO term (GO:0005891)

Using GO to instantiate the biological models enables semantic queries on the model elements by specifying the identity numbers of the GO terms or other GO term contents. For example, if one wants to search for a biological model that contains the element representing a voltage-sensitive calcium channel complex, one can first retrieve the GO ID number by querying the Gene Ontology, and then use the ID number to search and discover the model that we have defined.

6.1.4 Use Meta-model to generate computational simulations

Since our proposed meta-model includes the crucial elements in CellML and CellML to Java model transformation is available. It is not difficult to develop a transformation stylesheet for the transformation from OWL models to simulations. We will modify XSL stylesheet developed by Mathml-X project and then use it to scan the OWL source code and extracts all variables, wrapping them in getters and setters according to the JavaBeans specification. The equations are placed in the correspondent methods. Initial variable values are extracted and properly assigned. Some basic metadata is extracted and placed in the source code as Java comments.

6.2 Transforming Experimental Data into Semantic Web Services

In this section, we introduce the procedure of transforming computer-based data acquired from biological experiments into semantic Web Services. The purpose of this transformation is to standardize biological data storage that exists in various formats, allow both human and computer software agents to understand the content of the databases, and enable web-based access and semantic manipulation to the databases.

Transforming experimental data into database Web Services with semantic descriptions involves constructing the database from acquired data, coding the controller programs of the database, deploying the controller programs as Web Services, and creating semantic descriptions for these Web Services.

There are many ways to achieve the above procedure. In our case, in order to adapt the semantic Web Service infrastructure and apply the meta-model for biological modeling that has been proposed in the previous section, we will use a highly advanced programming environment called NetBeans. The NetBeans IDE is an open-source integrated development environment written entirely in Java that supports Web Service technology and its related standards.

NetBeans IDE supports development of all Java application types (J2SE, web, EJB etc.), and comes with integrated Java relational database and Java application server for Web Service deployment. Especially, the NetBeans IDE Bundle for Web & Java Enterprise Edition (Java EE) provides tools for building all Java EE components,

including Enterprise Java Beans (EJBs), web pages, servlets, and Web Services by including APIs for Java Persistence, EJB 3 and JAX-WS that implement Web Service related functionality. NetBeans therefore is able to simplify the proposed Web Service generation process.

For converting experimental data storage with various formats into relational database, we will use a generic SQL script. This SQL script will transform text-based data storing documents into Java relational database with a standard schema. This schema is designed to describe the content of data acquired from electrophysiological experiments since we use Lewis & Hudspeth case as our example. The schema is mostly generic while with a minor part of variation. This variation depends on the setting of experimentation and can be reflected by the information defined in the ontology-based biological model. We therefore can always use the ontology-based biological models to control the generation of the SQL script.

For building controller programs that are able to manipulate the relational database and can be deployed as Web Services, we will use Entity Class generation wizard provided by NetBeans that applies the relational database schema to create plain old java objects (POJOs) for retrieving and updating the content of the relational database. We will propose a set of generic Java methods which use these Java objects to achieve advanced manipulation of the databases. These methods will then be deployed as Java EE Web Services in the integrated application server in NetBeans.

The deployment of these Java database manipulation methods will automatically generate WSDL files that describe the services provided by the manipulation methods as network endpoints. This enables web-based semantic queries to the backend relational database.

Finally, we can build semantic descriptions on top of the generated WSDL files. The semantic descriptions are represented by OWL-S files, which are ontology-based. These OWL-S files can also be generated automatically by transforming the WSDL files. This transformation again will be controlled by the ontology-based biological model proposed in the previous section. With the OWL-S semantic descriptions, these database Web Services will then be able to connect with other components of our modeling framework.

In summary, the general procedure of the transformation are the following:

- Transform data source into relational database by generic SQL script;
- Build Java Entity Classes from the relational database schemas;
- Define generic operations that manipulate the Entity Classes to create Web Services;
- Create OWL-S semantic descriptions for the generated Web Services.

In the following sections, we will describe the transformation step-by-step.

6.2.1 Transform Data Source into Relational Database

In practice, experimental acquired data are either input by researchers or recorded and digitized by data acquiring devices. The data can exist in different formats of data-containing documents or can be stored in different types of relational databases. Meaningful data-containing documents are structured as datasheet and can always be converted into text-based spreadsheets, such as Microsoft excel files or CSV files, which can be used to exchange data between disparate applications.

In the case of Lewis & Hudspeth, by using whole-cell recording techniques, experimental data are saved in a spreadsheet which contains both the specifications of the experiment and the recorded electrophysiological data. The specifications of the experiment 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 the Lewis & Hudspeth case, the columns are defined as the following:

SignalsExported=Imemb,Vm

Signals=‘Imemb, ‘Vm, ‘Imemb, ‘Vm, ‘Imemb, ‘Vm’ …...

‘Time (s)’; ‘Trace #1 (pA)’ ‘Trace #1 (mV)’;‘Trace #2 (pA)’ ‘Trace #2 (mV)’;‘Trace #3 (pA)’ ‘Trace #3 (mV)’….

In which ‘Trace#’ is used to generalize the table structure so that any signals can be stored by the table. Each ‘Trace#’ is then specified according to the signal, such as membrane current (‘Imemb’) and membrane potential (‘Vm’).

This kind of structured datasheets can always be imported into relational database by using SQL statements. SQL is a standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database. SQL works with relational database programs like MS Access, DB2, Oracel, Derby (Java databse), etc.

Given an example of experimental recording with 8 sweeps in the Lewis & Hudspeth case, we can program the following SQL code for the data importing to a relational database:

```
CREATE TABLE MODELS (
MODEL_ID VARCHAR(24) NOT NULL,
SOURCE_URI  VARCHAR(128),
AUTHOR VARCHAR(24),
MODIFIED TIMESTAMP,
PRIMARY KEY (MODEL_ID)
);
```

Which creates the table that allows user to store the information such as model name, URI link of the literature reference for the model, the author(s), and the time of creation.

```
CREATE TABLE SETTING (
SETTING_ID VARCHAR(24) NOT NULL,
PROFILE  VARCHAR(128),
MEM_CAPACITANCE FLOAT (3),
MEM_RESISTANCE FLOAT (3),
ACCESS_RESISTANCE FLOAT (3),
TIMECONSTANCE FLOAT (3),
HOLDING_CUR FLOAT (3),
PRIMARY KEY (SETTING_ID)
);
```

Which creates the table that stores the laboratory settings for electrophysiology.

```
CREATE TABLE INDEXING (
SWEEPID SMALLINT NOT NULL,
VM VARCHAR(24),
IMEMB VARCHAR(24),
PRIMARY KEY (SWEEPID)
);
```

Which creates the table that stores the column names of the experimental datasheet.

```
CREATE TABLE DIMENSIONS (
```

```
DIMENSION_ID SMALLINT NOT NULL,
ENTITY VARCHAR(128),
DIMENSION VARCHAR(24),
PRIMARY KEY (DIMENSION_ID)
);
```

Which creates the table that stores the dimension of each column of the experimental datasheet.

```
CREATE TABLE DATASHEET (
TIME FLOAT(6) NOT NULL,
TRACE1A FLOAT (6),
TRACE1B FLOAT (21),
TRACE2A FLOAT (6),
TRACE2B FLOAT (21),
……
TRACE8A FLOAT (6),
TRACE8B FLOAT (21),
PRIMARY KEY (TIME)
);
```

Which creates the table that stores the actual experimental data.

```
INSERT INTO DIMENSIONS
   VALUES(1,'MEM_CAPACITANCE', 'pF'),
                              (2,'MEM_RESISTANCE', 'GOhms'),
                              (3,'ACCESS_RESISTANCE', 'MOhms'),
                              (4,'TIMECONSTANCE', 'ms'),
                              (5,'HOLDING_CUR', 'nA'),
                              (6,'TRACEnA', 'mV'),
                              (7,'TRACEnB', 'nA');
```

Which instantiate the ‘dimensions’ table with the values used by lewis & Hudspeth case.

```
CALL SYSCS_UTIL.SYSCS_IMPORT_TABLE (null,'DATASHEET','d:\hudspeth-ica.csv',null,null,null,0);
CALL SYSCS_UTIL.SYSCS_IMPORT_TABLE (null,'INDEXING','d:\hudspethindex.csv',null,null,null,0);
```

Where file 'hudspeth-ica.csv’ contains the actual recording datasheet, and hudspethindex.csv contains the names of the data columns. In this case, the indexing file is as simple as the following:

```
1,trace1a,trace1b
2,trace2a,trace2b
……
8,trace8a,trace8b
```

By using the above SQL statements, the acquired datasheet of Lewis & Hudspeth case can be imported into a relational database. In our case, we import the datasheet into a Derby database (Java database) in NetBeans.

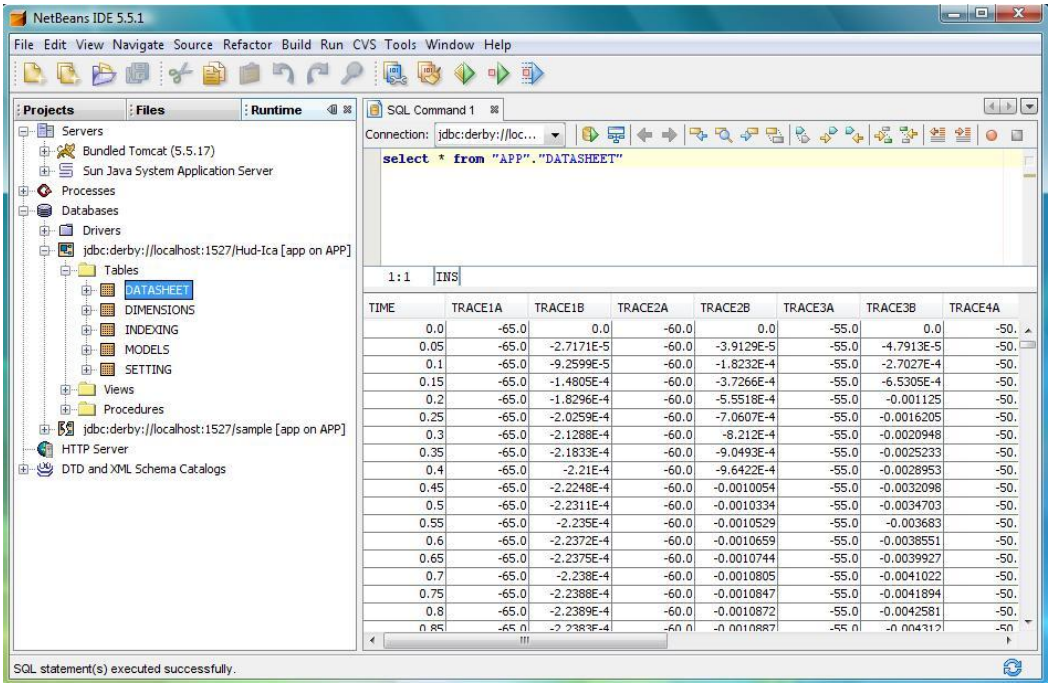


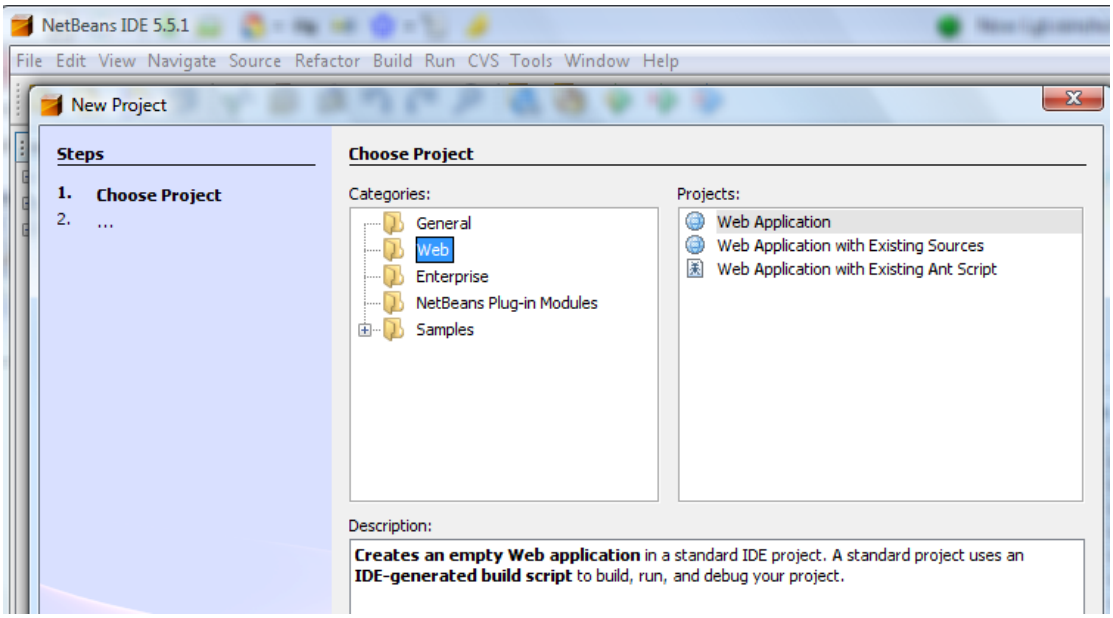
Figure 6.5 The result of importing experimental data into Java database in NetBeans IDE

NetBeans also allows the direct access to external relational databases by JDBC-to-ODBC driver. ODBC is short for Open DataBase Connectivity, a standard database access method to access any data from any application, regardless of which database management system handles the data. Java Database Connectivity (JDBC) is an API for the Java programming language that defines methods for querying and updating data in relational databases. JDBC together with a JDBC-to-ODBC Bridge driver enables connections to any ODBC-accessible data source in NetBeans.

6.2.2 Generate Java Entity Classes from Relational Database

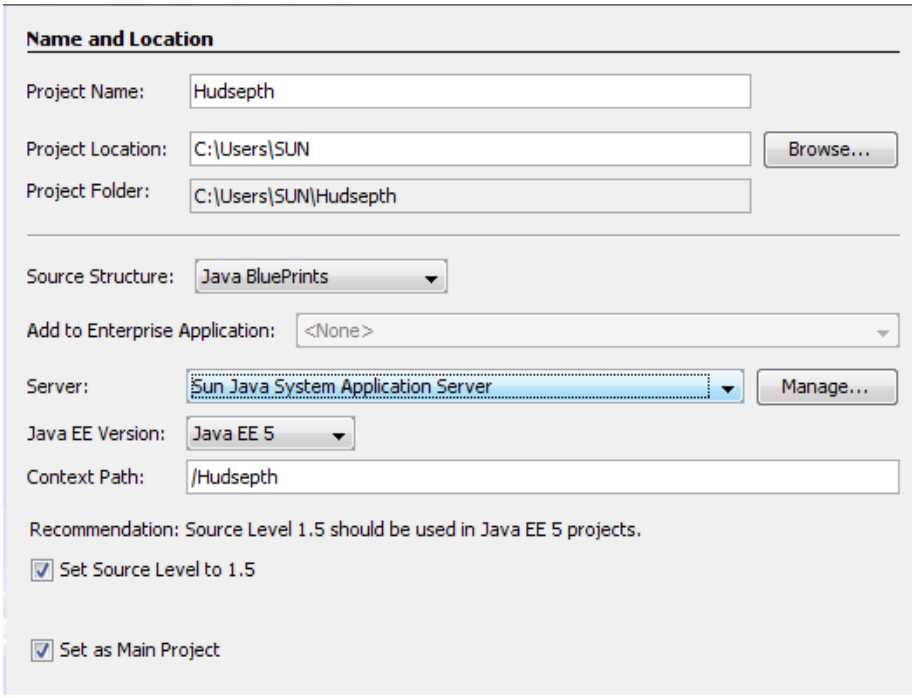
The generation of Java entity classes from relational databases is relatively simply as the NetBeans IDE has implemented a semi-automatic transformation function that retrieves the schema from the databases and applies XSL transformation engine to create Java POJOs (plain old Java objects). The general procedure is the following:

1. Creating a new Web Application project



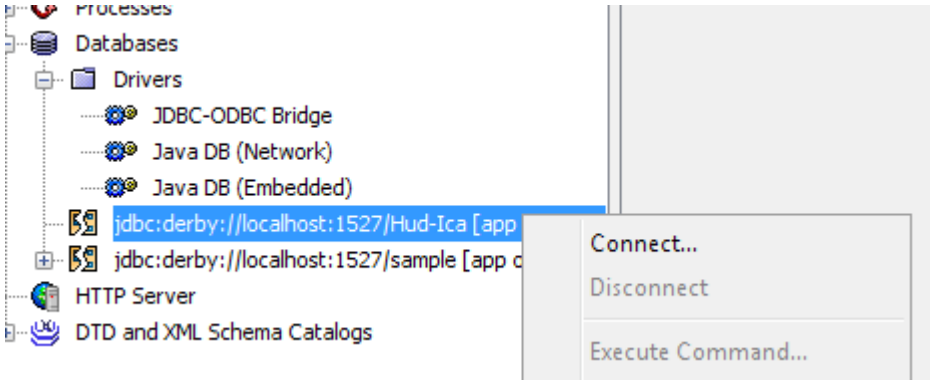
In order to be able to generate a Web Service at the end, we need to select the project type as 'Web Application'.

2. We will need to name the project and choose Sun Java Application Server for future deployment and Java EE 5 for the latest Web Service APIs.



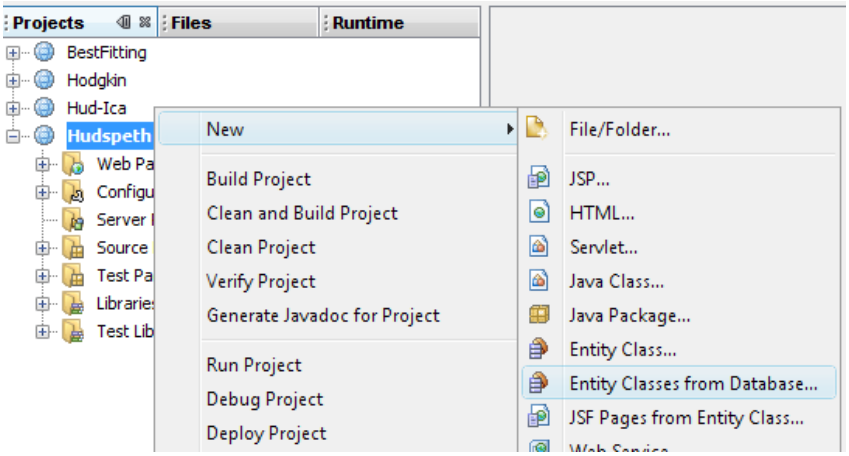
3. Connect to the database

To begin the generation of entity classes, we need connect to the database we have built from the previous step.

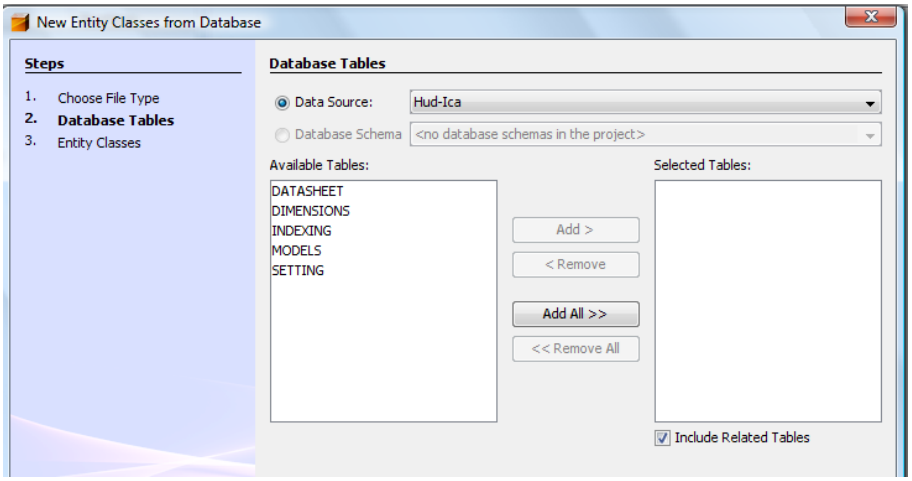


4. Generate entity classes from the connected relational database

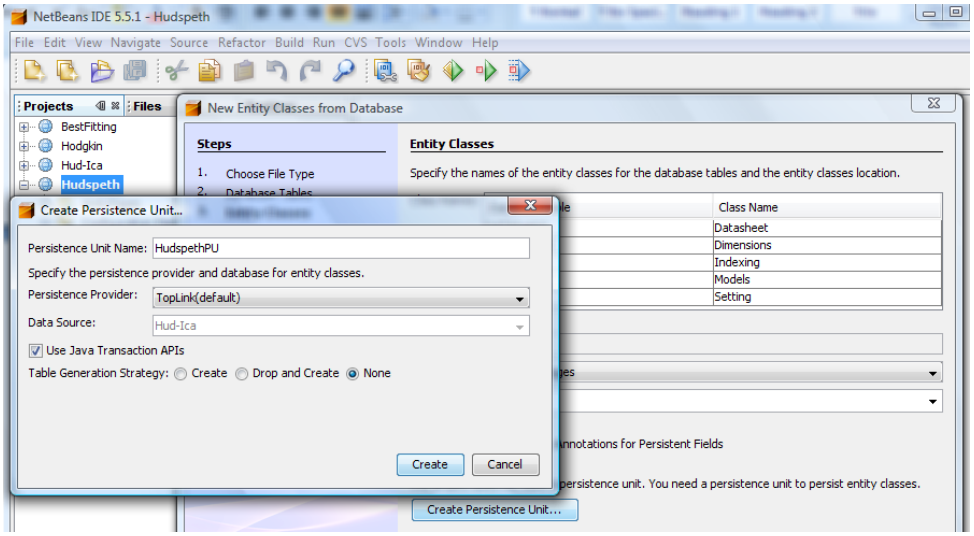
Right-click on the name of newly created project, a menu will show up and give the option called ‘Entity Classes from Database’.



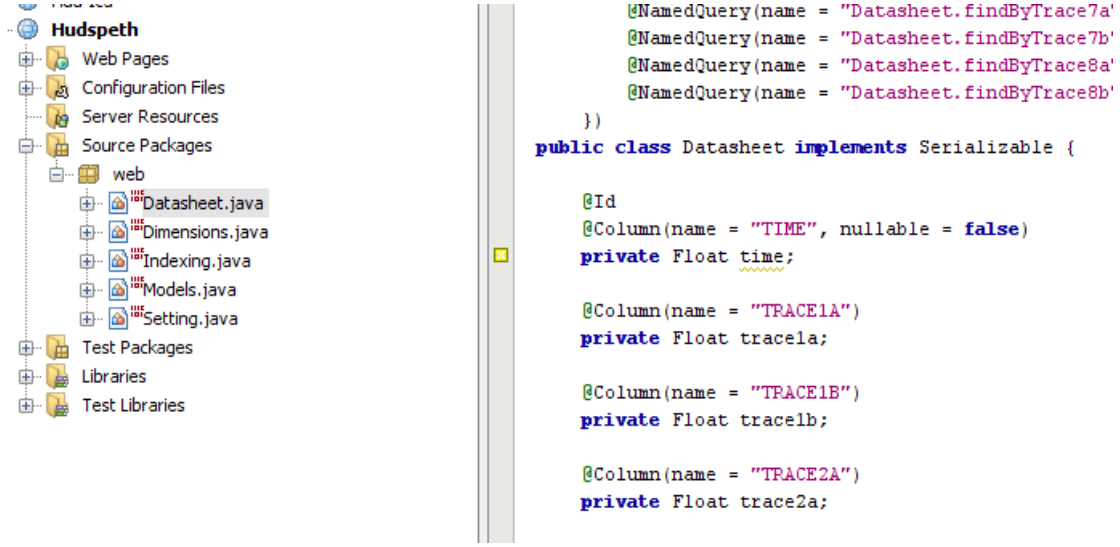
By selecting this option, a new window will come up that allows us to choose the data source, i.e. the relational database we have created. After choosing the data source, the names of the available tables will be retrieved automatically. In our case, we will then select ‘Add All’, so that there will be entity classes for each of the table.



