

CelOWS: A Ontology Based Framework with Web Services for Conceptual Modeling in Systems Biology

Ely Edison Matos¹, Fernanda Campos^{1,2}, Regina Braga^{1,2},
Rodrigo Weber dos Santos^{1,3}

¹ Master Program in Computational Modeling

² Núcleo de Pesquisa em Qualidade de Software

³ FISIOCOMP- Lab. of Computational Physiology

^{1,2,3} Universidade Federal de Juiz de Fora

Campus Universitário S/N - Martelos 36036-900

Juiz de Fora – MG – Brasil

+55.32.2102.3378

{ely.matos,fernanda.campos,regina.braga,rodrigo.weber}@ufjf.edu.br

1 Introduction and Motivation

Tools, methods and techniques made possible a better understanding of functions, structures and processes related to biophysics and physiology. Computational models have become valuable tools for the understanding of multi-scale and multi-physics phenomena that underlie complex biophysical structures and processes. In silico models allow different information acquired from different physical scales and experiments to be combined and to complement each other, providing a better picture of the involved processes and structures [1].

Several important questions arise during the development and use of biophysical models. One refers to appropriate representation of the model. Diagrams, literal description and equations can be used in the publication of models, but typographical errors as well as the lack of the definition of initial conditions and parameters can occur. Other question is related to the implementation. The complex mathematical equations usually demand advanced numerical methods and the implementation of such methods may be a limiting factor for use the model properly.

These questions led to creation of CellML [2], a specific markup language to represent biological models. As CellML is an XML-based language, biological and mathematical concepts can be used in various computer applications to solve the model. CellML has been designed to support large-scale modelling efforts: it is modular, enabling models and portions of models to be constructed independently and later integrated into a larger coherent model. It uses MathML [3] to provide both a human and computer-readable representation of the mathematical relationships of biophysical models.

CellML can be used to represent, to store and to share models. However, CellML was created with the intention of describing published models, then it is not totally adjusted to the process of creation of new models, as it does not provide annotation mechanisms to facilitate reuse and modification of new components.

Another important question is about validation of CellML models. Based on XML, the validation of the model is only syntactic. Through the use of DTD (Document Type Definition) or Schemas [4] the model can be validated considering the syntax errors and its compliance to CellML specification [5]. Semantic questions either can not be effectively treated (e.g. to prevent that a component "ionic channel" contains a component "membrane") or is left to the implementation phase (e.g. conformity of different units of measure as "seconds" and "milliseconds").

In this paper, we propose the use of ontologies, description logics and semantic rules to handle these issues. An ontology, named Cell Component Ontology – CELO, is presented. This ontology aims to extend (not to replace) the description provided by CellML, improving the validation, composition and sharing of physiological cell models. The ontology is the base component of a prototype framework, called CelOWS, for modelling of biological systems, involving the search, annotation, composition and execution of biological models.

This paper is organized as follow. In section 2 a summarized version of CELO ontology is presented. Section 3 explores the integration with CellML models. Section 4 presents some related works and section 5 discusses the conclusions and future work.

2 Cell Component Ontology - CELO

The integration of ontologies with CellML will play a major role in the development of software tools to biological modelling [6]. An ontology can be described as a explicit specification of a concept that can be shared [7]. A typical ontology is composed of a taxonomy and a set of inference rules. Taxonomy describes classes of objects and relationships between them. The inference process allows capturing the knowledge that is not explicit in the taxonomy.

The specific goal of the proposed Cell Component Ontology – CELO – is to provide a high-level description of a CellML model, working as a “conceptual model” that adds semantics to physiology models. The use of semantics to express the intrinsic knowledge in the model aims to improve the validation of the model, effective reuse of components already defined in another model, automate some composition processes based on components characteristics, and to build repositories of available models in the Web, where the search can semantically be carried through.

