CelOWS: A Service Oriented Architecture to Define, Query and Reuse Biological Models

Ely Edison Matos[1], Fernanda Campos[1,2], Regina Braga[1,2],
Rodrigo Weber[1,3], and Daniele Palazzi[1,2]

[1] Master Program in Computational Modeling
[2] Software Quality Research Group
[3] Computational Physiology Laboratory
Federal University of Juiz de Fora - Juiz de Fora – Minas Gerais – Brazil
{ely.matos,fernanda.campos,regina.braga,
rodrigo.weber}@ufjf.edu.br,
daniele.palazzi@ice.ufjf.br

Abstract. The amount of information generated by biological research has lead to an intensive use of models. Mathematical and computational modeling needs accurate description to share, reuse and simulate models as formulated by original authors. In this paper, we introduce the Cell Component Ontology - CelO, expressed in OWL-DL. This ontology captures both the structure of a cell model and the properties of functional components. We use this ontology in a Web project – CelOWS - to describe, query and compose CellML models. It aims to improve reuse and composition of existent components and allow semantic validation of new models.

Keywords: Semantic web services, ontology, SOA, e-Science.

1   Introduction

Intense research in biological science has generated great volume of data. Tools, methods and techniques improve the understanding of new functions, structures and processes related to biophysics and physiology. The increase of the computational power and the use of numerical methods stimulate the development and the application of more complex models [1]. These models allow the combination of different physical experiments and in different scales into a computational simulation, providing an accurate view of the studied phenomena.

   The computational simulation of a model involves two important aspects. The first one is the model representation. Although diagrams, literal description and equations can be used to publish the models, typographical errors, as well as the necessary conditions to the simulation may generate fatal errors. The second aspect is concerned to the implementation phase. The necessity to apply advanced numerical methods creates a barrier for the effective use and study of the model.

   These questions have stimulated the development of CellML [2], a markup language for representing biological models. Based on XML (eXtensible Markup Language), CellML specifies elements that can be used to represent a model in a formal

way, without ambiguities, legible for humans and computer-readable. The mathematical equations are represented in MathML, which makes it independent of a specific implementation language. CellML can be used to represent, store and share models. However, the CellML approach does not provide mechanisms to make component reuse an easy going step.

This research is aimed at using and discussing ontologies and semantic rules for creation, validation, storage and sharing of biological models, through the service oriented architecture - CelOWS. It allows the storage, research, reuse, composition and execution of described models using the CelO (Cell Component Ontology) ontology. The main goal of the CelOWS is the composition of a new model from the reuse of different models from a specific repository. The CelO ontology, described in OWL-DL [3], aims to add semantics to the models. It allows the representation of the intrinsic knowledge, to improve its validation, reuse components from other models, automate some composition processes and develop model repositories where queries are semantically carried out.

The remainder of the paper is organized as follows. The next section discusses the background of the work. Section 3 presents the proposed CelO ontology and its main elements. Section 4 describes the CelOWS architecture, use cases, validation process and examples of application. Related works are presented in Section 5, while Section 6 concludes the paper.

2   Related Concepts

In this research context, biological models are related to mathematical and computational representation of some biological properties. In the field of physiology and electrophysiology the functional elements of the system are generally represented by an abstraction named "component". Each component is a mathematical model that aims to represent the behavior of the biological element. The interaction between the biological elements is represented by the association of different components.

The electrophysiology models currently span from simple models of the electric activation based on polynomials to three-dimensional complex models [4]. The cell components models can be represented through diagrams and have a series of associated equations. Diagrams and text descriptions, however, can have typographical errors or incorrect initial or contour conditions to the simulation. These concerns were recently addressed via the development of the CellML language for representing biological models.

CellML establishes a standard format for defining and sharing biological models. The models are then represented as an interconnected network of components. Each component represents a biological element of interest as, for example, the cell membrane or an ionic channel. From the viewpoint of computational representation, a component is a unit that can interact, add or encapsulate other components.

A model receives a name and an unique identification (cmeta: id) used to identify the model URI. The mathematical equations are expressed in MathML and they describe the behavior of each component in the model. A variable is a nominated entity associated to a component, representing amounts used in the equations. An initial value and some attributes (units, public and private interfaces, etc.) can be associated to a variable. Connections between components are represented through the variables mapping.