Then the persistence unit needs to be created, so that the generated Entity classes will be able to recognize the database they link to. The name of Persistence unit will be used in the next section when creating database controller class that accesses Entity classes.



Finally the entity classes are automatically generated. Each table will create one POJO, and each column of the table will create one setter and one getter function for retrieving and updating the linked data source.



6.2.3 Generic Java Methods for Database Control

After the generation of entity classes from relational database schema, now we need to define a set of Java operations/methods to use these entity classes as their internal objects, so that these operations can have the ability to access the Java persistence database. What's more important, we need to deploy these methods as Web Services in order to enable web-based enquires to the database.

In order to control the generated entity classes, we need to import the following packages in Java EE 5:

```
import javax.annotation.Resource;
import javax.persistence.PersistenceUnit;
import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
```

Then we can use the following lines to create an EntityManager class. An EntityManager instance is associated with a persistence context (in our case, the relational database that stores the experimental data for Lewis & Hudspeth case). A persistence context is a set of entity instances in which for any persistent entity identity there is a unique entity instance. Within the persistence context, the entity instances and their lifecycle are managed. This interface defines the methods that are used to interact with the persistence context. The EntityManager API is used to create and remove persistent entity instances, to find entities by their primary key, and to query over entities.

```
public class ControlDatasheet {
    @Resource
    private UserTransaction utx;
    @PersistenceUnit(unitName = "HudspethPU")
    private EntityManagerFactory emf0;
    private EntityManager getEntityManager() {
        return emf0.createEntityManager();
    }
    ……
```

Where the @PersistenceUnit annotation define the link to the relational database by give the specific unit name as we define in the previous section. The getEntityManager method can then access to the database and returns an EntityManager object.

With the EntityManager, we are able to use Java Persistence query language statements as the input parameters to access the relational databases. The Java Persistence query language defines queries for entities and their persistent state. The query language allows you to write portable queries that work regardless of the underlying data store. The query language uses the abstract persistence schemas of entities, including their relationships, for its data model, and it defines operators and expressions based on this data model. The scope of a query spans the abstract schemas of related entities that are packaged in the same persistence unit. The query language uses a SQL-like syntax to select objects or values based on entity abstract schema types and relationships among them.

```
public String querySingleValue(@WebParam(name = "query")String query) {
    String inquery = "SELECT MIN(p.time) FROM Datasheet as p";
    if (query.charAt(0)=='S') inquery = query;
    return getEntityManager().createQuery(inquery).getSingleResult().toString();
}
```

As we can see, the above Java method use String "SELECT MIN(p.time) FROM Datasheet as p" as the parameter to the EntityManager and get a single value result from the database. The entry it queries is the one that has the minimum value in the 'time' column of table 'Datasheet'.

Similarly, we can define a collection of methods for database enquiry. The methods we defined only serve as samples of such kind of data manipulation operation and

are by no means comprehensive. Researchers can always expand this set of methods when necessary. The basic rule is to use formal Java persistence query language and Java entity classes to access the database at the backend.

After the creation of these methods, we can now deploy them as Web Service using JAX-WS API. JAX-WS stands for Java API for XML Web Services. JAX-WS is a technology for building Web Services and clients that communicate using XML. JAX-WS allows developers to write message-oriented as well as RPC-oriented Web Services. In JAX-WS, a Web Service operation invocation is represented by an XML-based protocol such as SOAP. The SOAP specification defines the envelope structure, encoding rules, and conventions for representing Web Service invocations and responses. These calls and responses are transmitted as SOAP messages (XML files) over HTTP.

In order to deploy Java methods as Web Service, we need to import the following packages:

```
import javax.jws.WebService;
import javax.jws.WebMethod;
import javax.jws.WebParam;
import javax.jws.WebResult;
import javax.jws.soap.SOAPBinding;
import javax.jws.soap.SOAPBinding.ParameterStyle;
```

Then by annotating the method as the following, it can then be deployed in the application server.

```
@WebMethod
@WebResult(name="qValue")
public String querySingleValue(@WebParam(name = "query")String query) {
	String inquery = "SELECT MIN(p.time) FROM Datasheet as p";
	if (query.charAt(0)=='S') inquery = query;
	return getEntityManager().createQuery(inquery).getSingleResult().toString();
}
```

6.2.4 Semantic Description for Data Web Service

One of the crucial steps to build Web Services that fulfil the Semantic Web vision is to define a semantic layer on top of the service layer. This enables users to locate, select, employ, compose, and monitor Web-based services automatically.

In this section, we will briefly introduce the relationship between non-semantic web description WSDL and semantic web description OWL-S and how to use pre-defined ontology to generate OWL-S from WSDL files. The generated OWL-S file will then be used for semantic-based Web Service discovery and composition. We will give more details above Web Service composition in the next section.

When deploying non-semantic Web Services, a service description file can be generated automatically by JAX-WS API based on the endpoints defined by the programming implementation and describe their message content and binding details. This description is generated in the web standard format called Web Services Description Language (WSDL).

Web Services Description Language (WSDL) specifies a protocol- and encoding-independent mechanism for Web Service providers to describe the means of interacting with offered services. WSDL is an XML vocabulary which describes network-reachable services and maps these to a messaging-capable collection of communication endpoints. WSDL separates the abstract definition of service and messages from their concrete binding to a network port and message format. Services are defined using six major elements:

- **Types**, which provides data type definitions used to describe the messages exchanged.
- **Message**, which represents an abstract definition of the data being transmitted. A message consists of logical parts, each of which is associated with a definition within some type system.
- **PortType**, which is a set of abstract operations. Each operation refers to an input message and output messages.
- **Binding**, which specifies concrete protocol and data format specifications for the operations and messages defined by a particular portType.
- **Port**, which specifies an address for a binding, thus defining a single communication endpoint.
- **Service**, which is used to aggregate a set of related ports.

OWL-S is OWL-based and allows for the description of a Web Service in terms of a *Profile*, which tells "what the service does", a *Process Model*, which tells "how the service works", and a *Grounding*, which tells "how to access the service" (See the figure below).

The Profile and Process Model are considered to be *abstract* specifications, in the sense that they do not specify the details of particular message formats, protocols, and network addresses by which a Web Service is instantiated. The role of the grounding is to provide these more *concrete* details. OWL-S uses WSDL to ground OWL-S services, which is based on the fact that OWL-S' concept of grounding is generally consistent with WSDL's concept of *binding*. As a result, it is a relatively straightforward task to ground an OWL-S atomic process.

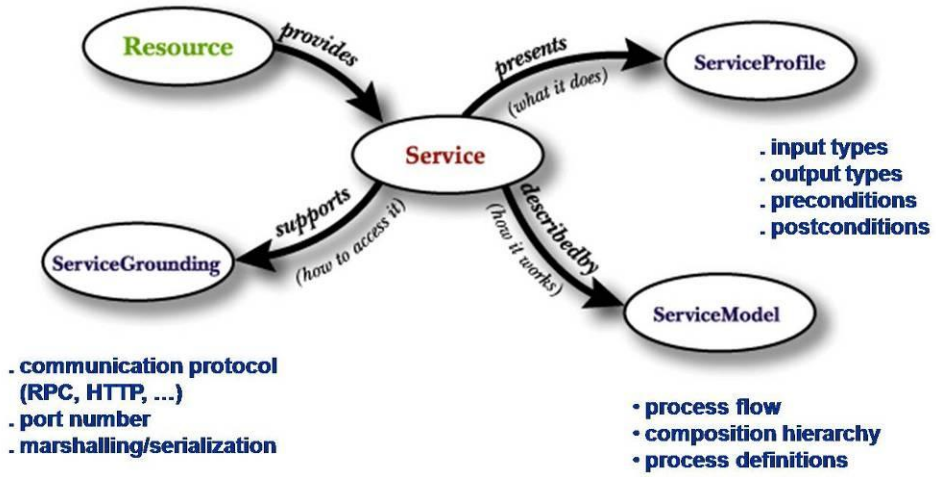


Figure 6.6 OWL-S upper ontology (IAM, 2005)

In the Lewis & Hudspeth case, deploying the database Web Service within Java EE platform will automatically generate a web-based WSDL document. This document is available online while the service is active. A sample access link can be:

http://localhost:8080/Hudspeth/ControlDataSheetService?WSDL

Where ‘Hudspeth’ is the project name when building the database service in Java in NetBeans and ‘ControlDataSheet’ is the class defined in the project to be deployed as Web Service. The OWL-S will then describe the service and WSDL grounding accordingly.

For example, the semantics for describing the service grounding can be

```
<grounding:WsdlAtomicProcessGrounding
rdf:ID="ControlDatasheetProcessGrounding">
<grounding:owlsProcess rdf:resource="#ControlDatasheetProcess"/>
        <grounding:wsdlDocument
        rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
                http://localhost:8080/Hudspeth/ControlDatasheetService?WSDL
</grounding:wsdlDocument>
……
```

Comparing to WSDL format, OWL-S is rather complicated. It is therefore difficult to generate OWL-S by automation. For example, the following well-formatted OWL-S code represents the service description and service profile description for Lewis & Hudspeth case. We can see that the value of each service class is different.

```
<!-- Service description -->
<service:Service rdf:ID ControlDatasheetService">
        <service:presents rdf:resource="# ControlDatasheetProfile"/>
        <service:describedBy rdf:resource="# ControlDatasheetProcess"/>
        <service:supports rdf:resource="#& ControlDatasheetGrounding"/>
</service:Service>

<!-- Profile description -->
<mind:InformationService rdf:ID="& ControlDatasheetProfile">
        <service:isPresentedBy rdf:resource="# ControlDatasheetService"/>
        <profile:serviceName xml:lang="en">
Hudspeth:ControlDatasheet.manipulateSheet Service</profile:serviceName>
        <profile:hasInput rdf:resource="# MethodString "/>
        <profile:hasOutput rdf:resource="# ReturnValues "/>
</mind:InformationService>
```

We've found a way to standardize these description values by using Document Type Definition. A DTD is primarily used for the expression of a schema via a set of declarations that conform to a particular markup syntax and that describe a class, or type, of document, in terms of constraints on the structure of that document. DTD defines the document structure with a list of legal elements and attributes. We use DTD Entities in this case, which are variables used to define shortcuts to standard text or special characters. For example, we define DTD Entities for the Lewis & Hudspeth case as the following:

```
<!DOCTYPE uridef [
……
  <!ENTITY jprojectName "Hudspeth">
  <!ENTITY jclassName "ControlDatasheet">
  <!ENTITY jmethodName "manipulateSheet">
  <!ENTITY jparaIn1 "method">
  <!ENTITY jparaReturn "manipulatedValues">
```

```
  <!ENTITY owlsInput1 "MethodString">
  <!ENTITY owlsOutput "ReturnValues">
  <!ENTITY wsdl "http://localhost:8080/Hodgkin/&jclassName;Service?WSDL">
]>
```

Then the body of the OWL-S document can be standardized as the following and can be reuse by many other cases:

```
<!-- Service description -->
<service:Service rdf:ID="&jclassName;Service">
        <service:presents rdf:resource="#&jclassName;Profile"/>
        <service:describedBy rdf:resource="#&jclassName;Process"/>
        <service:supports rdf:resource="#&jclassName;Grounding"/>
</service:Service>

<!-- Profile description -->
<mind:InformationService rdf:ID="&jclassName;Profile">
        <service:isPresentedBy rdf:resource="#&jclassName;Service"/>
        <profile:serviceName
xml:lang="en">&jprojectName;:&jclassName;.&jmethodName;
Service</profile:serviceName>
        <profile:hasInput rdf:resource="#&owlsInput1;"/>
        <profile:hasOutput rdf:resource="#&owlsOutput;"/>
</mind:InformationService>
```

In order to pass array type input to Web Services, we need to define OWL class for array in advance in order to make it available to OWL-S descriptions. We need to define array type for all kinds of data type, such as double array, Boolean array, and so on. The following is an example of the definition of double array:

```
  <owl:Class rdf:ID="ArrayOfDouble">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"
        >1</owl:minCardinality>
        <owl:onProperty>
          <owl:DatatypeProperty rdf:ID="hasDouble"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
```

```
    <owl:Class rdf:ID="Array"/>
  </rdfs:subClassOf>
  <rdfs:label xml:lang="en">Double Array</rdfs:label>
 </owl:Class>
```

Since soap-based Web Services only return string type values, we also need to use type transformation to change the input/output into the right types. The following is an example of such transformation:

```
<grounding:xsltTransformationString>
        <![CDATA[
            <xsl:stylesheet version="1.0"

    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"

    xmlns:array="http://www.ucl.ac.uk/~ucbpzsu/array-ont.owl#"

    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

    xmlns:owl="http://www.w3.org/2002/07/owl#"
        xmlns="urn:uk:ac:ucl:www">
      <xsl:template match="//ManipulateDatasheet">
        <manipulateSheet>
          <method
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
            <xsl:value-of select="method"/>
          </method>
        </manipulateSheet>
      </xsl:template>
        </xsl:stylesheet>
    ]]>
</grounding:xsltTransformationString>
```

6.3 Transforming Analysis Methods to Semantic Web Services

We use JAX-WS API 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., can be 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 the semantic description for the data analysis Web Services by using OWL-S.

6.4 Web Service Composition

After having experimental data and mathematical methods transformed into Web Services and annotated with semantic descriptions, we are then able to orchestrate them together by defining workflow among them. A few graphical tools have been developed to facilitate creating workflow among semantic Web Services. A good example is 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). It can be used to guide users in the dynamic composition of Web Services by supporting OWL-S 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 $\alpha_m=\alpha_0 \, e^{(-(Vm+V0)/kB)+KA}$ 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 analyzer 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 analyzer 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 $\alpha_m=22800e^{(-(Vm+V0)/33)+510)}$. When all the equations in a model are parameterized, 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 files 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.

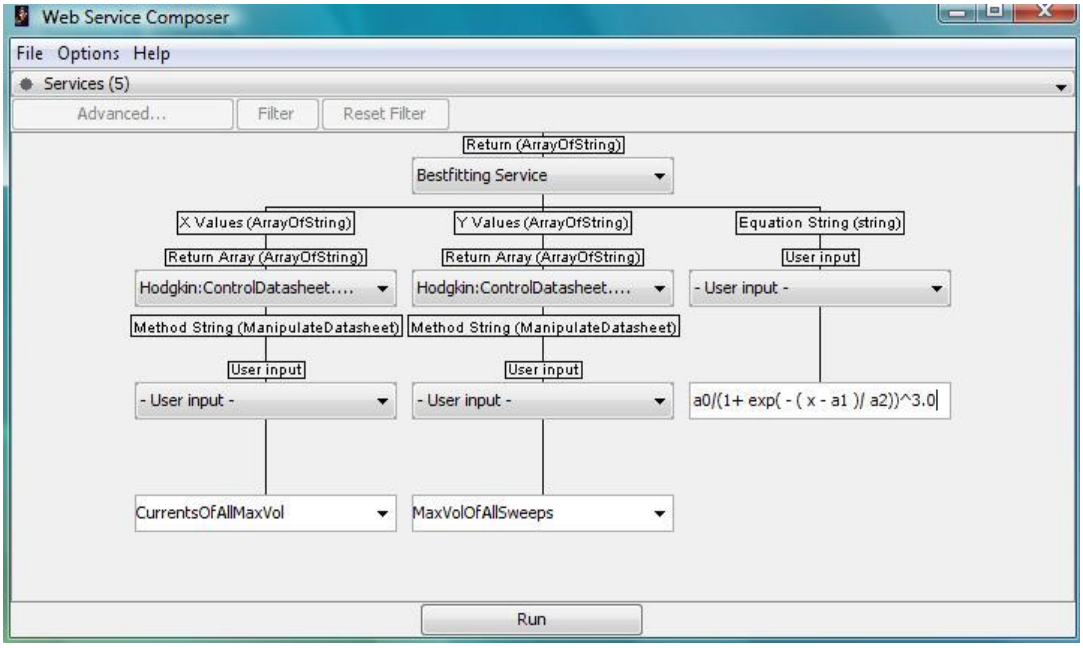


Figure 6.7 Compose data Web Service and data analyzer Web Service together by using service composer application

6.5 Generate Simulation by using Web Service Composition Models

By conducting all the above procedure, we constructed a fully specified ontology model for the biological system of interest, i.e. the model of hair cell electrophysiology. All parameters interactively specified are annotated with external links to OWL-S files, which represent the composition of database Web Services and data analysis Web Services used for parameterization. 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 Simluation Environment (CESE). A simulation is then generated automatically by transforming CellML model to JavaBeans programs.

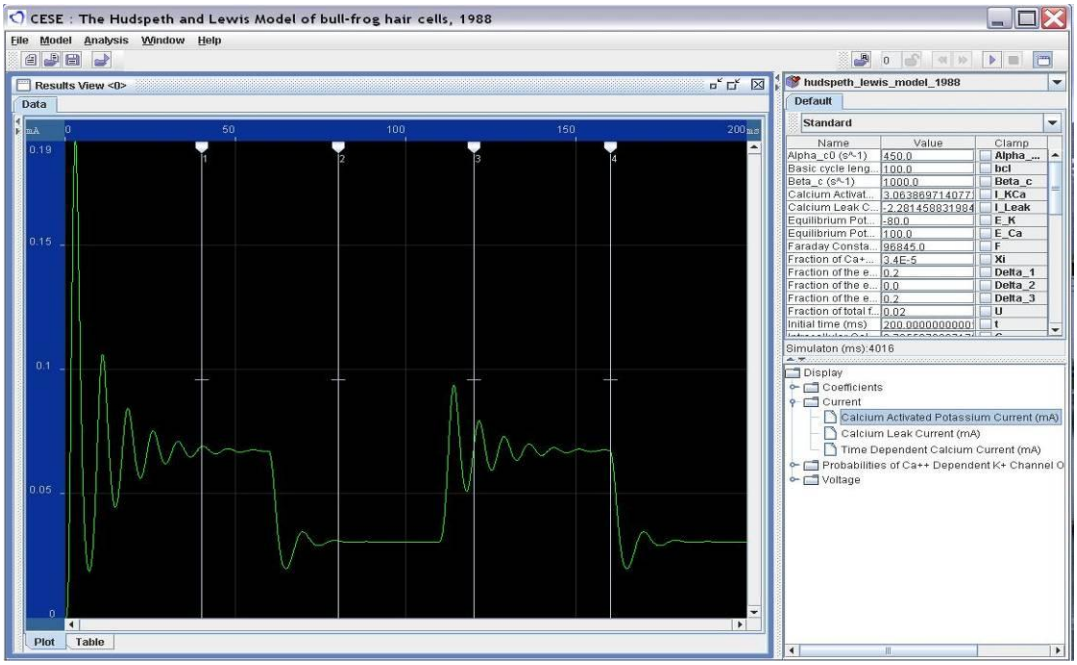


Figure 6.8 Simulation of electrical resonance in hair cells

Figure 6.8 shows the simulation result of electrical resonance described by equation $I_{K(Ca)} = g_{K(Ca)} \cdot (V_m - E_{K(Ca)})$ in the model of voltage oscillations in a frog hair cell.

6.6 Summary

In this chapter, we have given the step-by-step description of the proposed framework. First, we have describe the OWL-based meta-model for building biological models that is used as a template for model creation in systems biology. This meta-model enables linking biological models with external shared ontologies for biological term annotation, and serves as a basis to guide the transformation from experimental data sources and mathematical analysis methods to Semantic Web Services, and embeds references of external data sources, data analysis methods, and parameterization profiles for further reuse.

Then, we have introduced the procedure of transforming biological data sources into Semantic Web Services. After that, we have described the transformation from data analysis methods into data analyzer Web Services. Furthermore, we have introduced how to use OWL-S ontologies to defining the workflow among related semantic Web Services, in order to describe the parameterization processes for biological modeling. Finally, we have introduced the automatic transformation from instances

of the proposed OWL-based biological models to computational simulations. All the programming implementation of this framework is managed as a sourceforge project at:

https://sourceforge.net/projects/biologicalmodel/

In the following chapter we will evaluate our proposed framework by using the case studies of biological modeling we have introduced in chapter 4.

Chapter 7 : Framework Evaluation by Case Studies

We have seen how the proposed framework for modeling in systems biology is constructed in the previous chapter. The major components of the framework include the OWL-based meta-model that contains ontological entities and relationships for formally representing various knowledge in biological modeling, the procedure for transforming distributed biological data source and data analysis methods into Web Services, and OWL-based ontologies that defines these Web Services and enables service composition.

In this chapter, we will evaluate the proposed ontology-centered framework by applying the case studies in biological modeling that have been introduced in chapter 4. We will examine how well the framework is able to realize the goals that we intend to achieve. We want to demonstrate that by using this framework, we are able to facilitate many aspects of biological modeling in practice, including automatic model discovery, model reuse and composition, semi-automatic reconfiguration of model parameterization, and automatic transformation to simulations.

7.1 Model Discovery

Previous effort of developing biological modeling formats, such as CellML and SBML, is focused on providing formal representation that includes the main elements involved in the modeling processes. These formats, despite provide readability to both human and computers, do not have proper functionality to connect with shared biological ontologies, nor do they have logic layers that enable inference to respond to semantic queries.

In our framework, the meta-model for biological modeling is defined in OWL format, more precisely OWL-DL specification, thus it is able to fulfill these requirements.

Using shared ontologies

In our meta-model, we define the class *model* has properties called *listOfEntity* and *listOfRelationship*, whose values are instances of terms in Gene Ontology (GO). The GO project has developed three structured controlled vocabularies (ontologies) that describe biological processes, cellular components and molecular functions in a

species-independent manner. Using GO is not compulsory to instantiate our meta-model. Instead we use GO only because it's one of the most developed biological ontologies that have OWL format available, and also it provides sufficient support to the modeling in our case studies. Other commonly shared formal ontologies, such as those in OBO project, can be used as well.

Here we use the Hodgkin and Huxley case to demonstrate that by using the meta-model to build biological models, the model and its components are able to be annotated with ontological terms from shared biological ontologies. The description to be given is generic and therefore applies to all the case studies.

Step 1: In order to be able to use a certain shared ontology to annotate many biological models in the same way, it is necessary to import the OWL-formatted shared ontology as part of the meta-model. The model components can then be annotated with the instances of the biological terms from the shared ontology.

OWL models allow the declaration of the importing of the shared ontologies. The syntax of ontology import is as the following:

<owl:imports rdf:resource="http://www.geneontology.org/owl"/>,

where http://www.geneontology.org/owl is the URI of Gene Ontology.