At application level, CELO uses OWL (Ontology Web Language) [8] as representation language and SWRL (Semantic Web Rule Language) [9] for the accomplishment of inferences on the ontological individuals. SWRL complements the ontology with information not directly processed by the inference machine.

2.1 CELO Structure

The CELO structure is motivated by the necessity of providing 3 essential types of knowledge on a model: (a) which quantities are measured and which unities are used by the variables of the model? (b) which concepts of biology domain are associated to a model? (c) which are the components of the model and how can they be accessed? These questions led to the definition of three general classes in the top-level of the ontology, presented in Fig.1 (the OWL complete code, with properties, individuals and restrictions of CELO can be obtained from <http://celo.mmc.ufjf.br>).

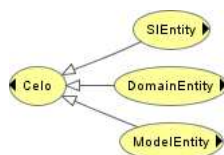


Fig. 1. CELO top-level classes.

SIEntity Classes. This class and his subclasses, which have a generic application, define a dictionary of unities and quantities what are used together with domain concepts and variables of the model. This dictionary is based on the International System of Unities (SI) [10]. New unities can be registered to allow the integration with CellML. The measured quantities also are based on SI. Fig. 2 presents the structure of class *SIEntity*.

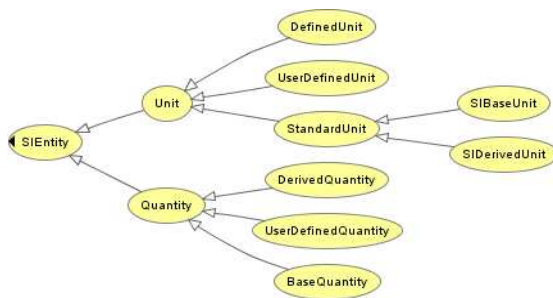


Fig. 2. CELO *SIEntity* classes.

DomainEntity Classes. This class and his subclasses define the terms that will be used like a vocabulary shared by the developers in the creation of models. These terms have an associate semantics, allowing to description of a model in higher level. Fig. 3 presents the structure of the class *DomainEntity*.

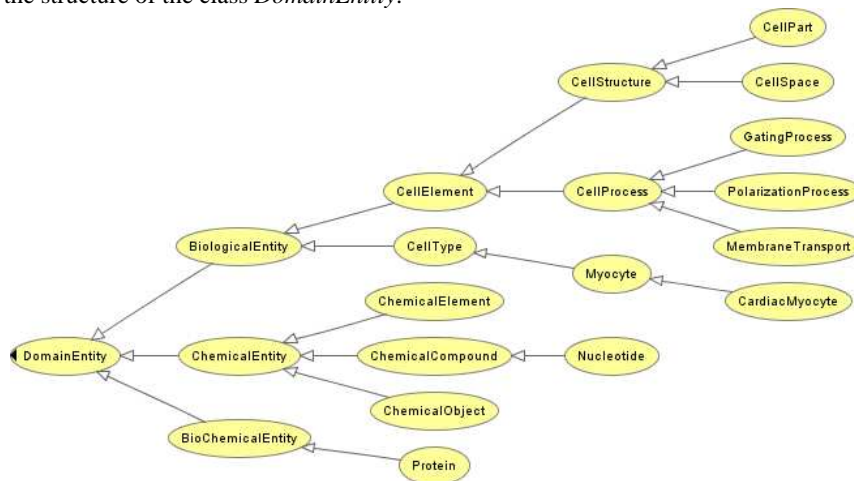


Fig. 3. CELO *DomainEntity* classes.

BiologicalEntity refers to the concepts of the area of the Biology. *CellElement* refers to the physical structure of the cell (*CellStructure*) and to the processes associated to the electrophysiology (*CellProcess*). The elements of interest in the cell are registered in *CellPart* and the location of these elements, in spatial terms, in *CellSpace*.

ChemicalEntity refers to the concepts of the area of the Chemistry. *ChemicalElement* refers to the chemical elements (such as sodium, potassium, etc.). *ChemicalCompound* refers to more complex chemical compounds. *ChemicalObject* is a generic concept for definition of atoms, ions and molecules.

