

Composição de modelos para a fisiologia: uma proposta de infraestrutura de e-science baseada em ontologias

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

¹ Mestrado em Modelagem Computacional

² Núcleo de Pesquisa em Qualidade de Software

³ Laboratório de Fisiologia Computacional

Campus Universitário – Universidade Federal de Juiz de Fora

Juiz de Fora – Minas Gerais – Brasil

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

Abstract: *The amount of information generated by research in biological sciences has lead to an intensive use of models. Mathematical and computational modeling need accurate description to share and simulate models as formulated by original authors. CellML is a markup language that uses XML to describe such models in the context of biological process. It allows a raw description of variables, equations and components, but it is insufficient to express semantically the model. Our research focus on the use of ontologies to enrich the construction of new cell models. In this paper we introduce the Cell Component Ontology - CelO, an ontology expressed in OWL-DL. This ontology captures both the structure of a cell model and the properties of functional components. We are using this ontology in a Web project where the use of CelO allows describing, reasoning, querying and composing CellML models. It aims to improve reuse and composition of existent components and allow semantic validation of new models.*

Resumo: *A quantidade e variedade de informações geradas pelas ciências biológicas têm levado a um uso intenso de modelos na área. Neste contexto, tanto a modelagem matemática quanto a modelagem computacional necessitam de uma descrição acurada que permita que esses modelos sejam compartilhados e simulados como foram formulados por seus criadores. CellML é uma linguagem de marcação que permite a descrição de variáveis, equações e componentes de modelos biológicos, mas é insuficiente para representar a semântica de tais modelos. Nossa pesquisa busca expressar de maneira formal a semântica desses modelos, através do uso de ontologias. Neste artigo, apresentamos a Cell Component Ontology – CelO, uma ontologia expressa em OWL-DL, que captura tanto a estrutura do modelo de uma célula quanto as propriedades dos seus componentes funcionais. A ontologia CelO é utilizada em uma arquitetura orientada a serviços para descrever, realizar inferências e consultar os modelos CellML. O objetivo principal da arquitetura é facilitar o reuso e a composição dos componentes existentes e permitir a validação semântica de novos modelos.*

1. Introdução

Pesquisas intensas em ciências biológicas têm gerado grande volume de dados. Ferramentas, métodos e técnicas possibilitam um melhor entendimento das várias funções, estruturas e processos relacionados à biofísica e à fisiologia. O aumento do poder computacional e a utilização de métodos numéricos estimulam o desenvolvimento e o uso de modelos mais complexos [Garny 2007]. Esses modelos permitem que dados de diferentes experimentos físicos e em diferentes escalas sejam combinados em uma simulação computacional, provendo uma visão mais acurada dos fenômenos estudados.

A simulação computacional de um modelo envolve duas questões importantes. A primeira se refere à representação do modelo. Embora diagramas, descrição textual e equações possam ser usadas na publicação dos modelos, elas estão sujeitas não só a erros tipográficos como à falta de definição das condições necessárias à simulação. A segunda questão está relacionada à implementação. A necessidade de aplicar métodos numéricos avançados atua como fator limitante para o efetivo uso e estudo do modelo.

Estas questões incentivaram o desenvolvimento da linguagem CellML [CellML 2007], uma linguagem de marcação criada especificamente para representação de modelos biológicos. Baseada na linguagem XML (*eXtensible Markup Language*), CellML especifica elementos que podem ser usados para representar um modelo de maneira formal, sem ambigüidades, legível por humanos e processável por máquina. As equações matemáticas são representadas em MathML (*Mathematical Markup Language*), o que as torna independentes de uma implementação específica.

Um modelo CellML é formado por componentes. Um componente é a menor unidade funcional no modelo. Geralmente o desenvolvimento de um novo modelo CellML usa componentes de um modelo existente como ponto de partida. Um novo componente é inserido no modelo tomado como base e o modelo completo é ajustado, através do estabelecimento de conexões com o novo componente e da definição de novas variáveis. O modelo é simulado várias vezes e, dependendo dos resultados obtidos, o novo componente é alterado ou substituído. Este processo pode se repetir várias vezes. Como em CellML as variáveis são locais aos componentes, a cada troca ou definição de componentes as variáveis devem ser redefinidas e as conexões refeitas. Isto torna o processo de modelagem trabalhoso e sujeito a erros.

Outra questão importante está relacionada à validação do modelo CellML. Como a linguagem é baseada em XML, a validação do modelo é eminentemente sintática. Através do uso de DTD (*Document Type Definition*) ou de XML Schemas o modelo pode ser validado quanto a erros de sintaxe e em relação à aderência à especificação CellML. Questões semânticas ou não podem ser efetivamente tratadas (e.g. evitar que um componente “canal iônico” contenha um componente “membrana”) ou são deixadas para a fase de implementação (e.g. a conformidade no uso de diferentes unidades de medida para as variáveis, como “segundos” e “milisegundos”).