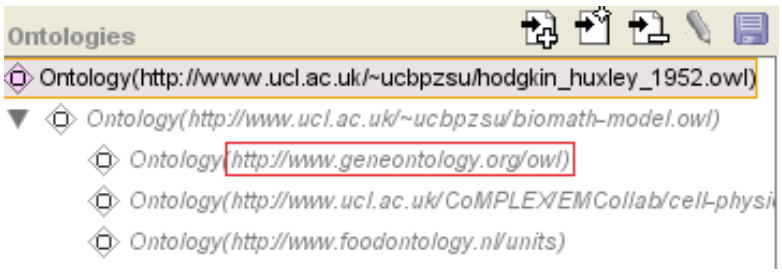


Figure 7.1 Gene Ontology is imported by proposed meta-model 'biomath-model.owl' using Protégé OWL ontology editor

Due to the large size of the actual GO ontologies, in our case studies we used a trim-down version of GO and imported it locally. Ontology importing can be done by giving their URI address. The ontology editor will be able to access the ontologies when connected to the internet.

Step 2: Instantiate a term of interest from the imported ontology

The current ontologies provides by the GO project are molecular function, biological process, and cellular component. Using GO to annotate models is equivalent to register the biological models to a comprehensive indexing system which classifies models by the defined relations among biological entities that uses shared terminology throughout.

For example, in the Hodgkin and Huxley case, the biological model contains description of the kinetic properties of voltage-gated sodium channels. The entity "voltage-gated sodium channel complex" is defined in GO as in Figure 7.2 and shared among all biologists who want to use it.

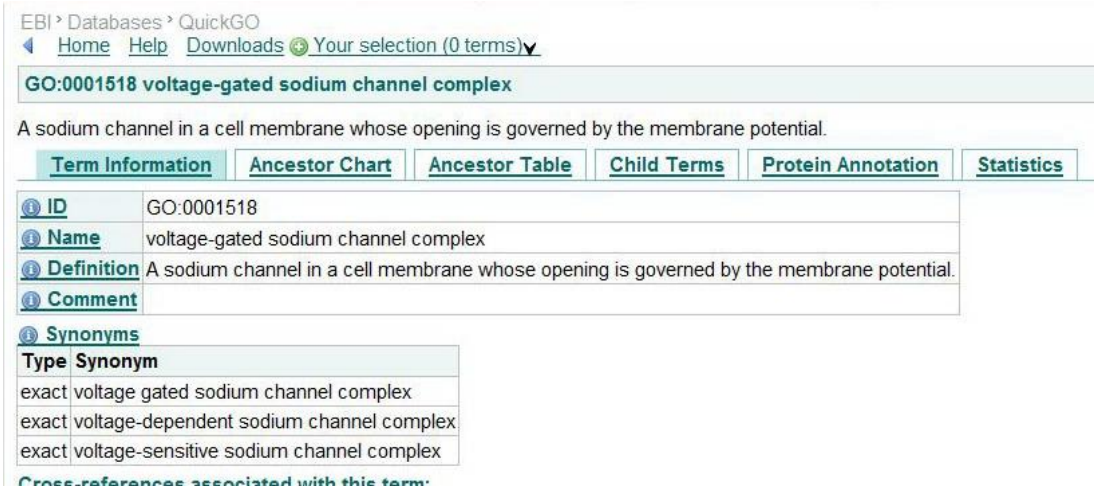


Figure 7.2 Entity "voltage-gated sodium channel complex" in Gene Ontology

By having the ID number of this entity in Gene Ontology, we are able to find the same entity within the imported OWL-based ontology file.

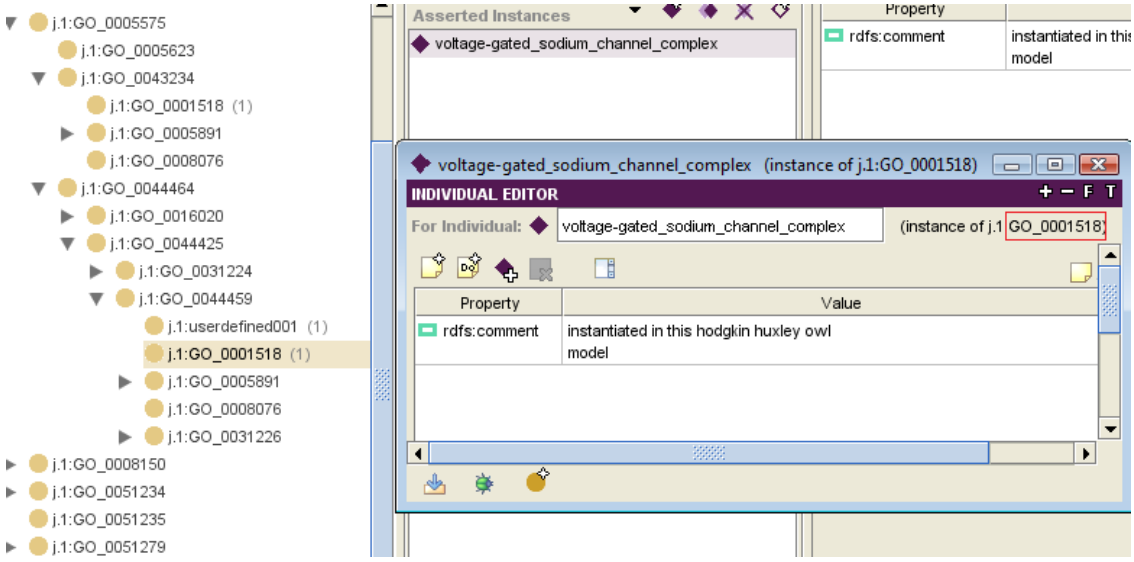


Figure 7.3 Looking up the entity in the imported shared biological ontology and instantiate it

We are then able to instantiate the entity for further use when building biological models.

Step 3: Annotate the model by including the instances of shared ontological entities

This step is rather straightforward, since our meta-model has defined the *model* class with the property that can use the instances of shared ontological entities as its value.

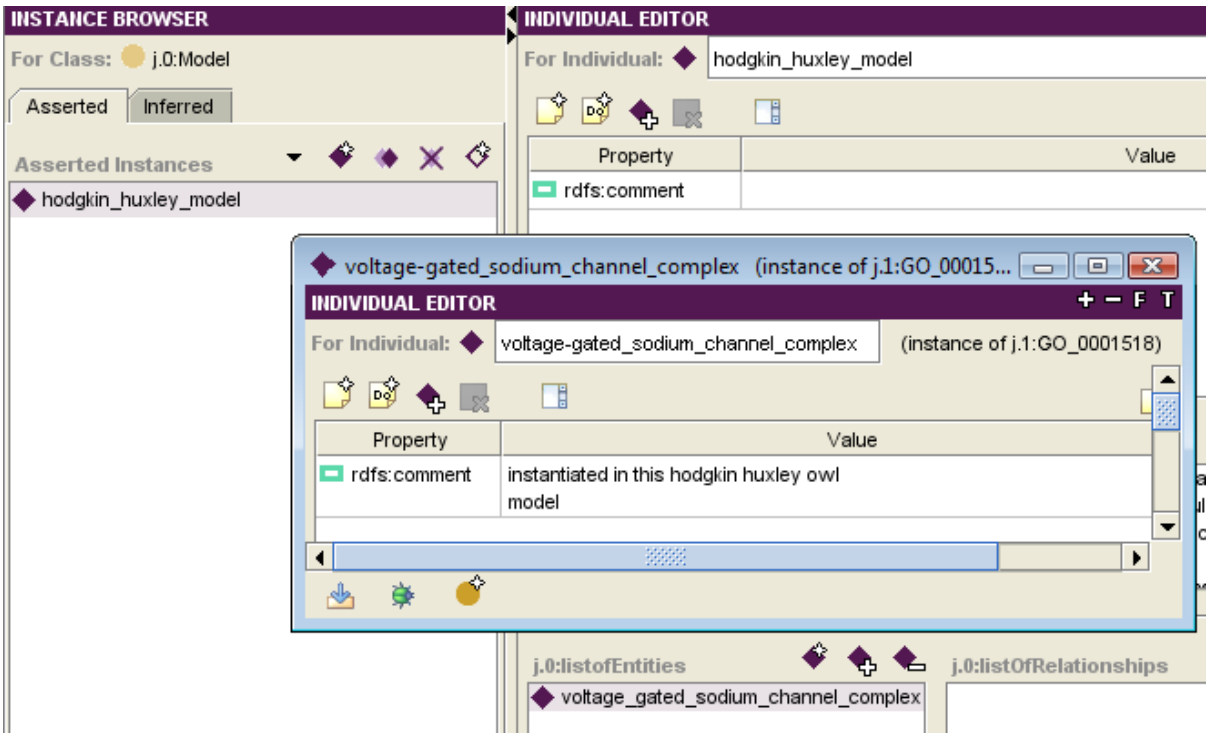


Figure 7.4 Use the instances of shared ontological entities in the biological model

By annotating our OWL model for Hodgkin and Huxley case with the term “voltage-gated sodium channel complex” from GO, the model is in fact registered to an indexing tree in the ontology database (see Figure 7.4).

By examining Figure 7.4, we can see that the “voltage-gated sodium channel complex” is the terminal branch of the “ancestor tree” inferred from the Gene Ontology. It has terms such as “cation channel complex” and “membrane part” as parent classes. If a semantic query is made to search for the models that involve the description of “cation channel” or “membrane part” in a cell, this model will then be able to give the right answer.

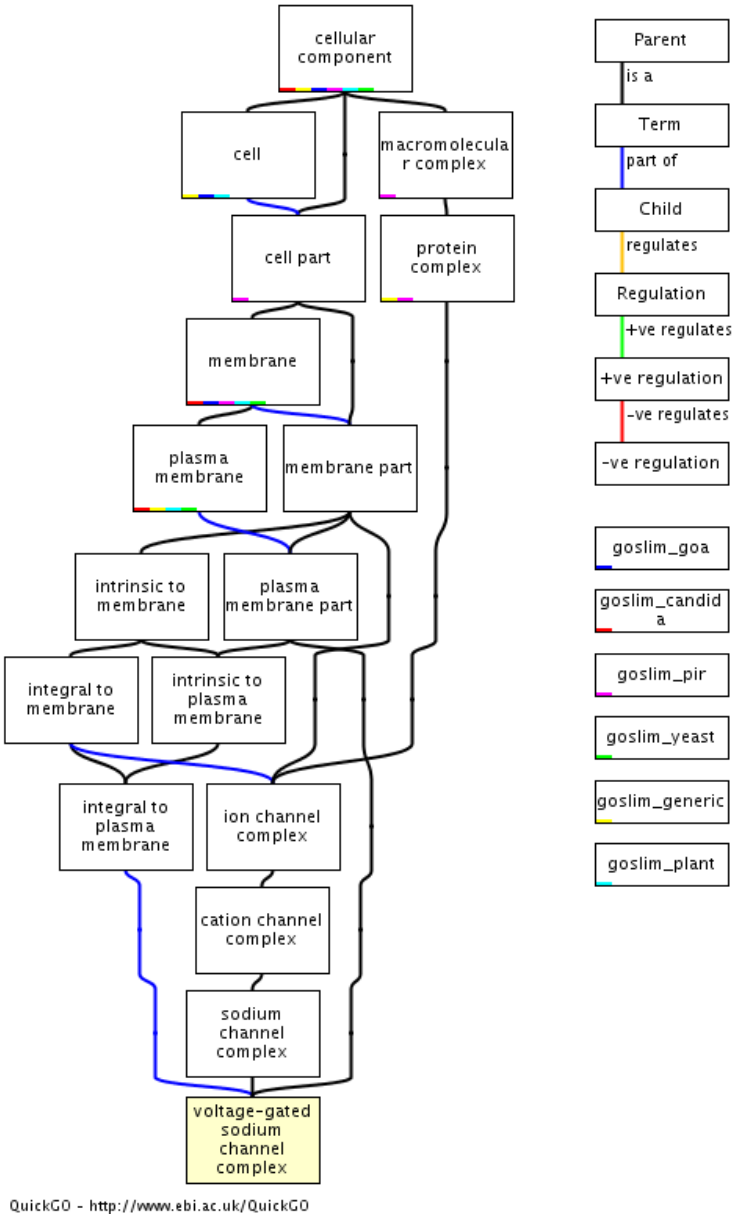


Figure 7.5 The "ancestor tree" that has the term "voltage-gated sodium channel complex" as the terminal branch (QuickGO browser)

Make Semantic Queries

In order to demonstrate that the biological models built by using our OWL-based meta-model are able to support semantic queries, we developed a simple querying facility. The implementation is provided in the package *AgentModel* within the sourceforge project at:

https://sourceforge.net/projects/biologicalmodel/

The implementation of this facility is built on top of the Java Agent Development Framework (JADE). JADE is a software framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. The agent platform can be distributed across machines and the configuration can be controlled via a remote GUI.

The querying language we used is RDQL (RDF Data Query Language). RDQL has been implemented in a number of RDF systems for extracting information from RDF graphs. An RDQL consists of a graph pattern, expressed as a list of triple patterns. Each triple pattern is comprised of named variables and RDF values (URIs and literals). An RDQL query can have a set of constraints on the values of those variables, and a list of the variables required in the answer set. Since OWL format is designed on top of RDF specification, the RDQL is able to query OWL files in the same way.

The syntax of RDQL is as follows:

SELECT ?x WHERE (?x base:listOfEntities ?y), (?y rdf:type j.1:GO_0001518)

In which "base" (specified in \complex\code\Config.properties) is the namespace of the OWL biological model template http://www.ucl.ac.uk/~ucbpzsu/biomath-model.owl

This semantic query is simply searching an RDF/OWL file if there is any instance of class 'listOfEntities' has the value of a GO ontology term with ID number 0001518, which represents the biological entity "voltage-gated sodium channel complex".

In our implementation, we deploy two agents in JADE:

1) Answer Agent: contains the biological models we built as the content of its knowledge and broadcast it through network.

2) Ask Agent: instead of having the specific knowledge of the biological model owned by Answer Agent, knows the OWL meta-model that we proposed, and makes RDQL queries through network. Since the OWL meta-model is shared among all the agents, the validity of the queries is guaranteed.

We have also built a simple GUI for users to make queries without the knowledge of how to use JADE framework. The GUI (showed in figure 7.7) can make queries of several GO terms by submitting the GO ID number to the Ask Agent and the Ask Agent will then use the GO ID submitted as part of the query and then subscribe to any agent who has the knowledge of such GO term. When an agent within the agent framework gives positive response and returns the queried knowledge, the Ask Agent will then pass the result back to our GUI and display in the dialogue box.

This implementation is only a simple demonstration to present the basic ideas involved in our case. It is not compulsory to use agent framework to publish the biological models. This can be done in many other ways such as using Web Services or mapping OWL models into RDF databases. Researchers can build more sophisticated software similar to the GUI we provide for more user-friendly interfaces.

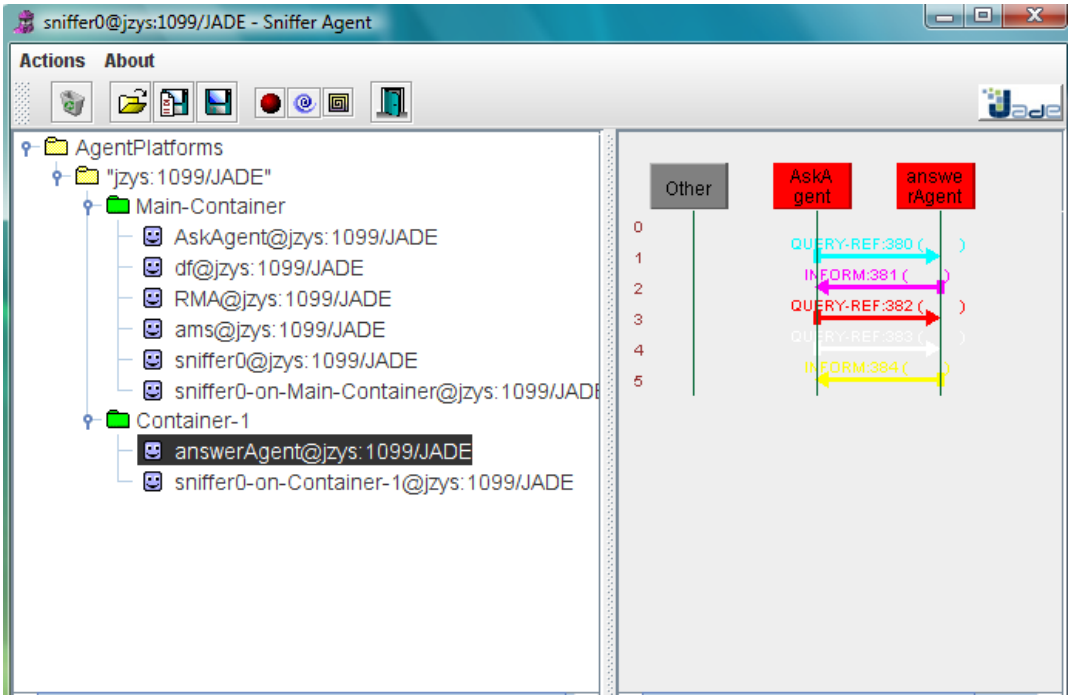


Figure 7.6 Using JADE to deploy agents and monitor their activities. In this case, the AskAgent asked AnswerAgent a few times and receives responses

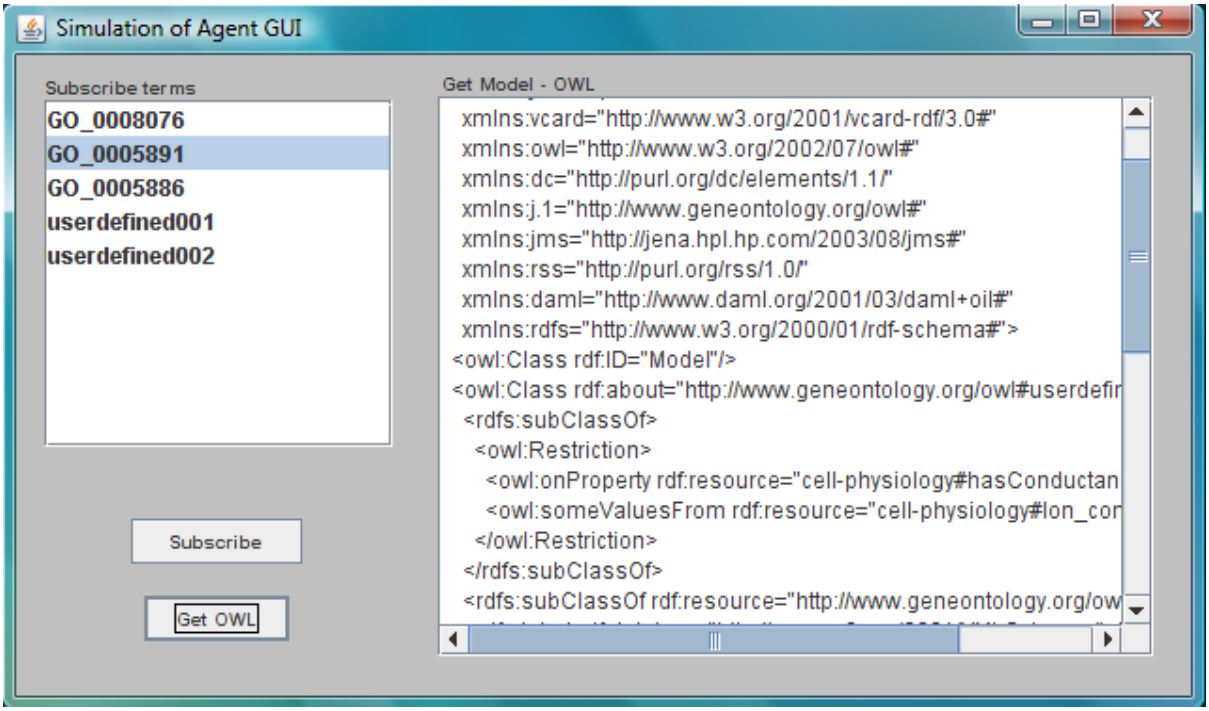


Figure 7.7 The GUI for user to make queries without dealing with the actual framework

7.2 Model Reuse

Formal knowledge representation for biological modeling shall not only be able to describe all the components of models and the processes of modeling, it is also crucial to achieve great reusability for further processing such as model composition or partly rebuilt, and enable maximal automation and dynamism in all aspects of modeling in systems biology.

It is very often that different biological phenomena occurred in various biological systems and species share the same underlying mechanism. Therefore, the ability to reuse the knowledge within existing models is of great value.

For example, in Hodgkin and Huxley model, the conductance of potassium current is described as proportional to the fourth power of a variable which obeys a first-order equation. It can be represented by the following equations:

$$g_K = \bar{g}_K n^4 \qquad \frac{dn}{dt} = \alpha_n (1-n) - \beta_n n$$

where $\bar{g}_K$ is a constant, and n is a dimensionless variable that varies from 0 to 1. α_n is the rate of closing of the channels and β_n is the rate of opening. Together, they give the total rate of change in the channels during an action potential.

α_n and β_n are described as functions of membrane potential:

$$\alpha_n = 0.01(V+10)/(e^{(V+10)/10}-1) \qquad \beta_n = 0.125e^{V/80}$$

Similarly, in Lewis and Hudspeth model, when it comes to describe voltage-dependent Ca^{2+} current activation, the similar scheme can be reused. The scheme is modified to a third-order kinetic scheme as the following:

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

in which $\bar{g}_{ca}$ is the limiting value of Ca^{2+} conductance when all Ca^{2+} channels are open, m is the time-dependent value of the activation parameter, *Vm* is membrane potential, and E_{ca} is the Ca^{2+} equilibrium potential. The activation parameter, m, varies between zero and unity with time,

$$\frac{dm}{dt} = \beta_m (1-m) - \alpha_m m$$

in which α_m and β_m are respectively the *m* gate's closing and opening rate constants.

When building these two models, a model reusing mechanism will be very helpful. The reusability of existing biological modeling formats such as CellML and SBML is however primitive.

In the section, we will demonstrate that using our OWL meta-model to build biological models will offer convenience to model reuse.

The reusability of our OWL models is based on the fact that all of them share the same modeling template. Therefore, all the model components are easy to be copied while preserving the modeling knowledge in the existing models. Since we have defined our *model* class to have property of *modelReference*, the reference

information of which models have contributed to the construction of the new model can be properly documented.

We have investigated the model reuse case by reusing part of the Hodgkin and Huxley model to build the Lewis and Hudspeth model. The procedure is as follows:

First, we create a new blank OWL file and import the Hodgkin and Huxley model.

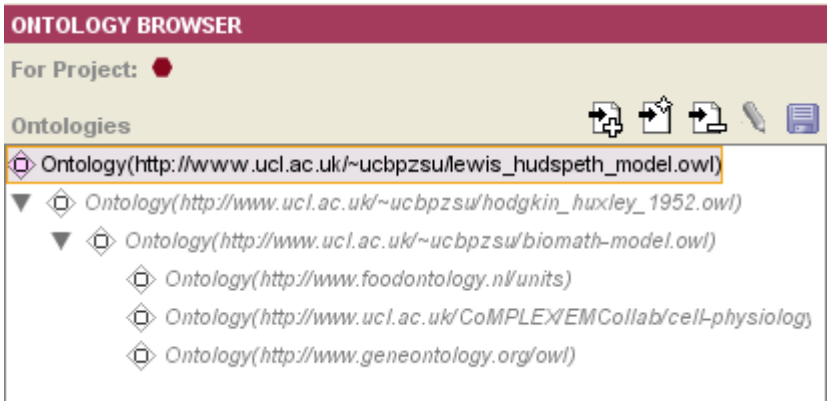


Figure 7.6 Import the Hodgkin and Huxley model into a blank model

Then, we copy all the models and equation that are needed for the new Lewis and Hudspeth model.