BioChemicalEntity refers to concepts of the area of the Biochemistry, which studies the chemistry of the biological processes that take place in the lively beings.

ModelEntity Classes. This class and his subclasses define the concepts that will be used like meta-model for the biological represented model. The objective is to have a description of high level of the model, making reference to a model CellML for the questions of simulation. Fig. 4 presents the structure of the class *ModelEntity*.

With the objective to treat a model like a service web, what has interfaces and can be "executed" (simulated), the structure of the class *ModelService* is inspired in the ontology OWL-S [11], used for semantic description of services web. Each instance is associated by a instance of *ModelProfile*, *ModelGrounding* and *ModelProcess*.

ModelProfile tells about what is the model: what the components of the model are, if it is associated to some specific compartment of the cell and what are the biological entities (instance of *BiologicalEntity*) associated to a model. *ModelGrounding* specifies which is the associated model for simulation, e.g. a CellML model. *ModelProcess* indicates how the model can be used, in other words, which parameters of input (*ModelParameterIn*) and output (*ModelParameterOut*) are associated to the model interface (*ModelInterface*).

ModelObject groups the objects that a model composes. *Equation* makes a list of the equations that implement mathematically the model, expressed in MathML. *Variable* represents the variables of the model. *ModelVariable* describes like the variables they participate in the equations of the model and *ComponentVariable* makes a list of the variables of each component.

Model represents the model, that can be atomic (containing only a single component) or composed (containing two or more components). *Component* represents the components of the model. These components can be described in the model itself (*InternalComponent*) or they can store references URI to external models (*ExternalComponent*).



Fig. 4. CELO *ModelEntity* classes.

2.2 Knowledge Reuse

An initial CELO Model is built from an existing CellML Model. An important premise about using CELO is that each CellML models represents a "atomic" component, what means that a component sufficiently self-contained can be used to compose a more complex component or model.

To build this initial model, some knowledge can be extract from CellML model, in spite of its just syntatic representation. It's very common that variables and components names are meaningful, like "sodium_channel" and "potassium_channel_gate" to represent a sodium channel and a gate on a potassium channel. Even simple names like "I_Na" (electric current associated to sodium ions) can be useful to extract semantic.

The strategy to capture this knowledge was to break the variables and components names in smaller tokens and to store these tokens in a datatype property called “hasDetail”. After, SWRL rules are applied to match these simple strings with individuals on CELO ontology. Together with other direct mappings (e.g. units names), we can obtain a semantically richer (although not complete) model from a CellML model.

2.3 Rules

Semantic rules are used with two basic purposes:

1. to infer knowledge that is implicit in the CellML model (e.g. name of variables), considering the already existent knowledge (for example, variables that are dimensionless must not be associated to chemical elements).
2. to validate semantically the model regarding his consistency; an example of validation is the checking if the grouping of the components does not injure the anatomical hierarchy.

The rules are defined using SWRL. Some rules used in the ontology are presented following:

- Set *hasCellElement* variable property:
`Variable(?x1) ^ hasDetail(?x1, ?x2) ^ CellElement(?x3) ^ hasName(?x3, ?x4) ^
swrlb:stringEqualIgnoreCase(?x2, ?x4) → hasCellElement(?x1, ?x3)`
- Set *hasMeasure* variable property, based on used unit:
`Variable(?x1) ^ hasDetail(?x1, ?x2) ^ StandardUnit(?x3) ^ hasName(?x3, ?x4) ^
swrlb:stringEqualIgnoreCase(?x2, ?x4) ^ measures(?x3, ?x5) → hasMeasure(?x1, ?x5)`
- Set *hasVariableUnit* variable property, based on used unit:
`Variable(?x1) ^ hasDetail(?x1, ?x2) ^ Unit(?x3) ^ hasName(?x3, ?x4) ^
swrlb:stringEqualIgnoreCase(?x2, ?x4) → hasVariableUnit(?x1, ?x3)`
- Set *hasDomainEntity* associated with a chemical element, excluding dimensionless variables:
`Variable(?x1) ^ hasDetail(?x1, ?x2) ^ ChemicalEntity(?x3) ^ hasName(?x3, ?x4) ^
swrlb:stringEqualIgnoreCase(?x2, ?x4) ^ hasVariableUnit(?x1, ?x5) ^ hasName(?x5, ?x6) ^
swrlb:notEqual(?x6, "dimensionless") → hasDomainEntity(?x1, ?x3)`