Para tratar estas questões, este trabalho usa ontologias e regras semânticas para criação, validação, armazenamento e compartilhamento de modelos biológicos. Uma ontologia denominada Cell Component Ontology – CelO, descrita em OWL-DL [Bechhofer 2004], é apresentada. A ontologia permite separar o conhecimento associado ao domínio da Biologia do conhecimento associado à execução do modelo. As regras são usadas para inferir novos conhecimentos a partir da descrição OWL.

O objetivo da ontologia CelO é acrescentar semântica aos modelos. Isto permite expressar o conhecimento intrínseco no modelo, melhorar sua validação, reusar efetivamente componentes de outros modelos, automatizar alguns processos de composição e construir repositórios de modelos em que as pesquisas sejam realizadas semanticamente. A definição da ontologia manteve algum grau de compatibilidade com CellML, considerando a aplicabilidade dessa linguagem para a descrição e simulação de modelos e a existência de centenas de modelos CellML disponíveis para uso.

Para validar o uso da ontologia e verificar sua real aplicabilidade, um framework orientado a serviços, denominado CelOWS, foi desenvolvido. CelOWS permite o armazenamento, a pesquisa, a composição e a execução de modelos descritos usando a ontologia CelO. O principal objetivo do framework é facilitar a tarefa de composição de um novo modelo a partir de modelos armazenados em um repositório.

Na seção 2 são revistos alguns conceitos associados a ontologias, ao sub-domínio da eletrofisiologia da célula cardíaca e à linguagem CellML. A seção 3 apresenta a ontologia CelO, com a descrição dos principais elementos. A arquitetura do protótipo CelOWS e cenários de uso são discutidos na seção 4. A seção 5 descreve alguns desafios e perspectivas para a pesquisa na área. Alguns trabalhos relacionados são apresentados na seção 6 e a seção 7 traz as considerações finais.

2. Conceitos Relacionados

2.1. Ontologia

Uma ontologia pode ser descrita como sendo uma especificação formal e explícita de uma conceitualização compartilhada [Gruber 1994],[Borst 1997]. Nessa definição, “formal” significa que a especificação deve seguir algum formalismo (por exemplo, a lógica de predicados); “explícita” significa que conceitos, propriedades, relações, restrições e axiomas são explicitamente definidos; “compartilhado” quer dizer conhecimento consensual e “conceitualização” se refere a um modelo abstrato de algum fenômeno do mundo real.

A construção de uma ontologia implica em adquirir o conhecimento do domínio sendo tratado e coletar as informações apropriadas que definam, com consistência, os termos usados formalmente para descrever tal domínio. A ontologia proposta neste trabalho pode ser estendida para um domínio mais amplo dentro da área de Biologia, mas o escopo inicial foi restrito ao sub-domínio da eletrofisiologia da célula cardíaca. Esta restrição visa diminuir o nível de complexidade e permitir uma avaliação gradativa da ontologia, tanto em termos de completude quanto em termos de funcionalidade.

Uma ontologia típica é composta de uma taxonomia e um conjunto de regras de inferência [Berners-Lee et al. 2001]. Uma taxonomia descreve classes de objetos e relacionamentos entre estes. Se descrita de forma apropriada, preserva o significado específico dos termos e expressões em um dado domínio de conhecimento. A inferência permite capturar o conhecimento que não está explícito na taxonomia.

Existem diversas linguagens para representação de ontologias. Este trabalho adota OWL (*Ontology Web Language*) [Bechofer 2004]. Para a realização de inferências sobre os indivíduos da ontologia foi usada a SWRL (*Semantic Web Rule Language*) [Horrocks 2004]. SWRL possibilita complementar a ontologia com informações não diretamente processadas pela máquina de inferência.

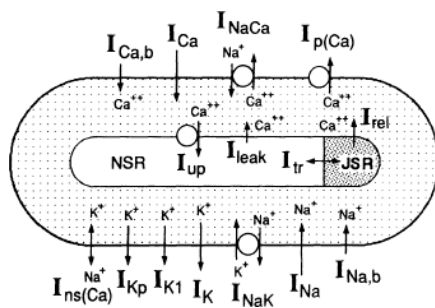
2.2. Eletrofisiologia da célula cardíaca

Os modelos em eletrofisiologia atualmente englobam desde modelos simples da ativação elétrica baseados em polinômios até modelos estocásticos complexos tridimensionais [Nickerson e Hunter 2006]. Esta seção, no entanto, apresenta apenas alguns conceitos básicos utilizados na ontologia proposta.