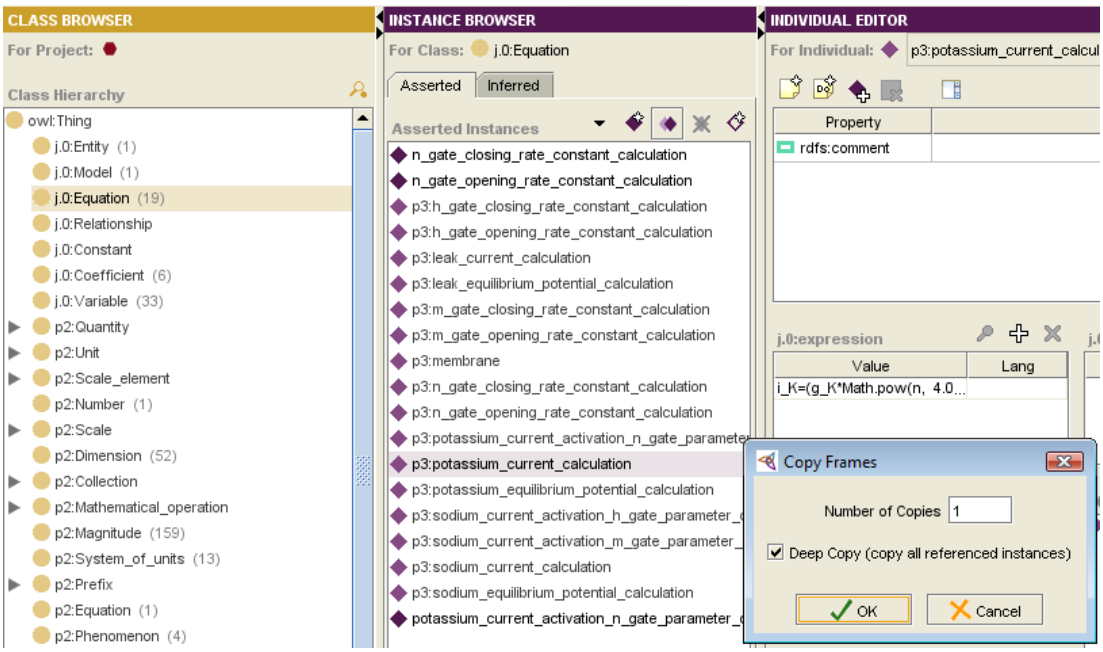


Figure 7.7 Make copies of the instances of models and/or equations

After that, we remove the model that has been reused from the importing declaration and import our meta-model instead.

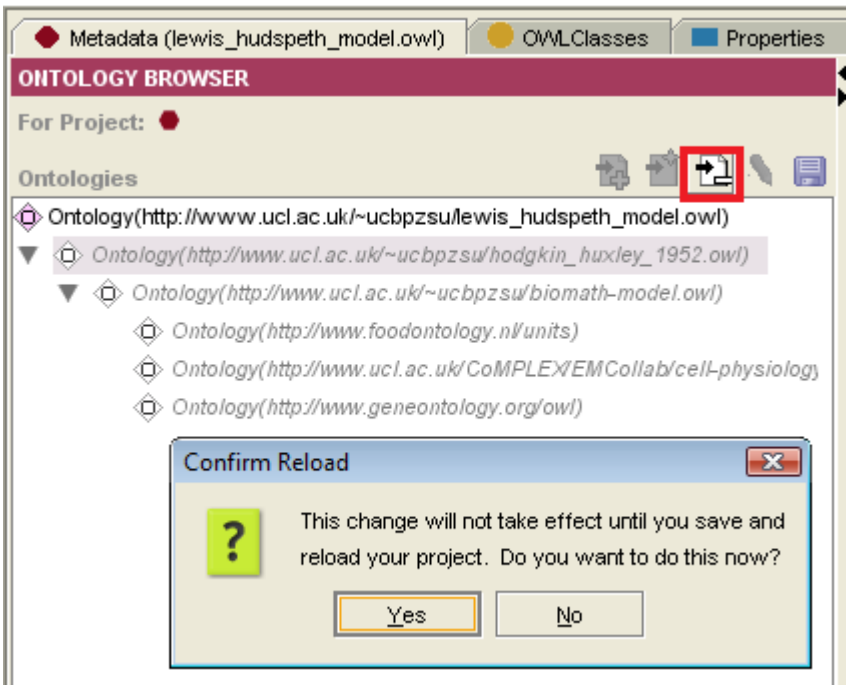


Figure 7.8 Remove the model from the importing list

Figure 7.9 Import the meta-model

Then, we will have a model built upon the meta-model and only have the ontological individuals we need. At this point, we update the value of *modelReference* to the URI of the model we have reused.

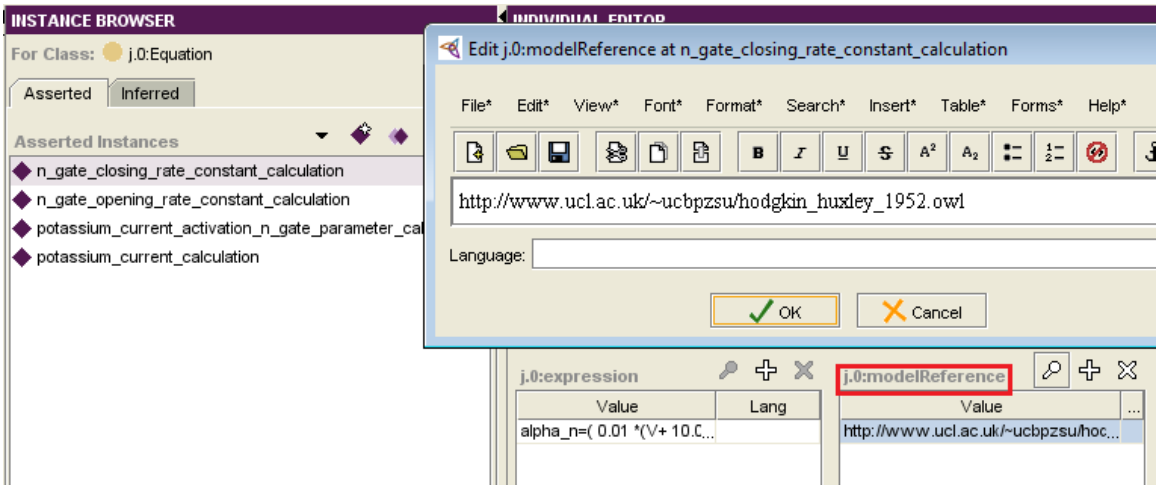


Figure 7.10 New model with model reference information

Based on this, we can then modify the reused part of the model to the exact model we meant to construct.

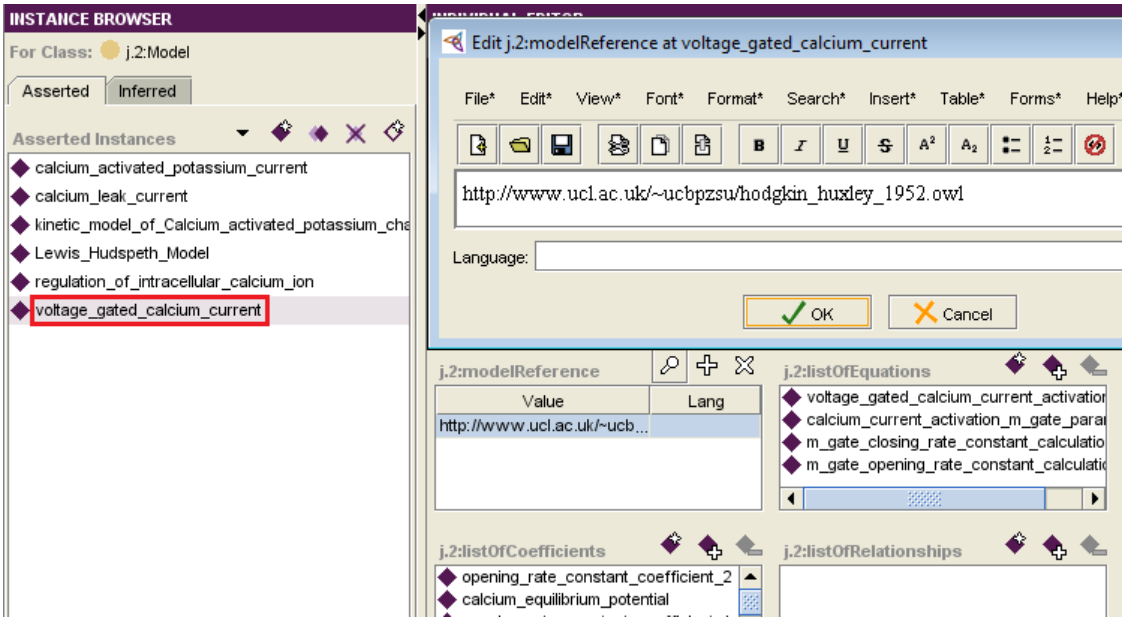


Figure 7.11 The final Lewis and Hudspeth model that has reused the Hodgkin and Huxley model to build the "voltage-gated calcium current" sub-model

One of the major benefits of model reuse is the increase of model traceability. This means that users can always trace back to the foundation models that have been used to construct the new models. Sometimes it happens that knowledge in the old models becomes outdated or updated. Models might evolve in different ways and concepts might be refined or redefined. Under these circumstances, consistencies among models can be checked by using shared ontologies, where outdated knowledge is

stored and distinctively marked (e.g. in GO outdated terms are marked as ‘obsolete’). Therefore, the evolution of ontologies will not discontinue the traceability of modeling knowledge.

7.3 Automatic generation of simulations

In the study of systems biology, one of the crucial processes is to abstract experimental biologists’ observations into scientific models, qualitatively analyze the relationship among the entities in the domain of interest, and/or quantitatively describe the entity-relationship network, parameterize the descriptions of the system with experimental data for simulation, compare the results against experimental observations for validation, and then give instructive feedback to both model construction and experimental observations.

The development of biological models therefore is an iteration process and the fast generation of computational simulations can be of great help to shorten the life cycle of such development.

We have developed XSLT transformation engine for the proposed framework, which can translate the biological models built by using the OWL meta-model into simulation directly. Also, in order to demonstrate that our model are back-compatible with the existing biological modeling effort such as CellML, we have built transformation engines to generate CellML models from OWL-based biological models as the interim format between OWL models and simulations.

The automatic generation from OWL-based biological models to simulations has been tested in all the case studies including Hodgkin&Huxley model, Lewis&Hudspeth model, Hofer’s calcium oscillation model, G-protein kinetics model, and the composite model of G-protein mediated calcium oscillation model.

We have written the transformation engines that convert these OWL models to CellML models, and then convert them into JavaBeans programs used in CESE as simulations.

With the ability of generating simulations from biological models automatically, it is extremely simple and efficient to test the validity of model components and parameterization.

7.4 Model Composition

In the study of systems biology, the ability to compose biological models is crucial in systems biology as systems biology is the 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.

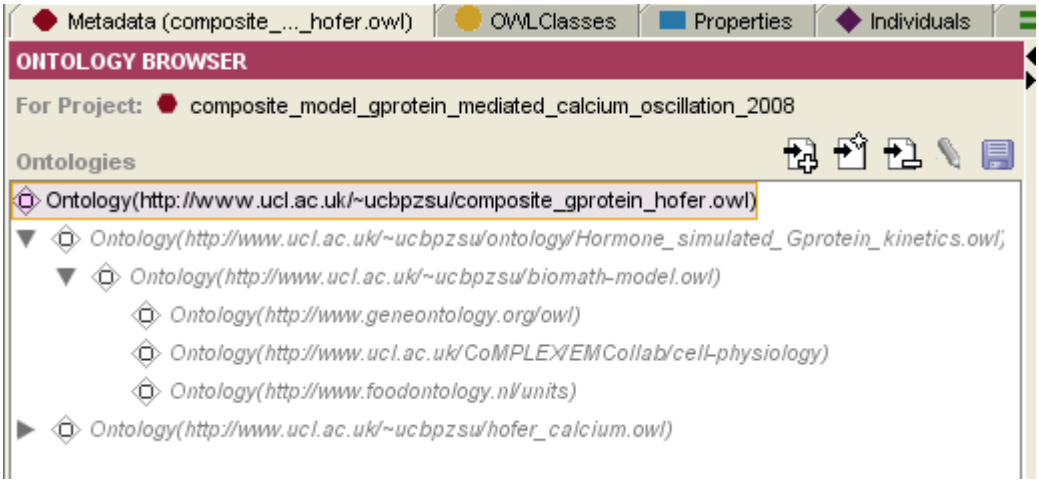


Figure 7.12 Import G-protein kinetics model and calcium oscillation model to create a composite model

We have examined the possibility to use the proposed OWL meta-model to facilitate model composition. The case used is the modelling of hormone-induced calcium signalling. During the evaluation, first we built two separate models, one describes the hormone induced G-protein kinetics, and the other describes the minimum model of calcium oscillation induced by IP3. We then connect these two models by importing both models into a blank OWL model and build the connecting equations for the composite model and the necessary information override to the sub-models.

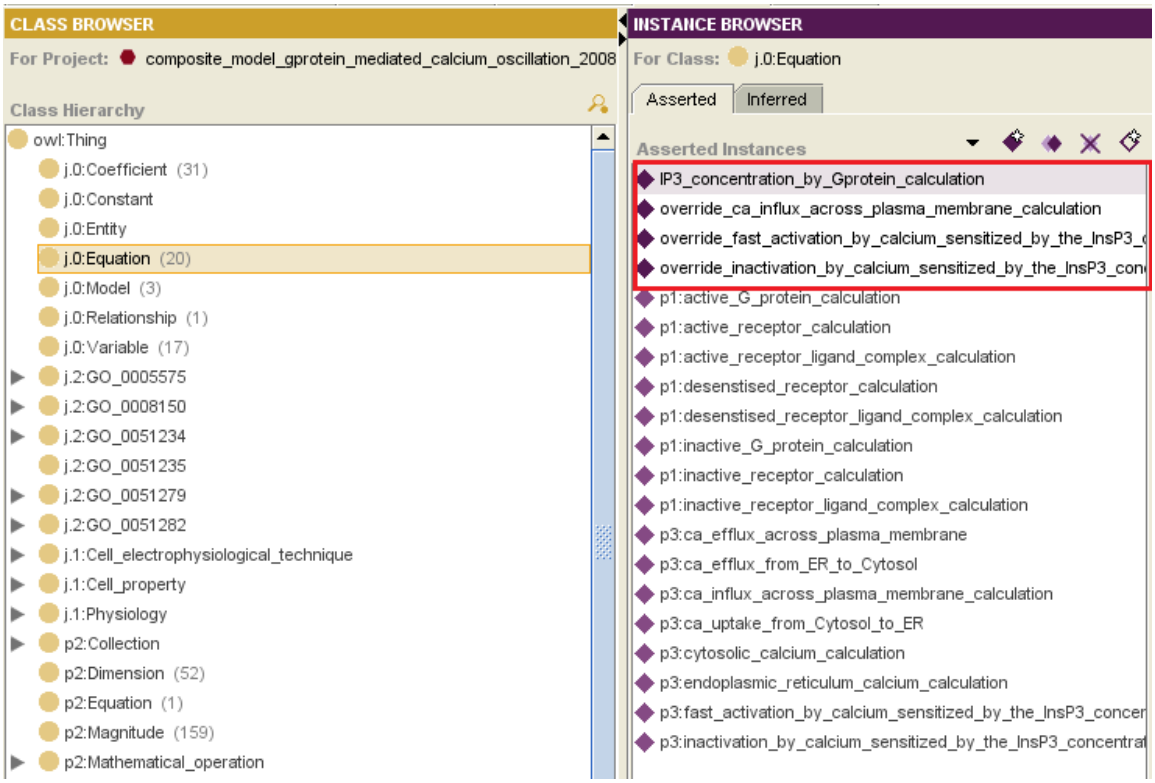


Figure 7.13 Build the complete composite model by describe the connector equations and update/override the equations that need to be modified

We have also built the XSLT transformation engine for composite models to generate simulation.

Due to some unexpected problems in the CESE (we identify that CESE uses a specific PDE solver that doesn't fit the case), we couldn't get the automatically generated simulations running in CESE to demonstrate the results. Therefore, we converted the OWL models to MATLAB simulations as an alternative of examine the model composition ability of these models.

Figure 6.17 shows the simulation result of hormone-induced G-protein activation model, where X-axis is the time ranging from 0 to 500 milliseconds and Y-axis is the percentage of G-protein that have been activated by hormone binding to the receptors on the membrane.

Figure 6.18 is the result of the minimum model of cytosolic oscillation introduced in section 4.3.2.

Figure 6.19 illustrates the result of the model composed by the above two models.

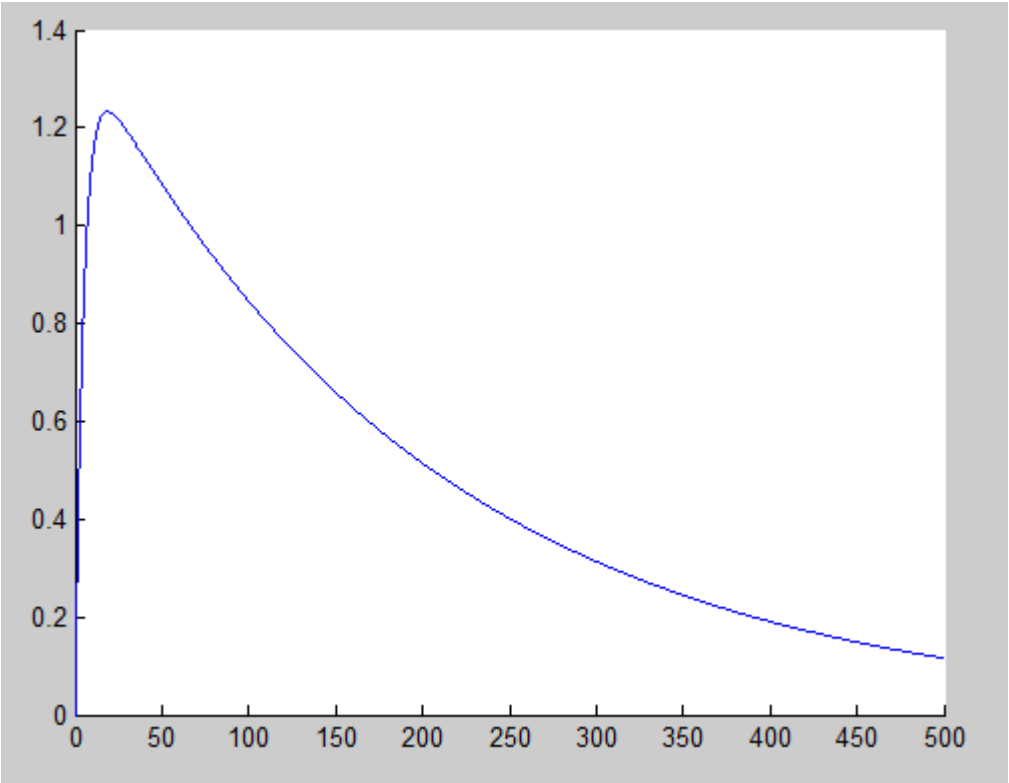


Figure 7.14 Simulation result of Hormone-induced G-protein activation in MATLAB

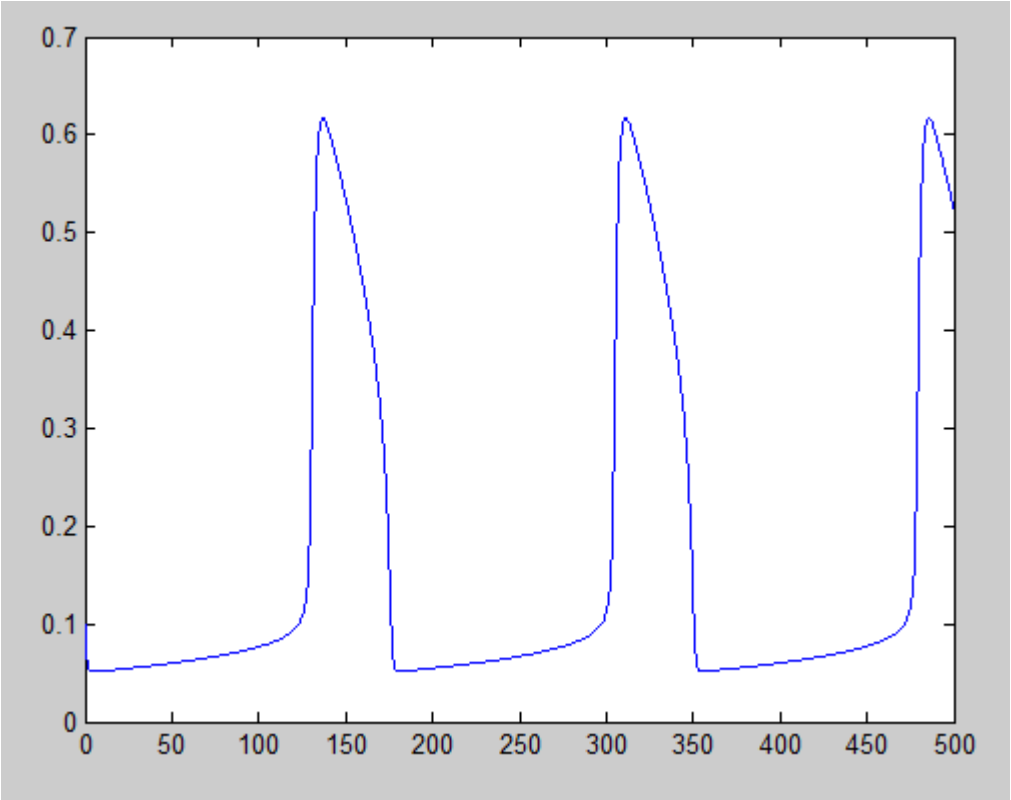


Figure 7.15 Simulation result of the minimum model for cytosolic calcium oscillation

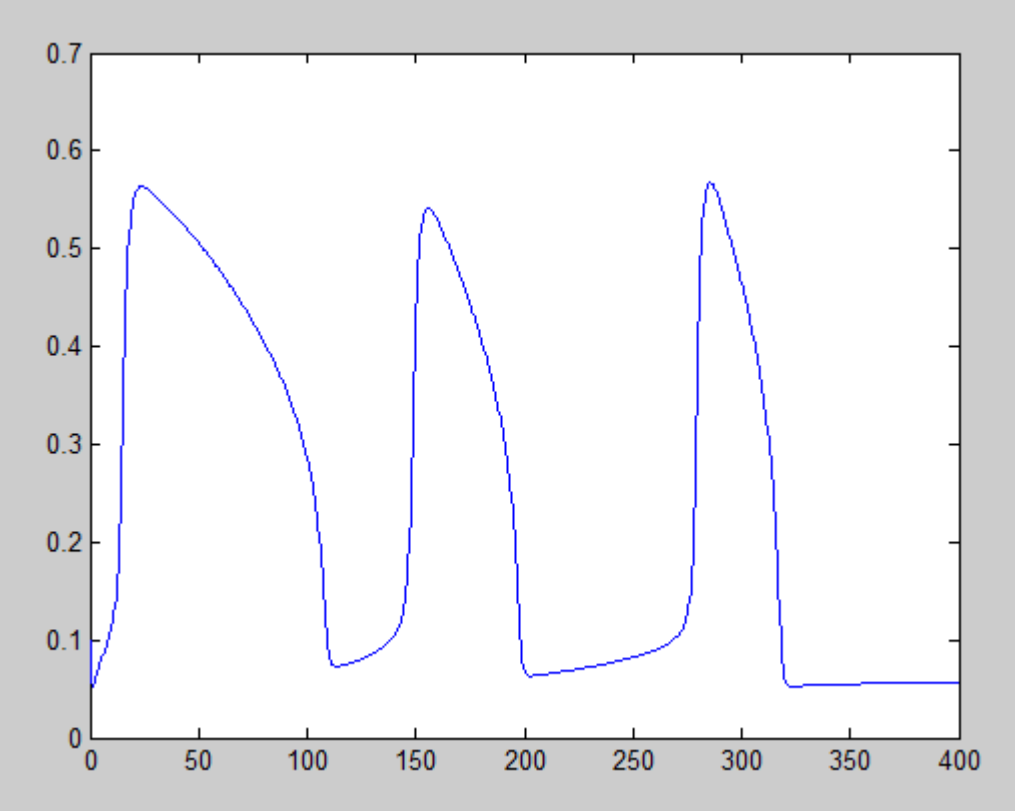


Figure 7.16 Simulation result of the composite model

Figure 7.16 shows the simulation result of the composite model that combines hormone-induced G-protein activation model and Hofer's minimum calcium oscillation model.