3 Using CellML Models with CELO

Considering the applicability of CellML for the description and simulation of models, the availability of hundreds of models and the semantic award given by the CELO, some mechanism of automatic integration between CellML and CELO must exist.

As part of efforts of CelOWS framework development, some auxiliary tools have been used. These tools allows to breaking an existing CellML model in many “atomic” models (each of them containing a single component), create a CELO description from a CellML model and create a CellML model from a composition of CELO models.

3.1 CELO Description

The creation of a initial CELO description from a CellML model is a process with many steps (Fig. 5). First, a XSL Transformation [12] is used to create an intermediate XML representation of model. After, the model CellML is analysed and for each component and there are extracted the variables and parameters of the mathematical equations. The program XML2OWL [13] is executed, taking the intermediate XML as input. Based on a configuration file, an intermediary OWL is built, where the XML elements are turned into individuals of CELO. SWRL Rules are applied (when the *detail* property is compared with property *hasName* of many individual from ontology). Finally, a reasoner is executed to classify the variables and components of model.

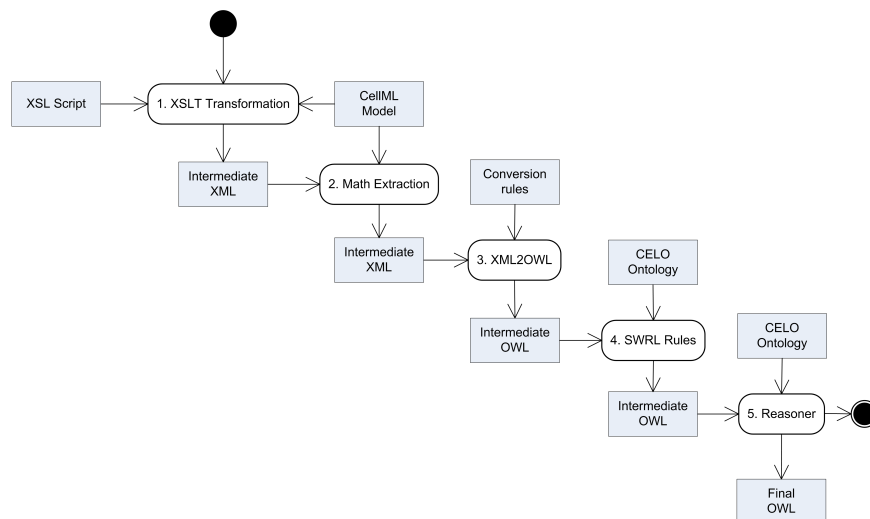


Fig. 5. Creating a initial CELO description from a CellML model.

3.2 CELO Composition

The composition of atomic models is one of the specific objectives of the design of CELO ontology. The goal is to promote the reuse of existent components through the composition of the most simple models, producing more complex models. The composition produces a new CellML model, through the import of previously defined components and automatic connection of these components.