3   Cell Component Ontology - CelO

Ontology can be described as a formal and explicit specification of a shared concep-tualization [6] [7]. The ontology construction implies in acquiring the domain knowl-edge and collecting the appropriate information that formally defines the domain terms. The CelO ontology can be extended for the Biology area, but by now it focus on the sub-domain of cardiac electrophysiology. This restriction aims to diminish the complexity level and allow a gradual evaluation of the ontology, in terms of its com-pleteness and functionality.

The specific goal of the CelO ontology is to describe the semantic of biological models. The use of the semantic to express the intrinsic model knowledge aims to improve its validation, reuse other models components, automate the processes of models composition and construct Web repositories with semantic queries. The struc-ture of the ontology aims its integration with simulations of CellML models, using tools as AGOS [8] and PCEnv [9].

For the development of the CelO ontology the Methontolgy methodology was adopted, which includes four phases [10]: specification, conceptualization, formaliza-tion and implementation. In the specification activity a document was defined with the ontology goals and main terms. During the conceptualization phase a set of repre-sentations organized the domain knowledge: terms glossary, concepts classification tree, diagrams of binary relationship and concepts dictionary. The activities of formal-ization and implementation phases were developed and the ontology built using the Protegé tool. The maintenance and evolution will occur whenever modifications in the ontology structure are necessary (Fig. 1)

The CelO ontology has three general classes in its top level and its structure pro-vides three essential types of knowledge: SIEntity (quantities and units associated to variables of the model), DomainEntity (Concepts of the Biological domain) and ModelEntity (Model components and interfaces).
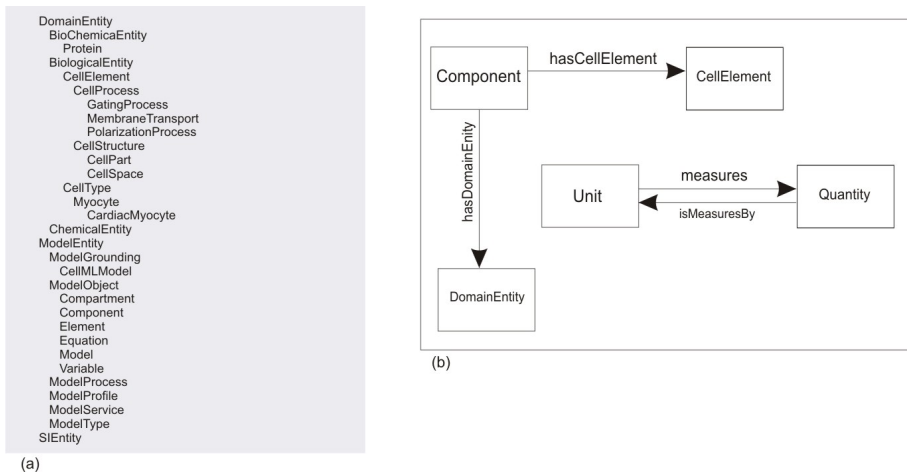
Fig. 1. Intermediate representation for the CelO ontology: (a) part of a classification concepts tree; (b) diagram of binary relationship of some concepts

In this paper we present only the *ModelEntity* subclasses. The complete CelO ontology and the performed tests are available in http://celo.mmc.ufjf.br.

The *ModelEntity* class and its subclasses (Fig. 2) define the used concepts in the biological model representation. The goal is to have a high level description, making reference to the CellML models for the simulation activity.

A challenging issue in the CelOWS architecture is to treat a model as a web service that has interfaces and can "be executed" (through simulation). The *ModelService* class provides a way of organizing the model vision as a service. Its structure is similar to OWL-S ontology [11], for semantic description of web service. An instance of the *ModelService* exists for each model and is associated to an instance of *ModelProfile, ModelGrounding* and *ModelProcess.*

ModelProfile gives information about the model: components association with some specific compartment of the cell and the biological entities associated to the model. *ModelGrounding* specifies the associated models, in case of a simulation. The *CellMLModel* class stores the URI of the associated CellML model. *ModelProcess* indicates how the model can be used, which are the input parameters (*ModelParameterIn*) and output ones (*ModelParameterOut*) that are associated to the interface model (*ModelInterface*). These parameters are directly associated to the model variables and can be used in the simulation or composition processes. *ModelType* is a generic class, used to characterize the biological problem of the model and its mathematical modeling.