7.5 Model Configuration

When building biological models, it is often required to configure the quantitative information of the models. This quantitative information not only includes the mathematical equations that the models use to underlie the mechanisms they describe, but it also includes the initial states of variables and parameters these equations apply. The value of variables and parameters come from various sources including existing literature, experimental databases, and the quantitative analysis of experimental data. In order to allow modelers to manipulate all kinds of quantitative information, it is necessary to provide them facility that is able to accessible distributed data sources and mathematical analysis methods.

As demonstrated previously, our framework is able to transform biological data sources and data analysis methods into Semantic Web Services and allows the composition of these services interactively. The composition of data source services

and data analysis services gives the possibility of semi-automatic parameterization. As shown in figure 7.17, after selecting the services, the service composition becomes a new service with which we can then calculate the output of the parameters. The profile of this service can be published online for further reuse. By retrieving this profile, other researchers will be able to know what experimental database and what mathematical methods have been used to generate the parameters of interest. They may then reconfigure the profile on their own.

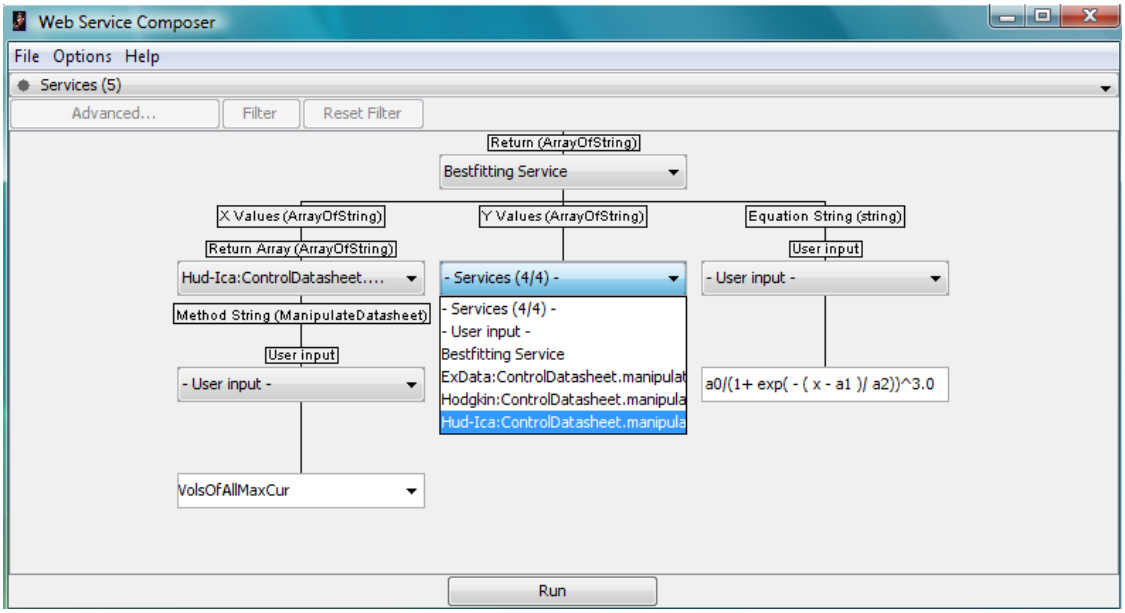


Figure 7.17 Interface of semi-automatic parameterization

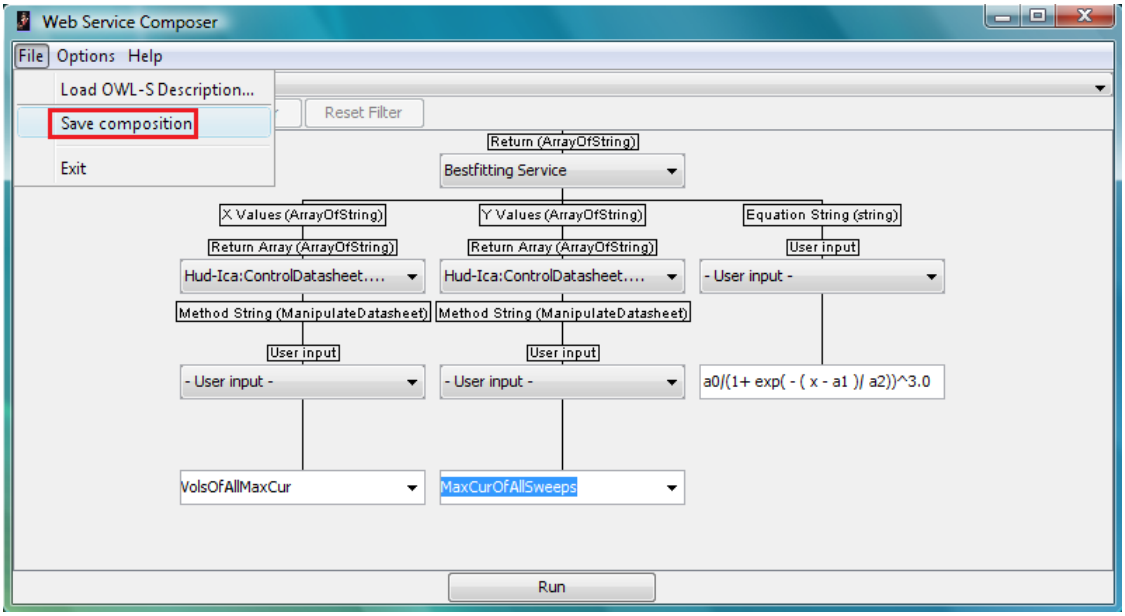


Figure 7.18 Save the composition profile to OWL-S files

We have used the Lewis and Hudspeth case to evaluate the model composition framework. We have successfully transformed the experimental data on which Lewis and Hudspeth model has built into a Web Service and published the semantic description of such service online. We then used the Web Service composer GUI retrieved this semantic description, which then gives us access to the original data store locally. After giving appropriate best-fitting function to the data, the parameters are calculated. The parameters are then stored in our OWL-based biological model. The OWL-S composition profile is published online and its URI is then stored in the OWL-based biological model.

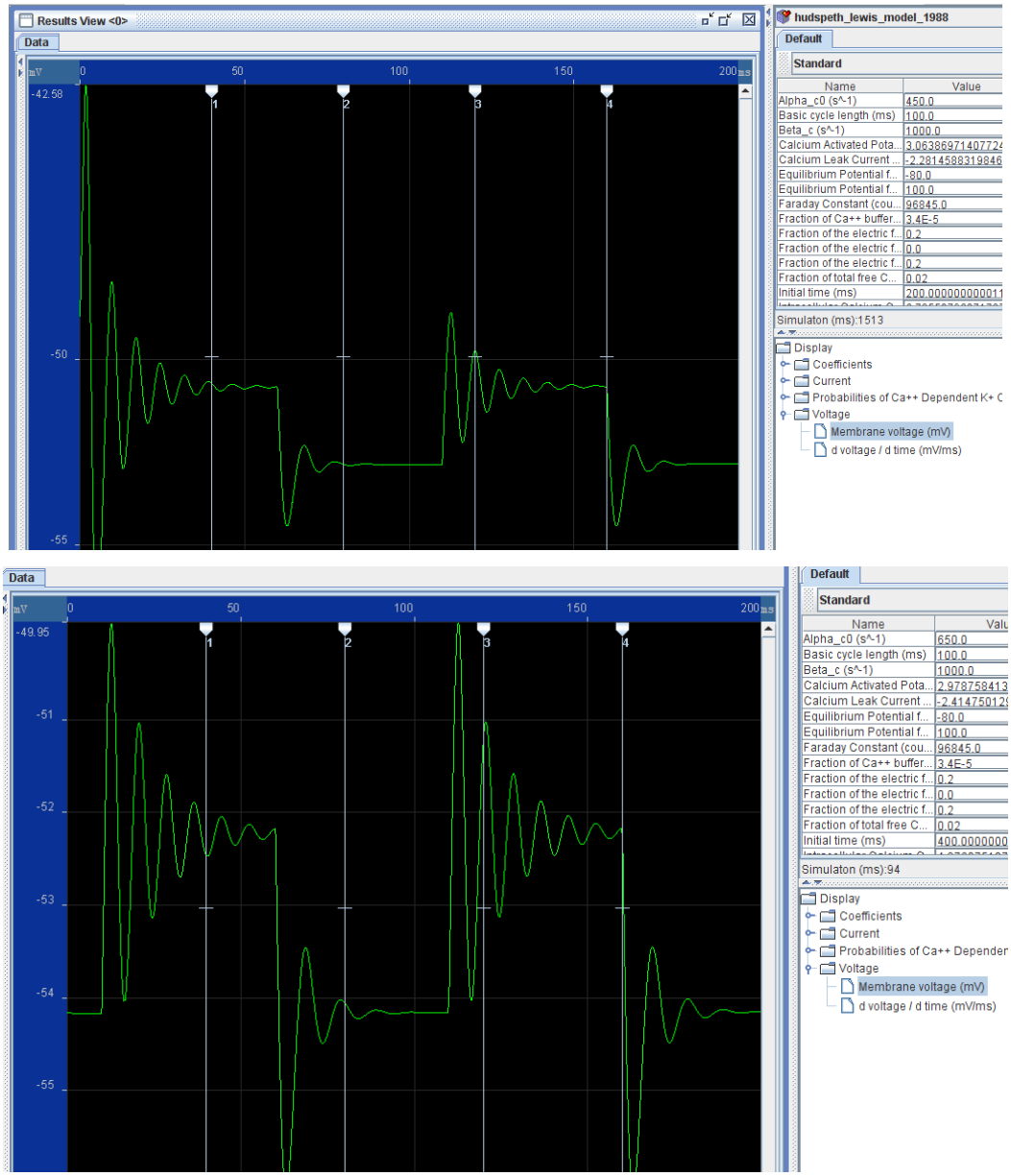


Figure 7.19 Comparing the results of automatically generated simulations by modify the parameters in OWL-based biological model

7.6 Summary

In this chapter we have evaluated the proposed framework with real cases of biological modeling. We have demonstrated that our framework can be of help in the aspects of model discovery, model reuse, model composition, automatic generating simulations from biological models, and allow interactive configuration of parameterization that review the effect of parameter modification directly in simulation results.

Chapter 8: Modeling Prestin – Framework Revisited

In this chapter I carry out a new, unseen, modeling exercise, in order to fully evaluate the benefits and limitations of the proposed knowledge standardization framework. This case study involves a currently ongoing experiment-modeling collaboration from the laboratory of my supervisor Prof Jonathan Ashmore in UCL Ear Institute.

The selected case is an effort of investigating the mechanism of electromotility in outer hair cells (OHCs) of the mammalian cochlea. More specifically, the case is concerned with the study of prestin, a protein densely packed in the OHCs, which is assumed to underlie the electromotility mechanism. This case shares the same system of interest, which is the hearing system, with the second case study introduced in Chapter 4. Therefore, the general background introduction will not be needed. There are however many differences between the new case and the previous one.

First and foremost, the selected case is an ongoing research project that is open to new assumptions and alternative theories. Most interestingly, contradictory experimental observations have been proposed by different research groups on some aspects of the study. Second, this new case investigates rather different phenomenon from the case in Chapter 4. The case in Chapter 4 is the study of electrical resonances in non-mammalian hair cells contributing to cochlear frequency tuning, while the new case is focused on the study of electromotility of mammalian outer hair cells contributing to cochlear amplification. Third, in the former case, modeling involves in connecting several sub-level models to a composite model; in the new case, modeling involves in evolving relatively simple kinetic models to similar but more complicated ones and adding constraints upon them. Four, different mathematical analysis approaches have been used for the two cases. More specifically, the former case uses the deterministic models while the new case uses stochastic models. We believe that the new case can fulfil the requirements of the suggested revision and is sufficient to validate the proposed framework in full.

In the following sections, first we will give the introduction of the biological background of cochlear outer hair cells electromotility. We will then describe the current discussion of the role of prestin in such mechanism. After that, we will give

the modeling approaches and mathematical analysis methods for the study. We will then apply our proposed framework to standardize related data and modeling knowledge, in order to examine the usefulness the framework. At the end, we will discuss the benefits and limitations of the framework and give closing remarks of the approach.

8.1 Biological Background

As we have introduced in Chapter 4 section 4.2.1, hearing depends on the conversion of sound stimuli into electrical signals, which are then sent to the brain for interpretation. The cochlea acts as the sensory organ for the hearing system. Its core component is the organ of Corti, which is distributed along the partition separating fluid chambers in the tapered tube of the cochlea. The cochlea contains sensory cells called hair cells. Hair cells are neuroepithelial cells positioned in the organ of Corti near the centre of the basilar membrane. Hair cells have the apical poles specialized for mechanotransduction and the basal poles specialized for the release of neurotransmitter.

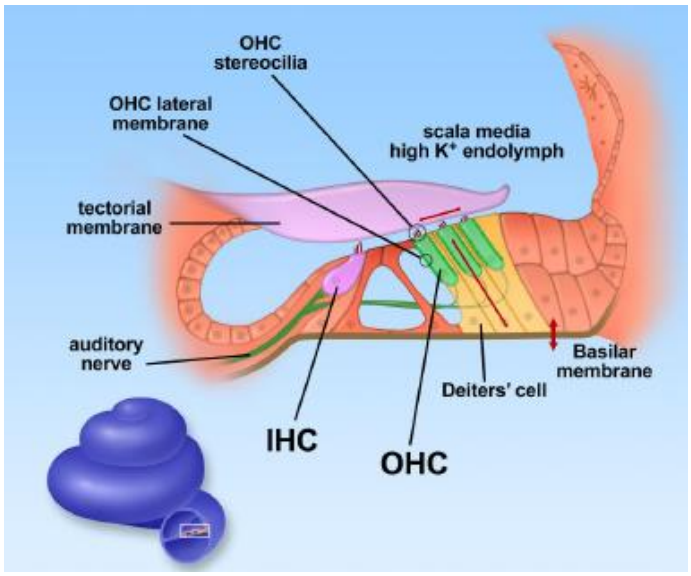


Figure 8.1 Organization of Organ of Corti and the hair cells on the basilar membrane in mammalian cochlea

Figure 8.1 is from the work by Jonathan Ashmore (Ashmore, 2008)

The mammalian cochlea contains two classes of hair cells - inner hair cells (IHCs) and outer hair cells (OHCs). While the IHCs are considered to be primarily

responsible for conveying electrical signals to the brain, OHCs are believed to play an important role in enhancing the sensitivity and frequency selectivity of mammalian hearing. Experimental measurements have shown that living ear is much more sharply tuned than that observed in dead ear, suggesting that there is an additional active process which feeds back energy to boost ear's response. OHCs are believed to be responsible for this active amplification process in hearing.

There are many different hypotheses as to how this process is achieved, but for mammals the most commonly accepted theory is that OHCs have a unique property known as electromotility. Electromotility is a mechanism in which OHCs are able to change length in response to electrical signals. The original observation was reported by Brownell et al. (Brownell et al., 1985) that isolated OHCs change length when electrically stimulated. Hyperpolarisation led to an elongation of the cell, and depolarisation led to a contraction of the cell. This was referred to as 'motility', with which hair cells actively provide feedback to cancel the dissipative forces in hearing system.

The voltage-dependent process of electromotility requires a charged voltage-sensor, just as voltage-dependent ion channels have voltage sensors. The mechanical response of OHCs is shown to be voltage-dependent and nonlinear, with the length changes displaying a sigmoidal dependence on the applied voltage. All cell membranes have a capacitance, which arises from the dielectric properties of the lipid bilayer. The intrinsic membrane capacitance is proportional to the total area of membrane, and is thus described as the linear membrane capacitance. However, a change in the total membrane capacitance arises as a result of the redistribution of the charge within the membrane, which gives the cell an additional voltage-dependent nonlinear capacitance on top of its intrinsic linear membrane capacitance. The correlation between the mechanical response of OHCs and their nonlinear membrane capacitance (NLC) that arose from the charge movement within the membrane suggested that electromotility was directly generated by a membrane-bound protein in the basolateral membrane.

The search for the molecular basis of electromotility was achieved in 2000, when a protein is identified to be highly expressed in cochlear tissue. The protein was named 'prestin' as it was able to confer on cells the ability to move *presto* (fast in Italian).

Electromotility is assumed to be driven by prestin. It is suggested that when a voltage-dependent conformation change occurs, the small change in protein area is translated into a change in cell length due to its high density in the membrane. The prestin protein has been successfully expressed in heterologous cells, which exhibited both a nonlinear capacitance and electromotility (Zheng et al., 2000).

Prestin is a member of superfamily SLC26 of integral membrane proteins and has been identified as the fifth member, SLC26A5. The SLC26 family is described as a family of anion-bicarbonate transporters, characterized by a sulphate transport motif in the amino acid sequence. Transporters are membrane transport proteins that assist the movement of substances across cell membranes by facilitated diffusion or active transport. All members of SLC26 family other than prestin have been shown to function as anion exchangers, with varying substrate specificities for a wide range of anions, including chloride (Cl^-), bicarbonate (HCO_3^-), sulphate (SO_4^{2-}), iodide (I^-), etc. Although prestin has not yet been shown to have any anion transport properties, it possesses all the sequence domains conserved throughout the SLC26 family.

At present there is controversy over the precise way in which prestin functions. When attempting to identify the voltage-sensor in prestin, Oliver et al. (Oliver et al., 2001) found that although several mutations of charged residues on the protein failed to abolish the NLC, removal of intracellular chloride (Cl_i) did abolish it. This led to the conclusion that rather than prestin containing an intrinsic voltage-sensor, intracellular chloride ions acted as an extrinsic voltage-sensor. The nonlinear membrane capacitance associated with prestin was found to depend upon intracellular chloride, Cl_i. However, there is no consensus about the origin of this dependence. This is partly resulted from the contradictory observations – that the NLC is abolished (Oliver et al., 2001) or is not abolished by complete removal of Cl_i (Rybalchenko and Santos-Sacchi, 2003).

Many attempts have been carried out to reproduce the experimental observation by quantitative modeling. No model has yet been presented to account for all the experimental observations. There is currently an ongoing effort in UCL Ear Institute that uses kinetic models to try to understand the function of prestin and how it underlies electromotility mechanism. In the following sections, the details of such modeling effort will be given (Muallem and Ashmore, 2006, Ashmore, 2008).

8.2 Modeling Prestin

8.2.1 Introduction

As introduced in the previous section, prestin is believed to underlie voltage-dependent length changes of OHCs, known as electromotility. This mechanical response is correlated with gating-charge movements, which is assumed to be coupled with a conformational change of prestin. This small change in protein area is then translated into a change in cell length due to the high density of prestin in the membrane ((Dallos et al., 1993, Iwasa, 1994).

Experimentally, charge movements are best measured by integrating the transient currents produced when the membrane potential is rapidly stepped. Measurements of gating charge movements under voltage clamp showed that the charge transferred across membrane, *Q*, had a sigmoidal dependent on membrane potential, *V* (see Figure 8.1). Therefore, the dependence of *Q* on *V* can be fitted with a Boltzmann function

$$Q(V) = \frac{Q_{max}}{1+e^{\beta(V-V_0)}} = Q_{max}\,\Phi_B(V) \qquad (8.1)$$

where Q_{max} is the maximum charge transferred across the membrane. V_0 is the voltage at which $Q=0.5Q_{max}$. The parameter β has an interpretation based on the statistical mechanics of a charge distributed across the electric field of the membrane and is given by $\beta = \frac{ze_0\delta}{k_B T}$, where *z* is number of elementary charges, δ is the fraction of the membrane field crossed, e_0 is the elementary charge, k_B is Boltzmann's constant, and *T* is the absolute temperature. In OHCs, β is found to be ~0.033mV^{-1} (Santos-Sacchi, 1991, Gale and Ashmore, 1997), implying that one elementary charge crosses ~80% of the membrane electric field. β depends on both the magnitude of the gating charge and the distance moved by it. It reflects the slope of the charge-voltage (*Q*-*V*) curve.