The composition process uses a XML configuration file to indicate which models will be composed and which structure of the model. Figure 6 shows a sample configuration file.

```

<?xml version="1.0" encoding="UTF-8"?>
<composition>
  <global uri="http://celo.mmc.ufjf.br/owl/noble_1962/environment.owl"/>
  <model uri="http://celo.mmc.ufjf.br/owl/noble_1962/membrane.owl" uses="global">
    <model uri="http://celo.mmc.ufjf.br/owl/noble_1962/leakage_current.owl" uses="global"/>
    <model uri="http://celo.mmc.ufjf.br/owl/noble_1962/sodium_channel.owl" uses="global">
      <model uri="http://celo.mmc.ufjf.br/owl/noble_1962/sodium_channel_h_gate.owl"/>
      <model uri="http://celo.mmc.ufjf.br/owl/noble_1962/sodium_channel_m_gate.owl"/>
    </model>
    <model uri="http://celo.mmc.ufjf.br/owl/noble_1962/potassium_channel.owl" uses="global">
      <model uri="http://celo.mmc.ufjf.br/owl/noble_1962/potassium_channel_n_gate.owl"/>
    </model>
  </model>
</composition>
  
```

Fig. 6. A sample XML configuration file for models composition.

The connection of the components is done through a semantic matching of parameters of the first component and parameters of second component. It is tested if both parameters measure the same quantities, both are associated to the same chemical elements and, finally if both has the same name. A SPARQL [14] query (Fig. 7) with a post-process is used to find compatible parameters.

```

select ?x0 ?y0 ?x1 ?y1 ?x2 ?y2 ?x3 ?y3 ?x4 ?y4 ?x5 ?y5 ?x6 ?y6 WHERE {<ns1:_ModelInterface>
celo:hasInterface ?x1.
  ?x1 a celo:ModelParameterIn.
  OPTIONAL{
    ?x1 celo:hasDomainEntity ?x2.
    ?x2 a celo:CellPart.
    ?x2 celo:contains ?x5.
    ?x2 celo:isContainedIn ?x6}
  OPTIONAL{
    ?x1 celo:hasMeasure ?x3}
  OPTIONAL{
    ?x1 celo:hasDomainEntity ?x4.
  
```

```

        ?x4 a celo:ChemicalElement}
?x1 celo:hasName ?x0.
<ns2:_ModelInterface> celo:hasInterface ?y1.
?y1 a celo:ModelParameterOut.
OPTIONAL{
    ?y1 celo:hasDomainEntity ?y2.
    ?y2 a celo:CellPart.
    ?y2 celo:contains ?y5.
    ?y2 celo:isContainedIn ?y6}
OPTIONAL{
    ?y1 celo:hasMeasure ?y3}
OPTIONAL{
    ?y1 celo:hasDomainEntity ?y4.
    ?y4 a celo:ChemicalElement}
    ?y1 celo:hasName ?y0
}

```

Fig. 7. A SPARQL query to connect components.

3.3 Searching models

The models, expressed in OWL, are stored inside a relational database. This allows the user to make “semantic queries”, using SPAQL syntax. Figure 8 show some sample queries.

```

List all models
PREFIX celo: <http://celo.mmc.ufjf.br/ontologies/celo.owl#>
select ?x WHERE {
    ?x a celo:Model }

List all components
PREFIX celo: <http://celo.mmc.ufjf.br/ontologies/celo.owl#>
select ?x WHERE {
    ?x a celo:Component }

List variables related with Action Potential
PREFIX celo: <http://celo.mmc.ufjf.br/ontologies/celo.owl#>
select ?x WHERE {
    ?x a celo:ActionPotential }

List components that variables are related with Action Potential
PREFIX celo: <http://celo.mmc.ufjf.br/ontologies/celo.owl#>
select ?x WHERE {
    ?x a celo:Component .
    ?y celo:isInterfaceVariableOf ?x
    ;a celo:ModelParameter
    ;a celo:ActionPotential }

List components with parameters related to Electric Current and Sodium
PREFIX celo: <http://celo.mmc.ufjf.br/ontologies/celo.owl#>
select ?x WHERE {
    ?x a celo:Component .
    ?y celo:isInterfaceVariableOf ?x
    ;a celo:ModelParameter
    ;celo:hasDomainEntity celo:Sodium
    ;celo:hasMeasure celo:EletricCurrent
}

```

Fig. 8. Sample SPARQL queries to find models and components.