ModelObject groups objects that compose a model and that are directly related to the underlying model. *Equation* names equations that implement the math model, expressed in MathML. *Variable* represents model variables. *ModelVariable* describes the variables roll in model equations and *ComponentVariable* is related to the variable of each model component. These variables can be used in the interface of the component (*InterfaceVariable*) or be local (*LocalVariable*). *DomainVariable* associates model variables with concepts of the *DomainEntity* class. *Model* represents the model itself, it can be atomic (only one component) or composed (two or more components). *Component* represents model components, that can be described in the proper model (*InternalComponent*) or can make reference to external ones (*ExternalComponent*). A model can be composed simultaneously by internal and external models.

The semantic rules are used to infer knowledge that is implicit in existing CellML models. The association between component variables and ontology individuals are defined and extracted from details in the CellML model. For example, the information may be extracted from the name of a variable and from predefined knowledge. For instance, we may have a rule that forbids the association of variables without dimension to chemical elements. Actually the rules can be also used to validate semantically the model in terms of consistency and completeness. A validation example is if the components group follows the anatomical hierarchy. Some rules of the CelO ontology, written in SWRL, are presented in Fig. 3:
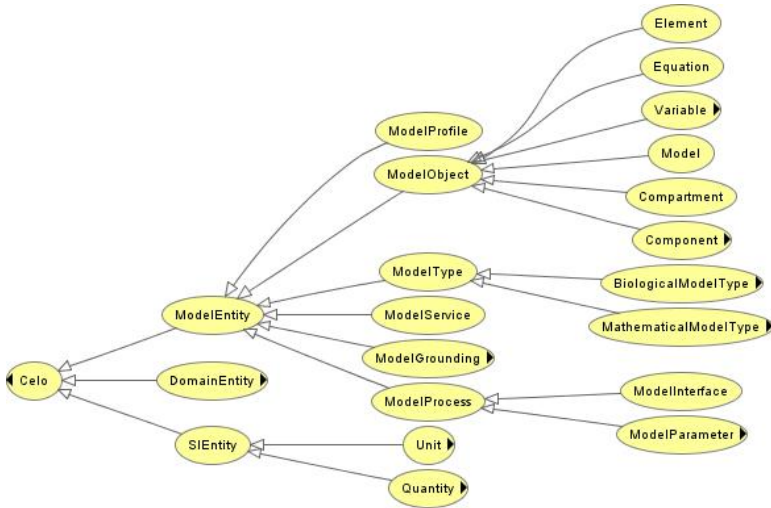
Fig. 2. Parcial representation of the CelO ontology in Protegé

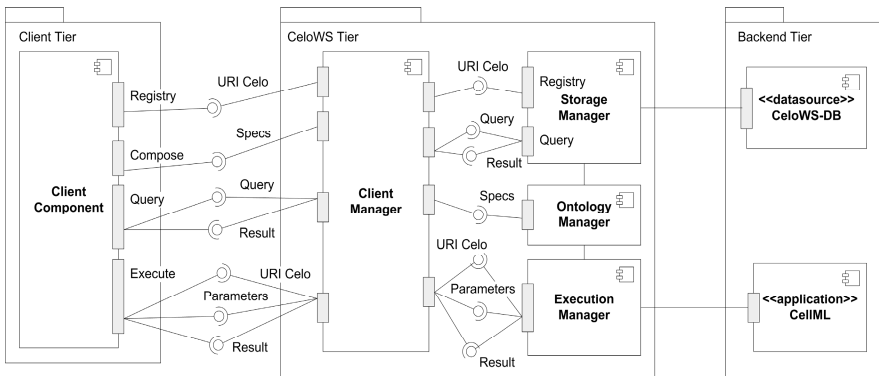**Fig. 3.** Examples of the CelO ontology rules

Fig. 4. Partial view of CelOWS framework architecture

There are different languages to represent the ontologies. Our research uses OWL , which is recommended by the W3C as the standard language used for web semantics projects. For inferences on individuals of the ontology the SWRL (Semantic Web Rule Language) was used [12]. SWRL makes possible the enrichment of the ontology with information not directly processed by the inference engine. This language was select because it is based on OWL.