During normal whole cell recording conditions, whether charge or membrane capacitance is measured is a matter of taste, since the capacitance-voltage (*C*-*V*) curve is the derivative of the charge-voltage (*Q*-*V*) curve. The capacitance is not a constant function of voltage, thus the *C*-*V* curve is often referred to as a nonlinear capacitance (NLC) on the hair cell literature (see Figure 8.2). The corresponding *C*-

V curve associated with the total charge *Q(V)* moved during a voltage step from a hyperpolarized potential is give by the derivate

$$\mathrm{NLC(V)} = \frac{\mathrm{dQ}}{\mathrm{dV}} = \mathrm{Q_{max}}\,\beta[1 - \Phi_B(\mathrm{V})]\Phi_B(\mathrm{V}) = \frac{\beta \mathrm{Q_{max}}\, e^{\beta(\mathrm{V}-\mathrm{V_0})}}{(1+e^{\beta(\mathrm{V}-\mathrm{V_0})})^2} \qquad (8.2)$$

NLC is an additional voltage-dependent capacitance on top of the intrinsic linear membrane capacitance (C_{linear}) which can be established from the geometric cell area, therefore the cell capacitance is

$$\mathrm{C_m} = \mathrm{C_{linear}} + \mathrm{NLC(V)} \qquad (8.3)$$

The maximum nonlinear capacitance (C_{peak}) depends on the number of elementary charges moved and the maximum slop of the *Q-V* curve, i.e. C_{peak} is dependent on Q_{max}, and β, and occurs at $V=V_0$

$$\mathrm{C_{peak}} = \frac{\beta \mathrm{Q_{max}}}{4} \qquad (8.4)$$

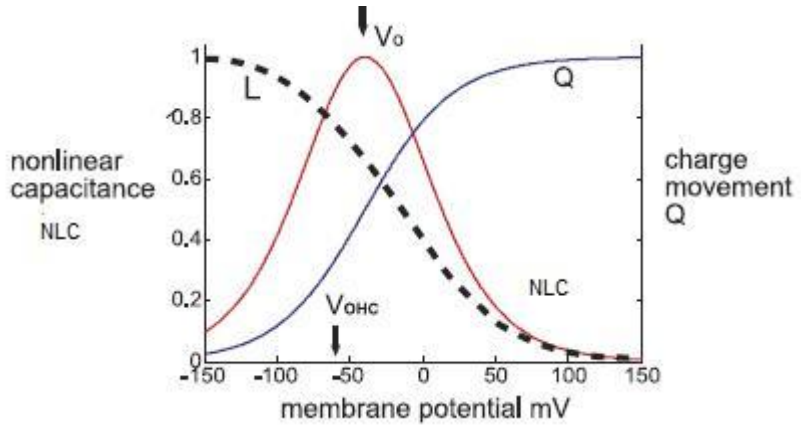


Figure 8.2 Dependence of nonlinear capacitance (NLC), charge movement (Q), and length change (L) on membrane potentials

In Figure 8.2 (Ashmore, 2008) the curves are all normalized and plotted according to *Equations 8.1* and *8.2* with $\beta=0.033\ mV^{-1}$ and $V_0=40\ mV$. V_{OHC} represents a typical OHC resting potential. The curves' position on the potential axis depends on experimental conditions.

The value of V_0 is variable. It can vary between -70 and -10mV even under normal experimental conditions. It also depends on many factors, which include the developmental stage of the hair cell, the phosphorylation state of the intracellular

proteins and on levels of intracellular calcium and chloride, the tension in the cell membrane, and so on.

In OHCs the NLC that arises from the redistribution of charge across the membrane has a peak value. This value is often in excess of $10pF$, and is therefore experimentally convenient to measure. The bell-shaped NLC is often considered to be the signature of OHCs and its properties are a primary target of studying OHCs electromotility.

The dependence of prestin associated NLC on intracellular chloride (Cl_i) has been established in previous studies (Oliver et al., 2001, Fakler and Oliver, 2002, Rybalchenko and Santos-Sacchi, 2003). The experimental observations of these studies are contradictory. In order to achieve a better understanding of how prestin works, models have been developed to reproduce the key experimental observations described above. In the following sections, we will introduce the modeling methods and models proposed by Muallem D. and Ashmore J. from UCL Ear Institute. The modeling work is still ongoing.

8.2.2 Methods

Simulation of Q and NLC

Most contemporary mechanistic descriptions of transporters employ an 'alternating-access' model, in which transporters undergo conformational changes, allowing transitions between states with either inward-facing or outward-facing binding sties. These models can be described using simple kinetic diagrams, consisting of the intermediate states of the transporter protein (E_i) and transitions connecting these states. Each transition is described by a dielectric coefficient, which reflects the dielectric distance over which charge is translocated during the transition.

For a reaction step $E_i \rightarrow E_{i+1}$, in which an electric charge q_i is translocated across a fraction of the membrane dielectric δ_i, the dielectric coefficient for the transition, α_i is given as

$$\alpha_i = \frac{q_i \delta_i}{e_0} = \gamma_i \delta_i \qquad (8.5)$$

where γ_i is the magnitude of elementary charge translocated. For a full cycle in which an ion of valency z is moved across the membrane, the sum of dielectric coefficient of all transitions in the cycle is given by

$$\sum_i \alpha_i = z \tag{8.6}$$

The reaction $E_i \leftrightarrow E_{i+1}$ can be described by rate constants f_i and b_i for the transitions in the forward and backward direction, with the equilibrium constant K_i given by

$$K_i = \frac{f_i}{b_i} \tag{8.7}$$

The application of the Boltzmann distribution, which describes the probability of a particle having a given energy, produces a relationship between K_i and the difference in free energy between the states, ΔG_i (Feynman 1965, Atkin 1986)

$$K_i = \overline{K_i} e^{\frac{\Delta G_i}{k_B T}} \tag{8.8}$$

where $\overline{K_i}$ is the value of K_i when $\Delta G_i = 0$. Since the transition involves the translocation of charge, there is an electrostatic contribution to ΔG_i, equal to the charge times the fraction of the transmembrane voltage V, it moves across. This gives

$$\Delta G_i = q_i \delta_i V = \alpha_i e_0 V \tag{8.9}$$

Therefore the rate constants for the transition are

$$K_i = \overline{K_i} e^{\frac{\alpha_i e_0 V}{k_B T}} = \overline{K_i} e^{\beta_i V} \tag{8.10}$$

where parameter β_i is defined similar in Equation 8.1.

If all voltage-dependent reaction steps are treated as transitions over a symmetric energy barrier, it gives

$$f_i(V) = \overline{f_i}(0) e^{\frac{\alpha_i e_0 V}{2 k_B T}} = \overline{f_i}(0) e^{\frac{\beta_i V}{2}} \tag{8.11}$$

$$b_i(V) = \overline{b_i}(0) e^{-\frac{\alpha_i e_0 V}{2 k_B T}} = \overline{b_i}(0) e^{-\frac{\beta_i V}{2}} \tag{8.12}$$

The rate constants of voltage-dependent reaction steps are therefore functions of dielectric coefficients and membrane potential V.

After a voltage perturbation, transitions through these voltage-dependent reaction steps result in a measurable transient current *I(t)*. Transient currents are simulated for a staircase voltage-ramp consisting of 1mV steps between -200mV and +200mV. The total charge moved across the membrane, *Q*, for each step was found by taking the area under each transient. *Q* is given by

$$Q = Ne_0\alpha_i \int [f_i(V)P_i(t) - b_i(V)P_{i+1}(t)]dt \quad (8.13)$$

where $P_i(t)$ is the occupancy probability of the state E_i. Following a voltage step, the occupancies of all states in a kinetic model will change, until a new steady-state is reached. The occupancy probabilities of each step can be found using the Q-matrix method by Colquhoun and Hawkes 1995 (Colquhoun and Hawkes, 1995). The NLC can then be calculated by differentiating *Q(V)*.

Q-Matrix Method

In the following we will introduce the Q-matrix method briefly.

$$E_1 \underset{k_{21}}{\overset{k_{12}}{\rightleftarrows}} E_2$$

Figure 8.3 A two-state kinetic model

Assuming a two-state model as in Figure 8.3, if we assign k_{ij} as the transition from state *i* to state *j*, and P_{ij} as the probability of the transition from state *i* to state *j* occurs. We have

$$\frac{dP_{11}}{dt} = -k_{12}P_{11} + k_{21}P_{12} \quad (8.14)$$

$$\frac{dP_{12}}{dt} = k_{12}P_{11} - k_{21}P_{12} \quad (8.15)$$

$$\frac{dP_{22}}{dt} = k_{12}P_{21} - k_{21}P_{22} \quad (8.16)$$

$$\frac{dP_{21}}{dt} = -k_{12}P_{21} + k_{21}P_{22} \quad (8.17)$$

This can be written in matrix notation as

$$\frac{d}{dt}\begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}\begin{pmatrix} -k_{12} & k_{12} \\ k_{21} & -k_{21} \end{pmatrix} \tag{8.18}$$

This matrix equation can be written in a compact form

$$\frac{d}{dt}P(t) = P(t)Q \tag{8.19}$$

The matrix Q on the right side containing rate constants q_{ij}, is the Q-matrix. The Q-matrix is written by placing the transition rate k_{ij} in the *i*th row and *j*th column, and then fill in diagonal elements (where $i=j$) with a number such that the sum of each row in the matrix is zero. This is entirely general regardless of the complexity of the particular kinetic scheme. The index variables i and j assume values from 1 to N, where N is the number of states in the scheme. Therefore an N-state scheme always produces an $N \times N$ matrix.

Solving Equation 8.19 gives us

$$P(t) = P(0)e^{Qt} \tag{8.20}$$

where $P(0)$ is the steady-state occupancies of each state before the perturbation is made.

The exponential of a matrix can be directly evaluated by numerical methods or in Matlab by using the 'expm' command. This gives all the values of $P_{ij}(t)$, and $P_i(t)$, the time course of occupancy probability of state E_i, is given by

$$P_i(t) = \sum_{j=1}^{N} P_{ji}(t) \tag{8.21}$$

In order to simulate transient currents, and consequently charge movements and NLCs, the Q-matrix for the initial steady-state is determined. First, whole-cell voltage clamp recordings are carried out, and the NLC was measured directly using a voltage-ramp. Acquired NLCs were fitted with the derivate of a Boltzmann function to give values for V_0, Q_{max}, β and C_{peak}. Rate constants are then optimized to minimize the squared deviation between the experimental data and the model. Rate constants are selected subject to two constraints:

1. In the domain charge moving step, voltage-dependent rate constants are chosen to be at least $7\times10^4 s^{-1}$, since OHCs electromotility can occur up to 70 kHz.
2. Unbinding rates are assumed to be fast and chosen to be 10^5 s^{-1}, similar to dissociation rates for fast binding and release from the aqueous phase.

Q-matrix is then calculated for each step in a staircase voltage-ramp, corresponding to the new voltage command. In these calculations N is the total number of prestin molecules. N was assumed to be 10,000 for an OHC patch, and 10^7 for a whole OHC. This gives NLC simulations, which can be compared with the NLCs acquired from experiments.

8.2.3 Mathematical Models

The dependence of prestin associated NLC on intracellular chloride (Cl_i) has been established in previous studies. Oliver et al. (Oliver et al., 2001) found that when Cl_i was totally removed and replaced with sulphate, the NLC was completely abolished, while Rybalchenko and Santos-Sacchi (Rybalchenko and Santos-Sacchi, 2003) showed that the magnitude of the NLC was almost unaffected when Cl_i was completely replace with sulphate. However, four key observations are consensus:

i) β of the NLC is ~$0.033 mV^{-1}$.
ii) The peak NLC (C_{peak}) decreases as [Cl_i] is reduced.
iii) The voltage at C_{peak} (V_0) shifts to more positive membrane potentials as [Cl_i] is reduced.
iv) The NLC is unaffected by removal of extracellular chloride (Cl_e) when [Cl_i]=150mM.

These observations suggested that intracellular chloride ions act as the voltage-sensor, such that a chloride ion is pushed across the membrane when membrane potential V is hyperpolarized. This thereby underlies the voltage-dependent charge movement, which can be detected as a transient current in response to a voltage-ramp or as a NLC.

Mathematical models have therefore been constructed to reproduce the key experimental observations described above. In the following, I will introduce the models proposed by Muallem D. and Ashmore J (Muallem and Ashmore, 2006).

In the described models, the following notations will be used:

E_i: This notation is used to label each different state in kinetic models. It is also assigned to represent the fraction of molecules in that state.

$E_i.A$: The point indicates that an anion A, is bound to the state E_i. For example if a chloride ion is bound to the state E_2, this bound state would be labelled as '$E_2.Cl$'.

Model 1: An Access Channel Model

It has been suggested that prestin works as an incomplete transporter using intracellular chloride ions as a voltage sensor without releasing chloride on the extracellular side (Oliver et al., 2001). The simplest way to describe this scheme is by a two-state 'access channel' model. The prestin protein is considered to have a half-pore facing the intracellular medium, so that an ion moving into the pore will experience part of the voltage drop across the membrane V. This leads to the effective affinity of chloride at the Cl^- binding site depending on V, and will cause a change in V to have kinetically the same effect as a change in $[Cl_i]$. This model can be expressed as a reaction scheme as in Figure 7.3, where E_0 and $E_1.Cl$ are the proportion of prestin in the unbound and bound state respectively.

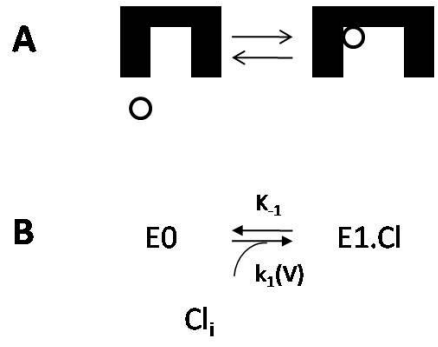


Figure 8.4 An access channel model

Figure 8.4 Illustration of an access channel model. A) Prestin changes from a contracted state to an expanded state when a chloride ion moves to a binding site inside the pore. B) The reaction scheme used to describe this scheme. Prestin changes from its initial unbound state E_0 to its bound state $E_1.Cl$. The forward rate k_1 depends on V and $[Cl_i]$. The backward rate k_{-1} is independent from V and $[Cl_i]$, and reflects the rate of dissociation of Cl_i from the chloride

binding site. The diagram and scheme are adapted from the work of Muallem and Ashmore (Muallem and Ashmore, 2006).

The reaction scheme leads to the differential equations

$$\frac{dE_0}{dt} = k_{-1}E_1.Cl - k_1E_0 \quad (8.22)$$

$$\frac{dE_1.Cl}{dt} = k_1E_0 - k_{-1}E_1.Cl \quad (8.23)$$

If chloride ions can reach the binding site by diffusion through the pore, then the effective concentration of chloride at the binding site $[Cl_i]_{bs}$ depends on both the concentration of chloride in the intracellular medium $[Cl_i]$ and the electric potential different between the intracellular medium and the binding site. The equilibrium distribution of chloride between the internal medium and the binding site $[Cl_i]_{bs}$ can be described by a Boltzmann expression similar to Equation 8.10

$$[Cl_i]_{bs} = [Cl_i]e^{\beta V} \quad (8.24)$$

where parameter $\beta = \frac{ze_0\delta}{k_BT} = \frac{\alpha e_0}{k_BT}$ has been introduced in previous sections. This makes the rate constant for the forward reaction voltage-dependent.

$$k_1(V) = k_1(0)[Cl_i]e^{\beta V} \quad (8.25)$$

Since a chloride ion has a valence, $z=-1$, β is negative and the forward rate of chloride binding is increased by hyperpolarisation. β is found to be ~$0.033mV^{-1}$, which corresponds to the dielectric coefficient α=0.8, i.e. a chloride ion moving across ~0.8 of the membrane. The dissociation rate k_{-1} is assumed to be fast and chosen as 10^5s^{-1}.

In this model, the charge moved across the membrane $Q(V)$ in response to a perturbation in either V or $[Cl_i]$ is proportional to the increase in the amount of prestin in the state $E_1.Cl$. Therefore $Q(V)$ is proportional to the steady-state proportion of E1.Cl at a given V. This gives

$$E_1.Cl = \frac{k_1(V)}{k_1(V)+k_{-1}} \quad (8.26)$$

As $k_1(V)$ increases such that $k_1(V) >> k_{-1}$, $E_1.Cl$ reaches 1. Therefore $Qmax$ corresponds to the condition that all the prestin molecules are in the state $E_1.Cl$.

Assuming that more chloride ions than prestin molecule are present even at very low [Cl_i], then for any [Cl_i], $k_1(V)$ can be made large enough to force all the prestin into the state $E_1.Cl$. Therefore no decrease in C_{peak} or Q_{max} can be observed. V_0 occurs when half of the maximal charge has been transferred, corresponding to the case when half of the prestin molecules are in state $E_1.Cl$, and half in state E_0. By using Equation 8.23, we have $\mathrm{E_1.Cl} = \frac{k_1(V)}{k_1(V)+k_{-1}} = \frac{1}{2}$. Combining it with Equation 8.25 gives

$$V_0 = \frac{1}{\beta} \ln \frac{k_{-1}}{k_1(0)[Cl_i]} \tag{8.27}$$

Thus the shift in V_0 due to a change in [Cl_i] from a first concentration to a second concentration is given by

$$\Delta V_0 = \frac{1}{\beta} \ln \frac{[Cl_i]_1}{[Cl_i]_2} \tag{8.28}$$

This means that the two-state access channel model produces a NLC, which depends on [Cl_i] as the V_0 of the NLC depends on [Cl_i]. However, unlike observed in experiments there is no change in C_{peak} or Q_{max} when [Cl_i] is reduced, and the direction of the shift in V_0 at reduced [Cl_i] is opposite to observation.

Model 2: A Three-state Model

The two-state access channel model can be extended to a three-state model. The chloride ion binds first at the mouth of the pore and is then translocated across the membrane to a second site at the top of the pore (see Figure 7.4A). Since the transition between the bound states of prestin is associated with charge translocation, this step must be voltage-dependent. Consequently this step is treated as a transition over a symmetric activation barrier, so that it is voltage-dependent in both directions. This three-state model can be described with a reaction scheme (see Figure 8.5B), where E_0, $E_1.Cl$ and $E_2.Cl$ are the proportion of prestin in that state respectively.

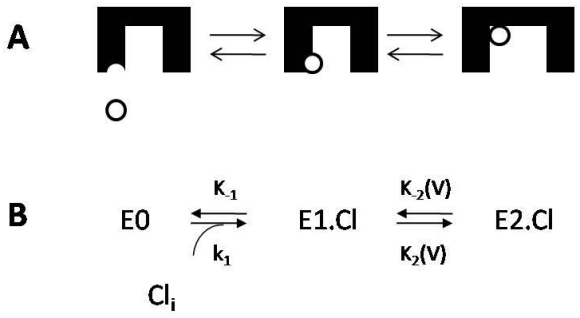


Figure 8.5 A three-state model

Figure 8.5 Illustration of a three-state model. A) Prestin changes from a contracted state to an expanded state when a chloride ion moves from the first binding site at the mouth of the pore to a second site at the top of the pore. B) The reaction scheme of the model. In the first step, the forward rate k_1 depends on $[Cl_i]$. The backward rate k_{-1} is a constant and simply reflects the rate of dissociation of Cl_i from the chloride binding site. In the second step, the chloride ion moves through the membrane to the second binding site. Both the forward and backward rate constants, k_2 and k_{-2} respectively, depend on V. The diagram and scheme are adapted from the work of Muallem and Ashmore (Muallem and Ashmore, 2006).

The reaction scheme leads to the differential equations

$$\frac{dE_0}{dt} = k_{-1}E_1.Cl - k_1[Cl_i]E_0 \tag{8.29}$$

$$\frac{dE_1.Cl}{dt} = k_1[Cl_i]E_0 - (k_{-1} + k_{-2})E_1.Cl + k_{-2}E_2.Cl \tag{8.30}$$

$$\frac{dE_2.Cl}{dt} = k_2E_1.Cl - k_{-2}E_2.Cl \tag{8.31}$$

The dissociation constant for the first step is similar to Equation 8.7, given by

$$K_D(Cl_i) = \frac{k_{-1}}{k_1} \tag{8.32}$$

Both the forward and backward rate constants of the second step are voltage-dependent. They are given similar to Equation 8.11 and 8.12

$$k_2(V) = k_2(0)e^{\frac{\beta V}{2}} \tag{8.33}$$

$$k_{-2}(V) = k_{-2}(0)e^{-\frac{\beta V}{2}} \tag{8.34}$$

Since the total amount of prestin is conserved

$$E_0 + E_1.Cl + E_2.Cl = 1 \tag{8.35}$$

In steady state Equation 8.29-8.31 equal to zero, combining with Equation 8.35 gives

$$E_2.Cl = \frac{1}{1+[\frac{k_{-2}}{k_2}(1+\frac{k_{-1}}{k_1[Cl_i]})]} \tag{8.36}$$

V_0 corresponds to the potential at which half the prestin molecules are in state E2.Cl, therefore

$$\frac{k_{-2}}{k_2}\left(1+\frac{k_{-1}}{k_1[Cl_i]}\right) = 1 \tag{8.37}$$

With Equation 8.33 and 8.34 we have

$$\frac{k_{-2}(0)}{k_2(0)}\left(1+\frac{k_{-1}}{k_1[Cl_i]}\right) = e^{\beta V_0} \tag{8.38}$$

This gives

$$V_0 = \frac{1}{\beta}\ln[\frac{k_{-2}(0)}{k_2(0)}\left(1+\frac{k_{-1}}{k_1[Cl_i]}\right)] \tag{8.39}$$

And

$$\Delta V_0 = \frac{1}{\beta}\ln\left[\frac{1+\frac{k_{-1}}{k_1[Cl_i]_2}}{1+\frac{k_{-1}}{k_1[Cl_i]_1}}\right] \tag{8.40}$$

The three-state model produces similar results to the two-state access channel model. There is no decrease in C_{peak} or Q_{max} when [Cl_i] is reduced and the shift in V_0 in response to reduced [Cl_i] is in the hyperpolarizing direction. The main different between the two models is that in the three-state model V_0 saturates at high [Cl_i].

Model 3: A Chloride Transporter Model

Both model 1 and model 2, which are nontransporting models, resulted in C_{peak} and Q_{max} not decreasing when [Cl_i] is reduced. This is in contradiction with experimental observations. Since prestin belong to a family of anion exchangers, a model in which prestin completes a full transport cycle have been developed (see Figure 8.6).

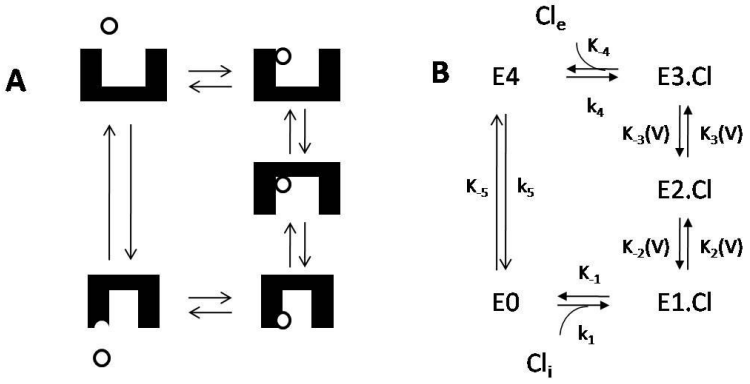


Figure 8.6 A five-state chloride transporting model

The diagram and scheme in Figure 8.6 are adapted from the work of Muallem and Ashmore (Muallem and Ashmore, 2006).

In this model a Cl^- ion binds to a binding site facing the intracellular medium, then moves through the membrane to a second site accompanied by a conformational change from where Cl^- moves to a third site facing the extracellular medium and is released. Prestin then returns to the inward facing state with no ion bound.

For simplicity it is assumed that the chloride ion crosses the whole membrane dielectric in two steps, $E_1.Cl \leftrightarrow E_2.Cl$ and $E_2.Cl \leftrightarrow E_3.Cl$. The binding and unbinding of both intracellular and extracellular chloride are assumed to be voltage-independent.

If the fraction of the membrane dielectric crossed in step $E_1.Cl \leftrightarrow E_2.Cl$ is δ_1 and the fraction of the dielectric crossed in $E_2.Cl \leftrightarrow E_3.Cl$ is δ_2, then $\delta_1 + \delta_2 = 1$, and the respective dielectric coefficients for a chloride ion with valence $z=-1$ are given by $\alpha_1 = z\,\delta_1$ and $\alpha_2 = z\,\delta_2$, so $\alpha_1 + \alpha_2 = -1$, and the rate constants for the two steps are

$$k_2(V) = k_2(0)e^{\frac{\beta_1 V}{2}} \tag{8.41}$$

$$k_{-2}(V) = k_{-2}(0)e^{-\frac{\beta_2 V}{2}} \tag{8.42}$$

$$k_3(V) = k_3(0)e^{\frac{\beta_2 V}{2}} \tag{8.43}$$

$$k_{-3}(V) = k_{-3}(0)e^{-\frac{\beta_2 V}{2}} \tag{8.44}$$

where $\beta_1 = \frac{\alpha_1 e_0}{k_B T}$ and $\beta_2 = \frac{\alpha_2 e_0}{k_B T}$.