4 CeloWS Framework

4.1. Goals

The main goal of CelOWS framework is to provide a infrastructure to storage, search, retrieve, compose and execute (simulate) components of biological models using ontologies. The framework must allow:

- Recording and storing components inside distributed relational;
- Queries on database based on semantic expressed by models;
- Composition of models, creating more complexes models;
- Execution of models, it means, submission of models to a simulation tool like PCEnv or AGOS..

4.2. Architecture

A high-level view of framework architecture is showed at Fig. 9. *URI CelO* represents a URI of CelO Model. This framework implements concepts of SOA (*Service Oriented Architecture*) and it work with three distinct tiers:

- CelOWS: the framework, implemented as a web service.
- Backend: a service layer, with database access and a interface with a simulation tool for the CellML Model.
- Client: a user interface to access the framework. It can be built with any language that allow access to web services.

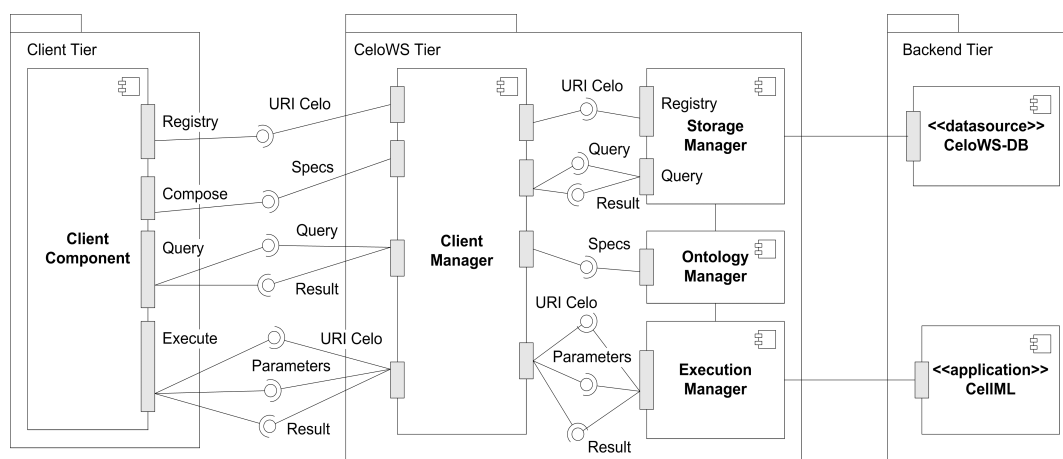


Fig. 9. Visão geral da arquitetura do framework CelOWS.

There are four services:

- **Registry:** To record a model inside system. From URI, the model is retrieved, a inference engine is applied and the model is stored inside a relational database.
- **Compose:** Composition of models. The user provide a XML specification of composition. A new CellML model is built from the code referenced by each CelO model.
- **Query:** Searching a model. The user provide a SPARQL query and he receives the URIs of models, components or variables that match the query.
- **Execute:** Simulation of a model. The user provide the model URI and the parameters to be used. The CellML model is retrieved and send to a simulation tool. The results are returned to user.

5 Related Works

A discussion about the use of ontologies in CellML is presented in [18]. During the CellML Workshop (2007) the theme was again discussed [19]. The CELO Ontology is a contribution as it represents the cell models using ontological languages and is integrated to CellML.

A research with ontologies applied to cell models is described in [20]. The authors present methods to support the development of complex structure of cell models, using a markup language (PMSML) and the Cell Model Ontology. The CELO ontology has a very similar purpose, but it uses more standardized languages, as OWL, SWRL and XSLT. It allows the use of a great variety of tools, besides the possibility of a collaborative work in the development of the ontology itself.