4   CelOWS: An Architecture for e-Science Applications

The CelO ontology is the base for querying and composition of biological models in a service oriented architecture named CelOWS. This architecture uses the concepts of ontology repository and semantic web services. CelOWS aims to provide an infra-structure to register, research, recovery, compose and execute (simulate) biological models using ontologies. The combination of semantic description of components, in a standard format that allows their composition with other components, to models that can be simulated by existing tools, brings a great flexibility for the modeling proc-esses in e-science projects.

CelOWS is implemented as a web service. This allows the repositories distribution and facilitates its use in scientific workflows and grid environments. Each model is also encapsulated in a web service, so it is able to be executed remotely (independ-ently or composed with other components) or it can simply inform its localization where the local execution code can be found.

A general vision of the CelOWS architecture is presented in Fig. 4. The CelO URI represents the ontology model. The architecture considers three different tiers:

- *CelOWS*: Implemented as a web service, can be installed in different sites, allow-ing the distribution of the repositories, and is interface independent, as it facilitates the integration with existing tools.
- *Backend*: it is the services tier, used by the CelOWS to access the database (Ce-lOWS-DB) and the execution tools to simulate CellML models.
- *Client*: implements the user interface and it can be developed in any language with access to web services.

The architecture offers four services for its users: a) *Registry*: From the URI given by the user, the model is then stored in a database.; b) *Compose*: The user gives XML files with the specification of the models that must be composed and the composition architecture (how the components will be connected among themselves). A new model is generated and stored in the database; c) *Query*: The user makes a SPARQL query [13] and receives, as a result, the models, components or variable that attends it; d) *Execute*: The user gives the URI model and the parameters to be used (from a pre-vious query). From the given URI of the CellML model and a simulation tool is lo-cated, the computing is then executed and the results returned.

The CelOWS services are distributed in four layers: a) *Client Manager*: Responsi-ble for all users interaction, implementing the Facade project pattern. Its purpose is to supply only one input/output point in the system, so customers do not have access to the internal structure of the CelOWS. b) *Storage Manager*: Responsible for the stor-age/recovery processes of the ontology in the database, and queries carried out by the

user, encapsulating the access to the data base. c) *Ontology Manager*: Responsible for the inference on the models, as well as for providing an API (Application Program Interface) to access CelO ontology. d) *Execution Manager*: Responsible for the execution of the CellML model associated with the CelO model. The access to the simulation tools must be encapsulated, enabling the architecture to be independent of a specific tool.

The use of the complete CelOWS infrastructure depends on the development of end users applications, using graphical interfaces and encapsulating its services.

Prototype validation: To test and validate CelOWS functionalities a prototype was built, with the following services: register, query and models composition.

Eight CellML models were used for testing the Models Composition service. For example, the model "A Modification of the Hodgkin-Huxley Equations Applicable to Purkinje Fibre Action and Pace-Maker Potentials" is represented in Fig. 5 through a UML Components Diagram. The diagram presents the components and its connections through the interface parameters
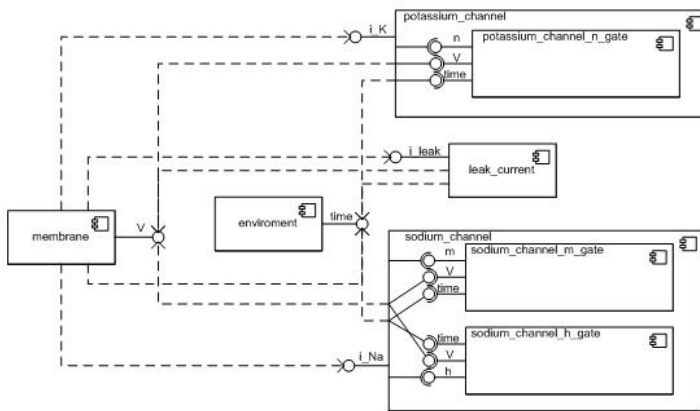

Fig. 5. Model used for testing the prototype

As it can be seen in the diagram, the model is composed of eight components: environment, membrane, potassium_channel_n_gate, potassium_channel, sodium_channel, leak_current, sodium_channel_m_gate, sodium_channel_h_gate. During the tests, the model was divided, creating eight CellML files (atomic models) and enabling eight CelO models. The original files, the membrane atomic model and the CelO model of the membrane are available in the site (http://celo.mmc.ufjf.br).