In model simulation, two particular cases were considered. In the first case, all the charge translocation occurred across the first step (α_1=*-1*, α_2=*0*), resulting in one voltage-dependent step and NLCs with β*~0.04mV*$^{-1}$, in poor agreement with experimental data. In the second case, the Cl^- ion crosses 80% of the membrane dielectric in the first transition and 20% of the membrane dielectric in the second transition, resulting in two voltage-dependent steps (α_1=*-0.8*, α_2=*-0.2*), and NLCs with β*~0.03mV*$^{-1}$ in agreement with the experimental observations.

Rate constant	Range of values tested	Optimized rate constants
k_1	10^6-$10^8 M^{-1}s^{-1}$	$10^8 M^{-1}s^{-1}$
k_{-1}	$10^5 s^{-1}$	$10^5 s^{-1}$
$k_2(0)$	$7\times10^4 s^{-1}$-$42\times10^4 s^{-1}$	$7\times10^4 s^{-1}$
$k_{-2}(0)$	$7\times10^4 s^{-1}$-$42\times10^4 s^{-1}$	$28\times10^4 s^{-1}$
$k_3(0)$	10-$10^5 s^{-1}$	$10^4 s^{-1}$
$k_{-3}(0)$	10-$10^5 s^{-1}$	$10^5 s^{-1}$
k_4	$10^5 s^{-1}$	$10^5 s^{-1}$
k_{-4}	10^6-$10^8 M^{-1}s^{-1}$	$2.5\times10^6 M^{-1}s^{-1}$
k_5	0-$10^5 s^{-1}$	$10^5 s^{-1}$
k_{-5}	0-$10^5 s^{-1}$	$10^5 s^{-1}$

Table 8.1 Range of rate constants tested in simulations for the Cl^- transporter model

The parameters in Table 8.1 is from the work of Muallem and Ashmore (Muallem and Ashmore, 2006), and are used in their simulation. We also use these parameters in the simulations generated by our framework.

For a cyclic process microscopic reversibility must be obeyed. Thus there are nine independent rate constants for this model. Table 8.1 shows the range of rate constant tested. With rate constant optimized, the NLC for [Cl_i]=*150mM* gives V_0=*-40mV*, and β*=0.032mV*$^{-1}$. For all combinations tested, the model produces a decrease in C_{peak} when [Cl_i] is reduced. No combination of rate constants tested produces a positive shift in V_0. The chloride transporter model is able to reproduce many aspects of the dependence of the NLC on [Cl_i] and [Cl_e] observed experimentally.

Model 4: A Chloride/Sulphate Exchanger Model

The chloride transporter model shows that in a cyclic model where Cl^- is fully transported across the membrane, a decrease in C_{peak} is a consequence of reduced [Cl_i]. However, none of the previous models can produce a positive shift in V_0 as [Cli] is reduced. Therefore a chloride/sulphate exchanger scheme is proposed as in Figure 8.7.

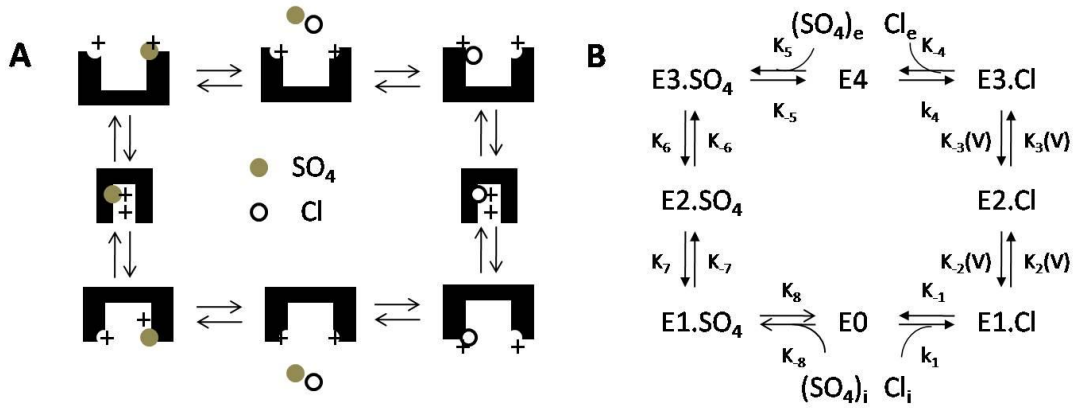


Figure 8.7 A Chloride/Sulphate exchanger model

The diagram and scheme in Figure 8.7 are adapted from the work of Muallem and Ashmore (Muallem and Ashmore, 2006).

Consider a Cl^- transporter model scheme as in Figure 8.6, but with the movement of Cl^- accompanied by the movement of intrinsic positively charged residues, so that there is a net movement of positive charge when Cl^- is transported from intracellular binding site toward the extracellular surface. During a complete cycle, the intrinsic charged residues must be returned to their original position, requiring an additional voltage-dependent step. Since SO_4^{2-} has previously been used to substitute for Cl^- in experiments, this model is implemented explicitly as a Cl^-/SO_4^{2-} exchanger, exchanging one Cl^- ion for one SO_4^{2-} ion. Therefore the net charge translocated across the membrane from the inside to the outside of the membrane is equivalent to that of a positively charge ion with a valence of +1. If the dielectric coefficients for the transition $\mathrm{E_1.Cl} \leftrightarrow \mathrm{E_2.Cl}$ and $\mathrm{E_2.Cl} \leftrightarrow \mathrm{E_3.Cl}$ are given by α_1 and α_2 respectively, and the dielectric coefficient for the transition $\mathrm{E_3.SO_4} \leftrightarrow \mathrm{E_2.SO_4}$ and $\mathrm{E_2.SO_4} \leftrightarrow \mathrm{E_1.SO_4}$ are given by α_3 and α_4 then

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1 \qquad (8.45)$$

If the translocation of chloride is assumed to be voltage-dependent and that $\alpha_{1+}\ \alpha_2=-1$, this gives $\alpha_{3+}\ \alpha_4=2$. Therefore the translocation of sulphate across the membrane must also be voltage-dependent.

In simulations, two cases were considered. In the first case, the Cl^- ion and the equivalent of two positively charged residues crosses 100% of the membrane dielectric in the first transition ($\alpha_1=-1$, $\alpha_2=0$), resulting in NLCs with $\beta\sim0.04mV^{-1}$. In the second case, the Cl^- ion and the equivalent of two positively charged residues crosses 80% of the membrane dielectric in the first transition and 20% of the membrane dielectric in the second transition, resulting in two voltage-dependent steps ($\alpha_1=-0.8$, $\alpha_2=-0.2$), resulting in NLCs with $\beta\sim0.03mV^{-1}$. Because this mechanism obeys microscopic reversibility, there are 15 independent rate constants for this model. The Cl^-/SO_4^{2-} exchanger model produces a decrease in C_{peak} on reducing $[Cl_i]$ for all combinations of rate constants.

8.3 Framework Evaluation

An ongoing biological modeling case has been introduced as above. In this chapter, we will evaluate the proposed ontology-centered framework with this new case. We intend to demonstrate that by using this framework, we are able to facilitate many aspects of biological modeling in practice, including automatic model discovery, semi-automatic reconfiguration of model parameterization, model reuse and composition, and automatic transformation to simulations.

8.3.1 Using the Proposed Framework

First, we used the OWL-based meta-model for building biological models, which is used as a template for model creation in systems biology, to construct models for all four models described in Section 8.2.3. These models use external shared ontologies for biological term annotation.

Then, we transformed experimental data into semantic web services. Data providing web services have connected with the semantic Web Service performing best-fitting method to fit Boltzmann function to give values for V_0, Q_{max}, β and C_{peak}. These values are then stored in the OWL models for further use in generating simulations.

We also developed transformation engine that can read the OWL-based biological models and extract the kinetic schemes to generate Q-matrix. The generated Q-matrices can be passed to a standalone simulation engine that use Q-matrix method to calculate the transition rates of each voltage-step. The NLC simulations are easily generated.

All the newly built models and programs have been uploaded to https://sourceforge.net/projects/biologicalmodel/, together with the programs of other cases.

8.3.2 Model Discovery

By using the proposed meta-model to build OWL-based biological models for all the prestin models described in previous sections, the models and their components are able to be annotated with ontological terms from shared biological ontologies. The annotation process is the same as described in Chapter 7.1.

In our meta-model, we have defined the class *model* has properties called *listOfEntity* and *listOfRelationship*, whose values are instances of terms in Gene Ontology (GO). In the prestin case, the biological models mainly describe the properties and activities of prestin proteins, which are anion transporters. Therefore, we use the entity defined in Gene Ontology, “transporter activity” (GO term ID GO:0005215) and “prestin” (Synonyms voltage-gated sulphate antiporter activity with GO term ID GO:0046609) to annotate the root of the models. We are then able to instantiate the entity for further use when building biological models.

These models can be easily discovered by GO term IDs in the interface we have developed (as described in Section 7.1). Various semantic queries can be used to test the result.

8.3.3 Model Parameterization

As demonstrated previously, our framework is able to transform biological data sources and data analysis methods into Semantic Web Services and allows the composition of these services interactively. The composition of data source services and data analysis services gives the possibility of semi-automatic parameterization.

As shown in figure 8.8, after selecting the services, the service composition becomes a new service with which we can then calculate the output of the parameters. The profile of this service can be published online for further reuse. By retrieving this profile, other researchers will be able to know what experimental database and what mathematical methods have been used to generate the parameters of interest. They may then reconfigure the profile on their own.

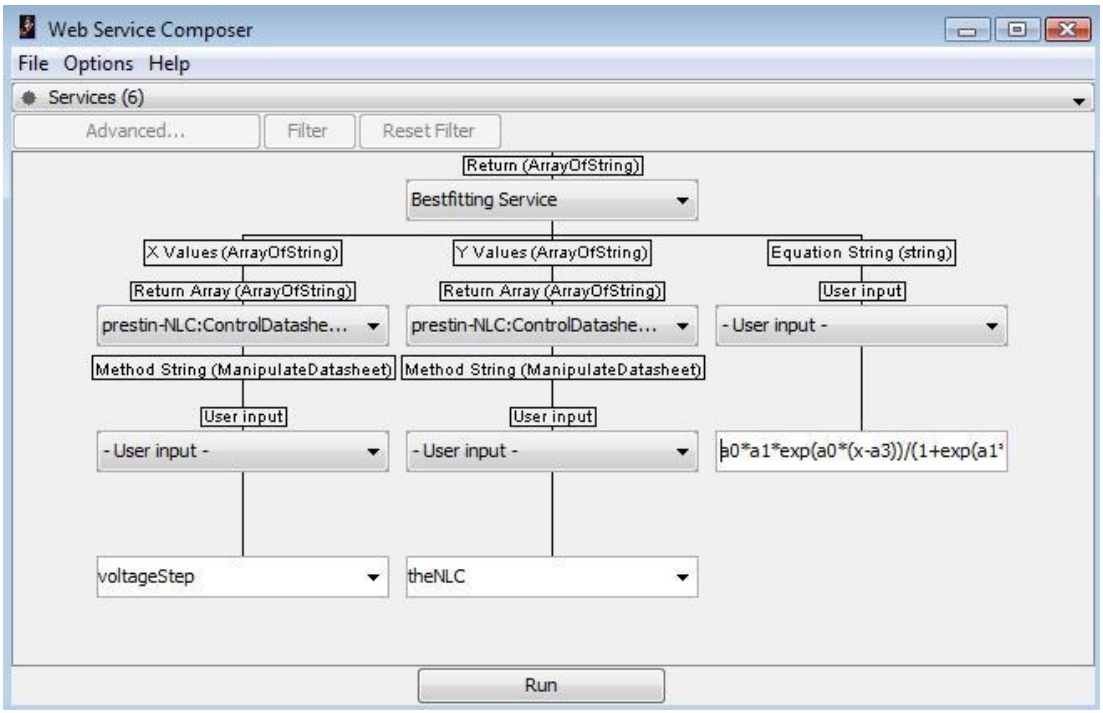


Figure 8.8 Parameter Configuration by Service Composition

In this case, we have transformed a set of experimental data acquired from whole-cell voltage clamp recording. The data represent the recorded nonlinear membrane capacitance C against the membrane potential V in a staircase voltage-ramp consisting of $1mV$ steps between -135mV to 65mV with 1mM $[Cl_i]$ and sulphate replacement. The data are transformed as Web Service in NetBeans and deployed online.

The data-providing service is then composed together with the best-fitting Web Service introduced in Chapter 6.3. Data are fitted with Boltzmann function $\mathrm{NLC(V)} = \frac{\beta Q_{\max}\, e^{\beta(V-V_0)}}{(1+e^{\beta(V-V_0)})^2}$ (as in Equation 8.2), to give values for V_0, Q_{max}, β and C_{peak}.

8.3.4 Model Reuse

The reusability of our OWL models is based on the fact that all of them share the same modeling template. Therefore, all the model components are easy to be copied while preserving the modeling knowledge in the existing models. Since we have defined our *model* class to have property of *modelReference*, the reference information of which models have contributed to the construction of the new model can be properly documented.

We have investigated the model reuse case by evolving simple kinetic models of prestin to more complicated ones by using the process introduced in Section 7.2. This practice demonstrates that with the model reuse ability provided by the proposed framework, biological models can be easily constructed and the reuse of existing knowledge can be traced back to previous modeling efforts.

8.3.5 Automatic Generation of Simulations

The fast generation of computational simulations can be of great help to shorten the life cycle of biological model development. XSLT transformation engines for the proposed framework can be developed to translate the biological models built by using the OWL meta-model into simulation directly.

In the case of prestin models, a transformation engine can be developed which reads the OWL-based biological models and extracts the kinetic schemes to generate Q-matrix. The generated Q- matrices can be then passed to a generic Matlab method that implements the Q-matrix method. The simulations are able to calculate the transition rates of each voltage-step and give the NLC simulations.

The automatic generation from OWL-based biological models to simulations has been tested in all the four prestin models introduced previous. We have implemented the generic Q-matrix method in Matlab, which can read the Q-matrices and produce NLC simulations. Due to the limitation of time, the program that demonstrates the extraction of Q-matrices from OWL-based biological models is omitted. It is similar to the XSLT engine we developed for previous cases, which can extract equations and values of parameters from the OWL files. Alternatively, we used the values of transition rates given by Muallem and Ashmore to form the Q-matrices. The Q-

matrices that are used by qmatrix.m as inputs reflect the rate constants and concentrations of ions such as intracellular chloride. The voltage dependency of the rate constants are represented in separate matrices and passed to qmatrix.m as a distinctive parameter (see run_prestin0-3.m in the program package for details).

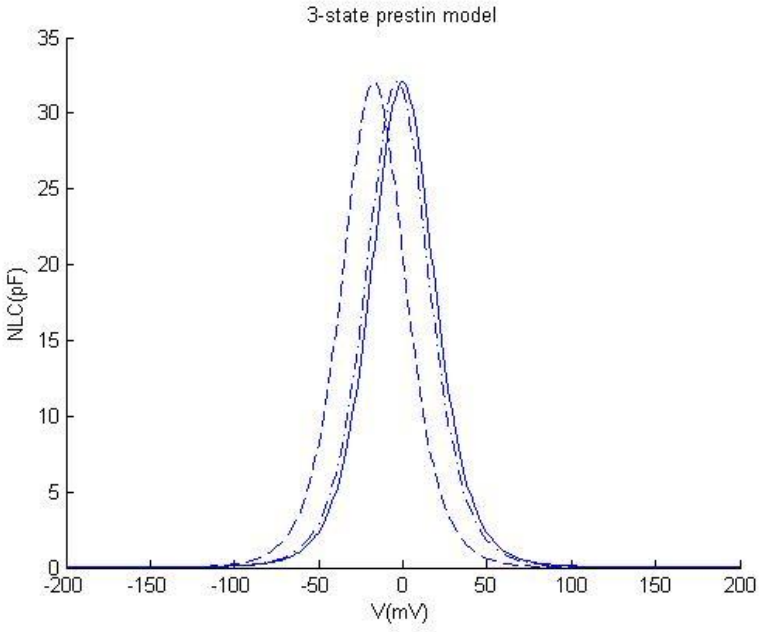


Figure 8.9A Simulation results of 3-state access channel model

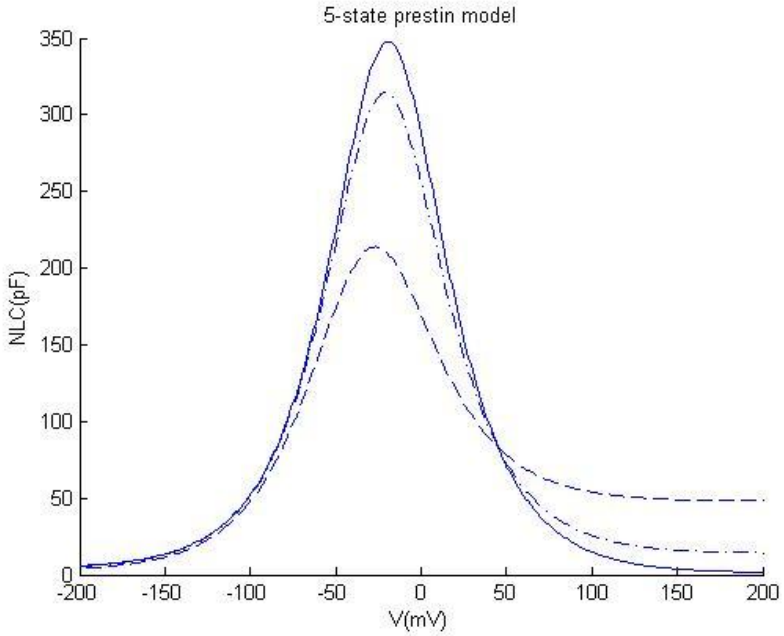


Figure 8.9B Simulation results of 5-state chloride/sulphate transporter model

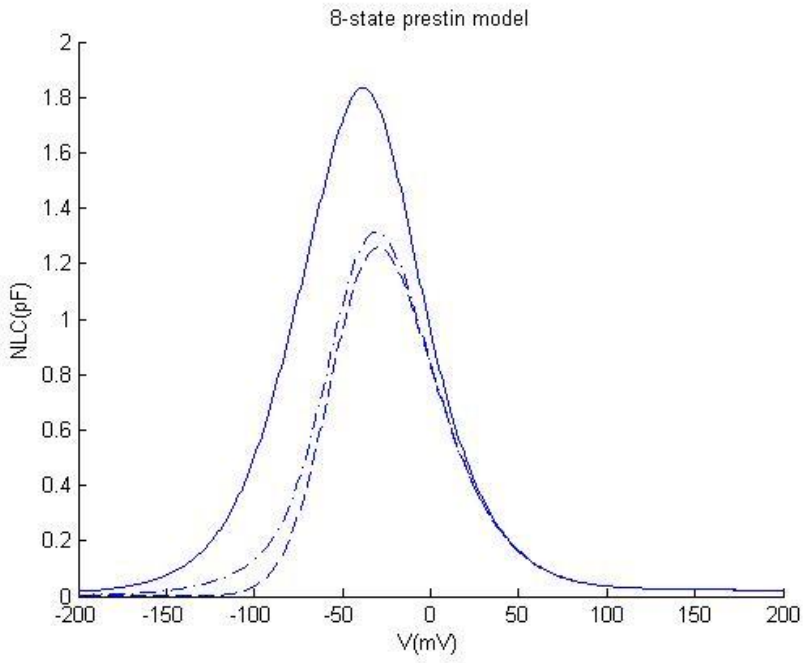


Figure 8.9C Simulation results of 8-state chloride/sulphate transporter model

Figure 8.9 Results of automatically generated simulations

In Figure 8.9 of all three graphs, the discontinuous curve (--) represents the NLC with [Cl_i]=*0mM*; the dotted curve (-.-) with [Cl_i]=*10mM*;and the continuous curve with [Cl_i]*=150mM.* All curves are produced under the condition of [Cl_e]=*150mM.*

Users can change the parameters in OWL-based models. The effect of changing parameters will then be displayed instantly from generated simulations. The simulations generated by our framework produce the same results as the hand-crafted simulations by Muallum and Ashmore. This approach is proved to be easy-to-use and generic. With the ability of generating simulations from biological models automatically, it is simple and efficient to test the validity of model components and parameterization.

8.4 Summary

In this chapter, we have introduced models for investigating the mechanism of electromotility in outer hair cells (OHCs) of the mammalian cochlea. We have evaluated the proposed framework following the proposed procedure. We are able to demonstrate that by using this framework, we are able to facilitate many aspects of biological modeling in practice, including automatic model discovery, semi-

automatic reconfiguration of model parameterization, model reuse and composition, and automatic transformation to simulations.

Chapter 9: Discussion and Future Work

In this thesis, we have given a practical approach to apply Semantic Web technologies to tackle the challenges in modeling in systems biology. We have proposed an ontology-centered framework that is capable of standardizing various modeling knowledge in systems biology and achieving collaboration among standardized knowledge sources.

We have evaluated the proposed approach by realistic case studies in biological modelling. We believe the evaluation result suggests that our approach is valid and plausible. The three cases that we have used are of biological significance and exemplary in the problem domain of biological modelling. They cover a reasonable range of models and are comprehensive enough to represent the major challenges faced in the day-to-day modelling effort. Admittedly, these cases are focused on physiological research, but we believe that our approach is not limited to help in this area only. In the future, we would like to test our framework with the cases of other types of biology. It is possible that the framework may be adjusted for including more knowledge components in such cases. Since our framework is underpinned by an ontology-based meta-model, the extensibility guaranteed by ontology models will be very advantageous to overcome many potential scalability issues.

Among existing efforts that aim at standardizing biological modeling knowledge and connecting knowledge components through web-based techniques, Taverna workbench from myGrid project shares many similarity with our work. The difference between myGird and our work is that myGrid is a large UK e-Science pilot project and it implements a middleware to enable a virtual workbench for data-intensive bioinformatics. The problem domain that Taverna deals with is data integration of genetic information. On the other hand, my work is focused on the modeling of physiological level. Also, despite the Taverna workbench claims using DAML+OIL ontologies to connect biological web services, the ontologies are seemingly inaccessible to users, and the semantic annotations are not displayed in the software implementation.

There is also effort of online registry for biological web services. For example, BioCatalogue is a system that collect, store, validate, and make available web-services in the biosciences. However, it is still in a preliminary stage and the usability is unknown. Another example of adapting semantic web vision in the life sciences is BioMoby project. The MOBY-S system defines an ontology-based messaging standard through which a client will be able to automatically discover and interact with task-appropriate biological data and analytical service providers. This project is connected with myGrid project, and as well is focused on tasks in bioinformatics. Therefore, these existing efforts are not yet suitable to address the physiological modeling problems as the framework we have proposed.