A specification language, called Composite Model Description Language (CDML), is proposed in [21]. Although it is more rich than CellML, it is also XML-based and the issues described previously are the same. CELO uses ontologies to handle the process of models composition.

Another group uses the association of ontologies and web services to support the modeling of biological systems [22]. An ontology is presented to represent the meta-model in OWL. OWL-S is used to specify the parameterization and semi-automatic composition of web services for the model execution. CELO ontology includes more than the meta-model, since biological domain concepts are described.

References

1. Garry A, Nickerson D., Cooper J., Santos R. W., McKeever S., Nielsen P., Hunter P.: CellML and Associated Tools and Techniques. (2007)
2. CellML – Cell Markup Language. <http://www.cellml.org>
3. MathML – Mathematical Markup Language. <http://www.w3.org/Math/>
4. XML Schemas. <http://www.w3.org/XML/Schema>
5. Cuellar, A.A., Nielsen, P.F., Bullivant, D.P., Nickerson, D., Hedley, W., Nelson, M., Lloyd, C.M.: CellML Specification 1.1. (2006).
6. Lloyd, C. M., Halstead, M. D. B., Nielsen, Poul. CellML: its future, present and past. *Progress in Biophysics & Molecular Biology* 85 (2004) 433–450. (2004)
7. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American* (2001)
8. OWL Web Ontology Language. <http://www.w3.org/TR/owl-features/>
9. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. <http://www.w3.org/Submission/SWRL/>
10. NIST: The International System of Units (SI). Barry N. Taylor, editor (2001)
11. OWL-S Semantic Markup for Web Services. <http://www.w3.org/Submission/OWL-S>.
12. XSL Transformations (XSLT). <http://www.w3.org/TR/xslt>
13. Amin, M. A., Morbach, J.: XML to OWL Converter. <http://www.lpt.rwth-aachen.de/Publication/abstractND.php?Nummer=LPT-2007-02>
14. SPARQL – Query Language for RDF. <http://www.w3.org/TR/rdf-sparql-query>.
15. CellML repository. The Modification of the Hodgkin-Huxley Equations Applicable to Purkinje Fibre Action and Pace-Maker Potentials. http://www.cellml.org/models/noble_1962_version05
16. CellML repository. A quantitative description of membrane current and its application to conductance and excitation in nerve. http://www.cellml.org/models/hodgkin_huxley_1952_version07
17. Barbosa, C., Santos, R., Amorim, R., Ciufo, L., Manfro, F., Oliveira, R., Campos, F.: A Transformation Tool for ODE based models. *Lecture Notes in Computer Science* 3991, 69–75. (2006)
18. Nielsen P., Halstead M., Cuellar A., Dunstan M., Bullivant D., Hunter P.: Ontologies in CellML: A Versatile Method to Describe Cellular Models. 11th International conference on Intelligent Systems for Molecular Biology (ISMB). International Society for Computational Biology (ISCB). (2003)
19. Halstead M.: Ontologies and Repositories. CellML Workshop 2007. <http://www.cellml.org/workshop2007>.
20. Takao, S., Kazuhiro, K., Akira, A., Takeshi, I., Masanori, K., Tetsuya, M.: Computer Support for Physiological Cell Modelling using an Ontology on Cell Physiology. In: 28th Annual International Conference of the IEEE Volume , Issue , Aug. 2006 Page(s):4171 – 4174. IEEE Press, New York (2006).
21. Margoninski, O., Saffrey, P., Hetherington, J., Finkelstein, A. and Warner, A., A Specification Language and a Framework for the Execution of Composite Models in Systems Biology, *LNCS Transactions on Computational Systems Biology* (2006).
22. Sun, Z., Finkelstein, A., Ashmore, J.: Using Ontology with Semantic Web Services to Support Modeling in Systems Biology. International Workshop on Approaches and Architectures for Web Data Integration and Mining in Life Sciences (WebDIM4LS), Nancy, France (2007)