One of the specific goals of CelOWS is the composition of atomic models. The idea is to promote the reuse of existing components, through the composition of simpler models, producing more complex ones. The composition produces a new CellML model, through the copy of previously defined components and the automatic connection of these components.

The composition process uses a XML configuration file to indicate which models will be composed and the structure of the new model. The connection of the components is made through the semantic combination of the parameters of the first component with the parameters of the second one. It is established if both parameters measure the same quantities, if both are associated to the same chemical element and, finally, if both have the same name. A SPARQL query is used to find if the output parameters are compatible to the input of the other component. As the composition process is made through the copy of the existing models, it can be considered static (the modification of a used model in the composition process does not modify the composed component).

In order to validate the CellML model, generated from the composition process, the PCEnv program version 0.3, for models simulation, was used. Graph V x Times representing the potential action was generated from the original CellML "the Modification of the Hodgkin-Huxley Equations Applicable you the Purkinje Fibre Action and Pace-Maker Potentials" (Fig. 6) and with the file generated through the composition and connection of components (Fig. 7). These graphs (as others generated during the tests) are identical, proving the validation of the composition process.
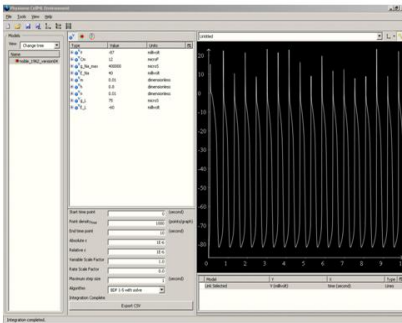
Fig. 6. Graph V x time generated with original CellML model
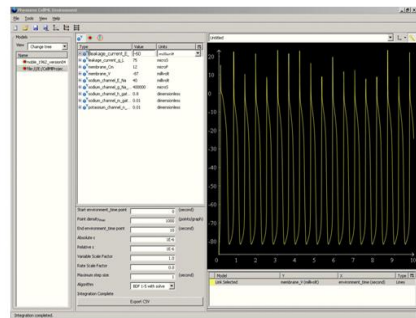
Fig. 7. Graph V x time generated with CellML model from a CelO model

To test and validate the semantic query functionalities in CelO models, some SPARQL queries (Fig. 8) were defined to locate variables or components that could be used in the composition process of a new model.