We have provided a solution that strikes a balance by being both practically motivated and implemented, and theoretically sound. We believe that our effort may have initiated a promising start of a new area of investigation, and we hope more effort can follow. Therefore, we would like to make a few suggestions for possible areas of future work.

First, we are keen to see fully-fledged software to be built based on the framework we have proposed. It could either be a standalone biological modelling workbench or a plug-in package of an ontology editor such as protégé. The basic components that we would suggest to be included in this software are: a graphical model construction environment to apply the meta-model, an interface for accessing shared biological ontologies for easy model annotation, an interface for composing external Web Services interactively and embedding the composition profiles directly back to the model to be built, and an simulation environment to have instant result of the model construction.

We would also like to expand this framework to include future development of semantic web techniques to support more sophisticated modeling tasks. First, reasoning engines that support description logic such as RacerPro (RacerSystems, 2004-2008) and Pellet (Clark&Parsia, 2004-2008) OWL reasoner could be applied to infer the relations between entities across different models. Having our proposed meta-model that supports model annotation by shared biological ontologies such as GO, we are provided a semantic registry service for biological 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 parameterization processes can be combined with case-based reasoning, which applies a retrieve-reuse-revise-retain approach (Lenz et al., 1998). By combining case-based reasoning to our proposed framework, some modeling tasks can be solved automatically by reusing existing cases. For example, the existing OWL-S models of parameterization can be discovered by using semantic reasoning (as discussed above). Since these models contain the information of the selection of biological database and the mathematical methods for calculating parameters, they can be reused as possible solutions for similar parameterization tasks. These possible solutions can then be revised by users as needed. With the capability of automatically generating simulations provided by our framework, the results of such revision can be monitored instantly. After the solution has been successfully adapted to the target problem, the new OWL-S models are then stored as possible solutions for future tasks.

Further, heuristic method can be combined with our framework. A heuristic is a technique designed to solve a problem that ignores whether the solution can be proven to be correct, but which usually produces a good solution to solve similar problems (Pearl and Judea, 1983). For example, when building the biological model for the Lewis & Hudspeth case discussed in chapter 4, we have reused the Hodgkin & Huxley model to describe the kinetics of ion channels. This can be done heuristically. Many models including the Hodgkin & Huxley model related to the modeling task can be automatically discovered and retrieved by semantic reasoning. They will then be composed together to generate likely models to describe the Lewis & Hudspeth case. The capability of automatically generating simulations by our framework will again be very helpful to validate the heuristically built models against experimental data in real time.

Chapter 10: Conclusion

This thesis addresses the problem of modeling in systems biology within the Semantic Web version. The aim of our work is to standardize various areas of knowledge including experimental data, quantitative biological models, and data analysis methods by taking advantage of the latest Semantic Web technologies, and enabling the collaboration among these standardized knowledge components.

In our study, we have identified the main challenges in biological modeling, and reviewed the current development and application of the Semantic Web in the life sciences. By examining the existing effort, we have discovered that the current application of the Semantic Web in biological modeling is still preliminary. We have found out that the main application for ontologies in the life sciences is currently focused on sharing biology terminologies. The Web Services used in the biological studies are limited and unorganized. We have argued that there are much more benefits to be had by using ontology along with Semantic Web Services to supporting modeling tasks in systems biology.

An ontology-centered Semantic Web infrastructure can provide a rich set of facilities for helping with formal knowledge representation, knowledge sharing, reuse, and collaboration. Instead of focusing on one of these angles, we have proposed a systematic approach, in order to maximize the advantage of adapting the Semantic Web vision.

After examining the main knowledge components of biological modeling by case studies, we have proposed our ontology-centered framework for biological modeling. In this framework, we have specified an ontology-based meta-model for building biological models, which exceeds the existing knowledge representation effort CellML and SBML. By using OWL as the format, our meta-model is able to support model annotation by shared biological ontologies, support semantic queries, automatic discoveries, easy model reuse and composition, and embedment of external knowledge such as profiles of parameterization processes. Having this meta-model as the foundation to connect standardized knowledge components, we have then proposed the procedure of transforming biological data sources and data analysis methods into Web Services. These Web Services can then be composed

together to perform parameterization in biological modeling. The decision-making and workflow of parameterization processes can then be stored in OWL-S models, and embedded in model instances built on our proposed meta-model.

By evaluating our ontology-centered framework with three case studies, we have demonstrated the improvement we have achieved in the aspects of model discovery, reuse, and composition. We have also demonstrated that by using this framework we can interactively configure the data sources and data analysis methods for the parameterization processes when building biological models, and then embed profiles of the rationale and workflow of such processes in biological models. Moreover, we have demonstrated that the ontology-based models can be directly converted to computational simulations so that the result of model construction can be seen instantly. This considerably shortens the life cycle of biological modeling.

Finally, we are able to conclude that by using ontology to represent biological models and using Semantic Web Services to standardize other knowledge components such as biological databases and data analysis methods, greater capability of knowledge sharing, reuse and collaboration can be achieved. We also conclude that ontology-based biological models with formal semantics are essential and necessary to standardize knowledge in compliance with the Semantic Web vision.

References

ALBERTS, B., BRAY, D., LEWIS, J., RAFF, M., ROBERTS, K. & WATSON, J. D. (1994) *Molecular Biology of the Cell, 4th Edition*, Garland Publishing.

ART, J. J. & FETTIPLACE, R. (1987) Variation of membrane properties in hair cells isolated from the turtle cochlea. *J Physiol,* 385**,** 207-42.

ASHMORE, J. (2008) Cochlear outer hair cell motility. *Physiol Rev,* 88**,** 173-210.

ASHMORE, J. & GALE, J. (2000) The Cochlea: A Primer. *Current Biology,* 10**,** 325-327.

BAIROCH, A. (2000) Serendipity in bioinformatics, the tribulations of a Swiss bioinformatician through exciting times! *Bioinformatics,* 16**,** 48-64.

BAKER, C. J. O. & CHEUNG, K.-H. (2007) *Semantic Web: Revolutionizing Knowledge Discovery in the Life Sciences*, Springer Science and Business Media, LLC.

BECKER, W. M., KLEINSMIT, L. J. & HARDIN, J. (2000) *The World of the Cell, Fourth Edition*, Addison Wesley Longman, Inc.

BERNERS-LEE, T. (2000) Semantic Web on XML: Slide 10 Architecture. .

BROWNELL, W. E., BADER, C. R., BERTRAND, D. & DE RIBAUPIERRE, Y. (1985) Evoked mechanical responses of isolated cochlear outer hair cells. *Science,* 227**,** 194-6.

BRYSON, K., LUCK, M. & JOY, M. (2000) Applying Agents to Bioinformatics in GeneWeaver. *Proceedings of the Fourth International Workshop on Cooperative Information Agents.* Springer-Verlag.

CHAY, T. R., LEE, Y. S. & FAN, Y. S. (1995) Appearance of phase-locked Wenckebach-like rhythms, devil's staircase and universality in intracellular calcium spikes in non-excitable cell models. *J Theor Biol,* 174**,** 21-44.

CLARK&PARSIA (2004-2008) Pellet OWL DL Reasoner.http://clarkparsia.com/pellet

COLQUHOUN, D. & HAWKES, A. (1995) A Q-Matrix Cookbook: How to write only one program to calculate the single-channel and macrosopic predictions for any kinetic mechanism. IN B, S. & E, N. (Eds.) *Single-channel Recordings.* Plenum, New York.

CRAWFORD, A. C. & FETTIPLACE, R. (1981) Non-linearities in the responses of turtle hair cells. *J Physiol,* 315**,** 317-38.

CUELLAR, A. A., LLOYD, C. M., NIELSEN, P. F., BULLIVANT, D. P., NICKERSON, D. P. & HUNTER, P. J. (2003) An overview of CellML 1.1, a biological model description language. *Simulation,* 79**,** 740-7.

DALLOS, P., HALLWORTH, R. & EVANS, B. N. (1993) Theory of electrically driven shape changes of cochlear outer hair cells. *J Neurophysiol,* 70**,** 299-323.

DARPA (2000) The DARPA Agent Markup Language.http://www.daml.org/

DCMI (1995-2008) Dublin Core Metadata Initiative.http://dublincore.org/index.shtml

DECKER, K., KHAN, S., SCHMIDT, C., SITU, G., MAKKENA, R. & MICHAUD, D. (2002) BioMAS: A Multi-Agent System for Genomic Annotation. *International Journal Cooperative Information Systems,* 11**,** 265--292.

DUPONT, G., SWILLENS, S., CLAIR, C., TORDJMANN, T. & COMBETTES, L. (2000) Hierarchical organization of calcium signals in hepatocytes: from experiments to models. *Biochim Biophys Acta,* 1498**,** 134-52.

FAKLER, B. & OLIVER, D. (2002) Functional properties of prestin: how the motor molecule works. *Biophysics of the Cochlea: From Molecules to Models***,** 5.

FENSEL, D., VAN HARMELEN, F., HORROCKS, I., MCGUINNESS, D. L. & PATEL-SCHNEIDER, P. F. (2001) OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems and Their Applications,* 16**,** 38-45.

FINKELSTEIN, A., HETHERINGTON, J., LI, L., MARGONINSKI, O., SAFFREY, P., SEYMOUR, R. & WARNER, A. (2004) Computational challenges of systems biology. *Computer,* 37**,** 4-26.

FINNEY, A. & HUCKA, M. (2003) Systems biology markup language: Level 2 and beyond. *Biochem Soc Trans,* 31**,** 1472-3.

FIPA (2000) FIPA Agent Communication Language.http://www.fipa.org/specs/fipa00061/

GALE, J. E. & ASHMORE, J. F. (1997) The outer hair cell motor in membrane patches. *Pflugers Arch,* 434**,** 267-71.

GIBSON, A., GAMBLE, M., WOLSTENCROFT, K., OINN, T. & GOBLE, C. (2007) The Data Playground: An Intuitive Workflow Specification Environment. *IEEE International Conference on e-Science and Grid Computing.*

GOBLE, C. (2005) Using the Semantic Web for e-Science: Inspiration, Incubation, Irritation *The Semantic Web – ISWC 2005.* Springer Berlin / Heidelberg.

GOLBECK, J., FRAGOSO, G., HARTEL, F., HENDLER, J., PARSIA, B. & OBERTHALER, J. (2003) The National Cancer Institute's (NCI) Thesaurus and Ontology *Journal of Web Semantics,* 1.

HODGKIN, A. L. & HUXLEY, A. F. (1952) A quantitative description of membrane current and its application to conduction and excitation in nerve. *J Physiol,* 117**,** 500-44.

HOFER, T. (1999) Model of intercellular calcium oscillations in hepatocytes: synchronization of heterogeneous cells. *Biophys J,* 77**,** 1244-56.

HOGUE, C., KARP, P. & SANDER, C. (2002) Biological Pathway Exchange (BioPAX).http://www.biopax.org/

HUDSPETH, A. J. (1983) Mechanoelectrical transduction by hair cells in the acousticolateralis sensory system. *Annu Rev Neurosci,* 6**,** 187-215.

HUDSPETH, A. J. (1985) The cellular basis of hearing: the biophysics of hair cells. *Science,* 230**,** 745-52.

HUDSPETH, A. J. & LEWIS, R. S. (1988) Kinetic analysis of voltage- and ion-dependent conductances in saccular hair cells of the bull-frog, Rana catesbeiana. *J Physiol,* 400**,** 237-74.

IAM (2005) OWL-S upper ontology.http://www.iam.ecs.soton.ac.uk/

IWASA, K. H. (1994) A membrane motor model for the fast motility of the outer hair cell. *J Acoust Soc Am,* 96**,** 2216-24.

KIFER, M., LAUSEN, G. & WU, J. (1995) Logical foundations of object-oriented and frame-based languages. *Journal of the ACM,* 42**,** 741-843.

KUMMER, U., OLSEN, L. F., DIXON, C. J., GREEN, A. K., BORNBERG-BAUER, E. & BAIER, G. (2000) Switching from simple to complex oscillations in calcium signaling. *Biophys J,* 79**,** 1188-95.

LENAT, D. (1996) CycL.http://www.cyc.com/cycdoc/ref/cycl-syntax.html

LENZ, M., BARTSCH-SPORL, B., BURKHARD, H.-D. & WESS, S. (1998) *Case-based reasoning technology. From foundations to applications*, Berlin, Germany, Springer-Verlag.

MADDEN, T. L., TATUSOV, R. L. & ZHANG, J. (1996) Applications of network BLAST server. *Methods Enzymol,* 266**,** 131-41.

MCGUINNESS, D. L. & HARMELEN, F. V. (2004) OWL Web Ontology Language Overview.http://www.w3.org/TR/owl-features/

MISSAN, S. & MCDONALD, T. F. (2005) CESE: Cell Electrophysiology Simulation Environment. *Appl Bioinformatics,* 4**,** 155-6.

MUALLEM, D. & ASHMORE, J. (2006) An anion antiporter model of prestin, the outer hair cell motor protein. *Biophys J,* 90**,** 4035-45.

MYGRID (2005) Taverna.http://taverna.sourceforge.net/

NAGARAJAN, M., VERMA, K., SHETH, A. P., MILLER, J. & LATHEM, J. (2006) Semantic Interoperability of Web Services - Challenges and Experiences. *Proceedings of the IEEE International Conference on Web Services (ICWS'06).*

NHS (2007) SNOMED Clinical Terms.http://www.connectingforhealth.nhs.uk/systemsandservices/data/snomed

NLM (1999-2008) Unified Medical Language System (UMLS).http://www.nlm.nih.gov/research/umls/

NOY, N. F. & MCGUINNESS, D. L. (2001) Ontology Development 101: A Guide to Creating Your First Ontology.

OBO (1999-2008) The Gene Ontology.

OLIVER, D., LUDWIG, J., REISINGER, E., ZOELLNER, W., RUPPERSBERG, J. P. & FAKLER, B. (2001) Memantine inhibits efferent cholinergic transmission in the cochlea by blocking nicotinic acetylcholine receptors of outer hair cells. *Mol Pharmacol,* 60**,** 183-9.

OMG (1997-2008) Unified Modeling Language (UML).http://www.uml.org/

PEARL & JUDEA (1983) *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, New York, Addison-Wesley.

PEREZ-REY, D., MAOJO, V., GARCIA-REMESAL, M., ALONSO-CALVO, R., BILLHARDT, H., MARTIN-SANCHEZ, F. & SOUSA, A. (2006) ONTOFUSION: ontology-based integration of genomic and clinical databases. *Comput Biol Med,* 36**,** 712-30.

PITCHFORD, S. & ASHMORE, J. F. (1987) An electrical resonance in hair cells of the amphibian papilla of the frog Rana temporaria. *Hear Res,* 27**,** 75-83.

PRINCETON (2006) WordNet.http://wordnet.princeton.edu/

RACERSYSTEMS (2004-2008) RacerPro OWL Reasoner.http://www.racer-systems.com/index.phtml

RICCOBENE, T. A., OMANN, G. M. & LINDERMAN, J. J. (1999) Modeling activation and desensitization of G-protein coupled receptors provides insight into ligand efficacy. *J Theor Biol,* 200**,** 207-22.

ROMERO, P., WAGG, J., GREEN, M., KAISER, D., KRUMMENACKER, M. & KARP, P. (2004) Computational prediction of human metabolic pathways from the complete human genome. *Genome Biology,* 6**,** R2.

RYBALCHENKO, V. & SANTOS-SACCHI, J. (2003) Cl- flux through a non-selective, stretch-sensitive conductance influences the outer hair cell motor of the guinea-pig. *J Physiol,* 547**,** 873-91.

SANTOS-SACCHI, J. (1991) Reversible inhibition of voltage-dependent outer hair cell motility and capacitance. *J Neurosci*, 11**,** 3096-110.
SCHUSTER, S., MARHL, M. & HOFER, T. (2002) Modelling of simple and complex calcium oscillations. From single-cell responses to intercellular signalling. *Eur J Biochem*, 269**,** 1333-55.
SHOHAM, Y. (1994) Agent oriented programming: An overview of the framework and summary of recent research. *Knowledge Representation and Reasoning Under Uncertainty.* Springer Berlin / Heidelberg.
SIG (2002-2008) Foundational Model of Anatomy (FMA).http://sig.biostr.washington.edu/projects/fm/index.html
SNEYD, J. & DUFOUR, J. F. (2002) A dynamic model of the type-2 inositol trisphosphate receptor. *Proc Natl Acad Sci U S A*, 99**,** 2398-403.
SNEYD, J., TSANEVA-ATANASOVA, K., YULE, D. I., THOMPSON, J. L. & SHUTTLEWORTH, T. J. (2004) Control of calcium oscillations by membrane fluxes. *Proc Natl Acad Sci U S A*, 101**,** 1392-6.
STANDFORD (1994) Knowledge Interchange Format.http://www-ksl.stanford.edu/knowledge-sharing/kif/
SUN, Z., FINKELSTEIN, A. & ASHMORE, J. (2007) Using ontology with semantic Web services to support modeling in systems biology. *Lecture Notes in Computer Science*, Volume 4832/2007**,** 41-51.
SYCARA, K. & PANNU, A. (1998) The RETSINA multiagent system: towards integrating planning, execution, and information gathering. *2nd International Conference on Autonomous Agents.*
THE MATHWORKS, I. (1994-2008) MATLAB.http://www.mathworks.com/
THE UNIPROT, C. (2008) The Universal Protein Resource (UniProt). *Nucl. Acids Res.*, 36**,** D190-195.
VASTRIK, I., D'EUSTACHIO, P., SCHMIDT, E., JOSHI-TOPE, G., GOPINATH, G., CROFT, D., DE BONO, B., GILLESPIE, M., JASSAL, B., LEWIS, S., MATTHEWS, L., WU, G., BIRNEY, E. & STEIN, L. (2007) Reactome: a knowledge base of biologic pathways and processes. *Genome Biology*, 8**,** R39.
W3C (1996-2003) Extensible Markup Language (XML).http://www.w3.org/XML/
W3C (1997-2001) Resource Description Framework (RDF).http://www.w3.org/RDF/
W3C (1999) XSL Transformations (XSLT).http://www.w3.org/TR/xslt
W3C (2000) Simple Object Access Protocol (SOAP) 1.1.http://www.w3.org/TR/soap/
W3C (2001a) URIs, URLs, and URNs: Clarifications and Recommendations 1.0.http://www.w3.org/TR/uri-clarification/
W3C (2001b) Web Services Description Language (WSDL) 1.1.http://www.w3.org/TR/wsdl
W3C (2004) OWL-S: Semantic Markup for Web Services.http://www.w3.org/Submission/OWL-S/
W3C (2005) Web Service Semantics - WSDL-S.http://www.w3.org/Submission/WSDL-S/
W3C (2006) Semantic Annotations for WSDL Working Group (SAWSDL).http://www.w3.org/2002/ws/sawsdl/
WIKIPEDIA_FRAME (2007) Frame Language.http://en.wikipedia.org/wiki/Frame_%28data_structure%29

WIKIPEDIA_ONTOLOGY (2008) Ontology_(computer_science).http://en.wikipedia.org/wiki/Ontology_(computer_science)

WOODS, N. M., CUTHBERTSON, K. S. & COBBOLD, P. H. (1987) Agonist-induced oscillations in cytoplasmic free calcium concentration in single rat hepatocytes. *Cell Calcium,* 8**,** 79-100.

YERGENS, D., HINER, J., DENZINGER, J. & NOSEWORTHY, T. (2006) Multi Agent Simulation System for Rapidly Developing Infectious Disease Models in Developing Countries. *Proc. AAMAS 2006 WS on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics.* Hakodate, Japan.

ZHENG, J., SHEN, W., HE, D. Z., LONG, K. B., MADISON, L. D. & DALLOS, P. (2000) Prestin is the motor protein of cochlear outer hair cells. *Nature,* 405**,** 149-55.

Appendixes

Table 1. Parameters for Hofer's calcium oscillation minimum model

Parameter	Description	Value
ρ	Ratio of the total area of plasma membrane to the effective capacity of Ca_{cyt}	0.02
α	Ratio of the total area o the ER to the total area of the plasma membrane	2.0
β	Ratio of the effective capacity of Ca_{er} to the effective capacity of Ca_{cyt}	0.2
k_1	Rate constant for $InsP_3$ mediated release of Ca^{2+} from ER	$40\,s^{-1}$
k_2	Rate constant for small leak flux of Ca^{2+} into cytosol	$0.02\,s^{-1}$
d_a	Threshold constant for Ca^{2+} mediated fast release of Ca^{2+} from ER	$0.4\,\mu M$
d_p	Threshold constant for $InsP_3$ mediated fast release of Ca^{2+} from ER	$0.2\,\mu M$
d_1	Fit parameter for $InsP_3$ mediated sustained Ca^{2+} from ER	$0.3\,\mu M$
d_2	Fit parameter for $InsP_3$ mediated sustained Ca^{2+} from ER	$0.4\,\mu M$
d_3	Fit parameter for $InsP_3$ mediated sustained Ca^{2+} from ER	$0.2\,\mu M$
v_0	Constant rate of background Ca^{2+} influx across cell membrane	$0.2\,\mu M s^{-1}$
v_3	Rate constant for Ca^{2+} uptake from cytosol into ER	$9.0\,\mu M s^{-1}$
v_4	Rate constant for Ca^{2+} efflux across cell membrane	$3.6\,\mu M s^{-1}$
v_c	Maximal rate for $InsP_3$ induced Ca^{2+} influx across cell membrane	$4.0\,\mu M s^{-1}$
K_0	Michaelis constant for $InsP_3$ induced Ca^{2+} influx across cell membrane	$4.0\,\mu M$
K_3	Michaelis constant for Ca^{2+} uptake from cytosol into ER	$0.12\,\mu M$
K_4	Michaelis constant for Ca^{2+} efflux across cell membrane	$0.12\,\mu M$

Ca_{cyt}	Initial value of free cytosolic calcium concentration	$0.1\,\mu M$
Ca_{w}	Initial value of total free calcium concentration in the cytosol and ER	$3.5\,\mu M$
IP_3	Initial value of $InsP_3$ concentration	$2.0\,\mu M$

Table 2 Parameters for Riccobene's G-protein coupled receptor activation model

Parameter	Definition	Value
α	Effect of ligand binding on receptor activation	$10-10^6$
k_{ds}	Desensitization rate constant	$10^{-4}-10s^{-1}$
k_{fR}	Forward rate constant for R to R* conversion	$10s^{-1}$
K_{act}	Equilibrium constant for R to R* conversion	1.0×10^{-4}
k_a	Activation rate constant for G to G*	$1.0\times10^{-7}s^{-1}$
k_i	Inactivation rate constant for G*	$2.0\times10^{-1}s^{-1}$
L	Ligand concentration	$10^{-12}-10^{-6}M$
R_{tot}	Total number of surface receptors	$5.5\times10^4 cell^{-1}$
G_{tot}	Total number of G-proteins	$1\times10^5 cell^{-1}$
k_f	Ligand association rate constant for R	$1.2\times10^8 M^{-1}s^{-1}$
k_r	Ligand dissociation rate constant for R	$7.3s^{-1}$
k_{f2}	Ligand association rate constant for Rds	$1.3\times10^2 M^{-1}s^{-1}$
k_{r2}	Ligand dissociation rate constant for Rds	$4.0\times10^{-5}s^{-1}$
K_d	Equilibrium dissociation constant (kr/kf)	$6\times10^{-8}M$