Implementation details: All the CelO ontology manipulation, as well as the described models in OWL, are in Protégé-OWL 3,4 (Editor and API) [14]. For inference the Pellet Reasoner [15] and the Jess Rule Engine [16] were adopted. For the development of the CelOWS we used the Eclipse, Java and PHP5 languages. For the ontology storage in a relational database and the SPARQL queries, the SOR (Scalable Ontology Repository) [17] was used.

```
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 all variables associated to
PREFIX celo: <http://celo.mmc.ufjf.br/ontologies/celo.owl#>
select ?x WHERE {
?x a celo:ActionPotential }
List components which variables are associated to Potencial Action
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 }
Show the components with associatd parameters to the electric chain measurements
and sodium chemical element.
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. SPARQL queries in ontological repository

5   Related Work

MONET Project [18] investigates the area of mathematical web services, with implementation of a broker, definition of the description services language MSDL (Mathematical Service Description Language), many domain ontologies and the component InstanceStore [19]. The CelOWS architecture is similar to the Monet architecture even though Monet does not use semantic web services. Project GENSS (Grid-Enabled Numerical and Symbolic Services) [20] is an extension of Monet project. It deals with the combination of grid computing and mathematician web services using an open framework based in software agents. One of CelOWS implementation goal as a web service is to allow its use in workflow environments for services composition and its use in grid computing. In this way some of the GENSS project proposals could be applied to our research. An association of ontologies and web services to support the biological systems modelling is described in [21]. An ontology to represent the models in OWL is presented. OWL-S is used to specify the parameterization and semi-automatic composition of web services of the model execution. CelOWS architecture has similar goals, using, however, a more open domain ontology, the biological one. However, OWL is not used to represent the complete model, but to describe semantics of biological models in CellML

6   Concluding Remarks

The increasing volume and distribution of data and processes in Bioinformatics speed up the discovery of new biological information. To manage these data and processes in an automatic and scalable form, the use of scientific workflows is now essential. On the other hand, the biological models have different possible representations, such as conceptual, mathematical and computational. Although it is desirable the

association of the workflows and modeling areas, the literature does not present specific proposals in this direction. We believe that this happens because generally the models are considered only in their "representation" aspect, while workflows deal with software components that can be "executed". Our research considers the approach of these areas using innovative technologies, based on well established standards. In the aspect of "representation" Cell Component Ontology - CelO is presented, to describe the semantics to the biological models. In the aspect of "process", the CelOWS architecture was developed to storage, query, reuse, compose and execute these models. CelOWS follows the service oriented architecture, being itself implemented as a web service. Some questions as scalability and performance have not been treated yet. Future work will focus on these questions, as well as the integration with other ontologies of the biological domain.

References

1. Garny, A., et al.: CellML and Associated Tools and Techniques. Elsevier Ireland Ltd. (2007)
2. Cell Markup Language, `http://www.cellml.org`
3. Bechhofer, S., et al.: OWL web ontology language 1.0 reference, `http://www.w3.org/TR/owl-ref/`
4. Nickerson, D., Hunter, P.: The Noble cardiac ventricular electrophysiology models in CellML. Prog. Biophys. Mol. Biol. 90(1-3), 346–359 (2006)
5. Luo, C., Rudy, Y.: A Dynamic Model of the Cardiac Ventricular Action Potential - Simulations of Ionic Currents and Concentration Changes. Circulation Research 74, 1071–1097 (1994)
6. Gruber, T.R.: Towards principles for the design of ontologies used for knowledge sharing. In: Guarino, N., Poli, R. (eds.) Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer, Dordrecht (1994)
7. Borst, W.N.: Construction of Engineering Ontologies. Phd Thesis (1997), `http://www.ub.utwente.nl/webdocs/inf/1/t0000004.pdf`
8. Barbosa, C., Santos, R., Amorim, R., et al.: A Transformation Tool for ODE based models. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2006. LNCS, vol. 3991, pp. 69–75. Springer, Heidelberg (2006)
9. Physiome CellML Environment, `http://www.cellml.org/downloads/pcenv`
10. Fernández-López, M.: Overview of Methodologies For Building Ontologies. In: Proceedings of the IJCAI 1999 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden (1999), `http://www.lsi.upc.edu/~bejar/aia/aia-web/4-fernandez.pdf`
11. Martin, D., et al.: OWL-S: Semantic Markup for Web Services. In: W3C Member Submission November 22 (2004), `http://www.w3.org/Submission/OWL-S/`
12. Horrocks, I., et al.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. In: W3C Member Submission, May 21 (2004), `http://www.w3.org/Submission/SWRL/`
13. SPARQL Query Language for RDF. W3C Recommendation, January 15 (2008), `http://www.w3.org/TR/rdf-sparql-query/`
14. Protégé-OWL – Ontology Editor for Semantic Web, `http://protege.stanford.edu/plugins/owl/`

15. Pellet: The Open Source OWL DL Reasoner, `http://pellet.owldl.com`
16. JESS, the Rule Engine for the Java Platform, `http://herzberg.ca.sandia.gov`
17. Lu, J., et al.: SOR: a practical system for ontology storage, reasoning and search. In: Proceedings of the 33rd international Conference on Very Large Data Bases, Vienna, Austria, September 23 - 27, 2007, Very Large Data Bases. VLDB Endowment, pp. 1402–1405 (2007)
18. MONET Consortium, `http://monet.nag.co.uk`
19. Bechhofer, S., Horrocks, I., Turi, D.: The OWL Instance Store: System Description. In: Nieuwenhuis, R. (ed.) CADE 2005. LNCS, vol. 3632, pp. 177–181. Springer, Heidelberg (2005)
20. GENSS, `http://genss.cs.bath.ac.uk/project.htm`
21. Sun, Z., Finkelstein, A., Ashimore, J.: Using Ontology with Semantic Web Services to Support Modeling in Systems Biology. In: International Workshop on Approaches and Architectures for Web Data Integration and Mining in Life Sciences (WebDIM4LS), Nancy, France (2007)