A célula é delimitada por uma membrana constituída de uma bi-camada fosfolipídica que controla o fluxo das substâncias que entram e saem de seu interior. A diferença na composição iônica dos meios intra- e extra-celular gera um gradiente eletroquímico entre os dois meios. O chamado **potencial de repouso** (*resting potential*) da membrana é a diferença de potencial entre o interior e o exterior da célula. A recepção de um estímulo altera essa diferença de potencial provocando fenômenos eletroquímicos com liberação de correntes iônicas, reduzindo o potencial elétrico até atingir um certo limiar. A partir daí é desencadeado um **potencial de ação** (*action potential*), um impulso elétrico transiente que percorre a membrana. A atividade elétrica do coração é consequência dos fenômenos elétricos gerados por estas variações de potencial [Malmivuo e Plonsey 1995].

Os diversos modelos existentes do comportamento elétrico das células podem ser vistos como modificações e refinamentos do modelo desenvolvido por Hodgkin e Huxley a partir de células nervosas [Nelson 2005]. Modelos mais específicos para células cardíacas foram propostos, tais como o modelo de Beeler-Reuter [Beeler e Reuter 1977] e os modelos de Luo-Rudy [Luo e Rudy 1991], [Luo e Rudy 1994].

Os componentes dos modelos de célula podem ser representados através de diagramas e possuem uma série de equações associadas. A Figura 1 apresenta o diagrama e uma parte das equações (extraídas do artigo original) do modelo Luo-Rudy II, um modelo do potencial de ação do ventrículo cardíaco [Luo e Rudy 1994]. Diagramas e descrições textuais, no entanto, estão sujeitas a erros tipográficos e à falta de definição das condições iniciais ou de contorno necessárias à simulação. Estas questões incentivaram o desenvolvimento da linguagem CellML para representação de modelos.



b. Currents through the L-type Ca^{2+} channel

$$I_{\text{Ca},i} = I_{\text{Ca}} + I_{\text{Ca},K} + I_{\text{Ca},\text{Na}}$$

$$I_{\text{Ca}} = d \cdot f \cdot f_{\text{Ca}} \cdot \bar{I}_{\text{Ca}}; \quad I_{\text{Ca},K} = d \cdot f \cdot f_{\text{Ca}} \cdot \bar{I}_{\text{Ca},K}; \quad \text{and} \quad I_{\text{Ca},\text{Na}} = d \cdot f \cdot f_{\text{Ca}} \cdot \bar{I}_{\text{Ca},\text{Na}}$$

For ion S, including Ca^{2+} , Na^{+} , and K^{+} ,

$$\bar{I}_S = P_S \cdot z_S^2 \cdot \frac{VF^2}{RT} \cdot \frac{\gamma_{Si} \cdot [S]_i \cdot \exp(z_S VF/RT) - \gamma_{So} \cdot [S]_o}{\exp(z_S VF/RT) - 1};$$

$$P_{\text{Ca}} = 5.4 \times 10^{-4} \text{ cm/s}; \quad \gamma_{\text{Ca},i} = 1; \quad \gamma_{\text{Ca},o} = 0.341;$$

$$P_{\text{Na}} = 6.75 \times 10^{-7} \text{ cm/s}; \quad \gamma_{\text{Na},i} = \gamma_{\text{Na},o} = 0.75;$$

$$P_K = 1.93 \times 10^{-7} \text{ cm/s}; \quad \gamma_{K,i} = \gamma_{K,o} = 0.75;$$

$$f_{\text{Ca}} = 1 / [1 + ([\text{Ca}^{2+}] / K_{m,\text{Ca}})^2]; \quad K_{m,\text{Ca}} = 0.6 \mu\text{mol/L};$$

$$d_s = 1 / [1 + \exp\{-(V + 10)/6.24\}];$$

$$\tau_d = d_s \cdot \{1 - \exp\{-(V + 10)/6.24\} / [0.035 \cdot (V + 10)]\};$$

$$f_s = 1 / [1 + \exp\{(V + 35.06)/8.6\}] + 0.6 / [1 + \exp\{(50 - V)/20\}];$$

$$\tau_f = 1 / \left(0.0197 \cdot \exp\{-[0.0337 \cdot (V + 10)]^2 + 0.02\} \right);$$

$$\alpha_d = d_s / \tau_d; \quad \beta_d = (1 - d_s) / \tau_d; \quad \alpha_f = f_s / \tau_f; \quad \text{and} \quad \beta_f = (1 - f_s) / \tau_f.$$

Figura 1. Diagrama em esquema e parte das equações do modelo LR-II [Luo e Rudy 1994]

2.3. CellML

CellML é uma linguagem de marcação, baseada em XML, que estabelece um formato padrão para definição e compartilhamento de modelos biológicos. Os modelos são representados como uma rede de componentes interconectados. Um modelo recebe um nome e um identificador único (cmeta:id) usado para referenciar o modelo através de uma URI (*Universal Resource Identifier*). Um componente é a menor unidade funcional de um modelo. Equações matemáticas expressas em MathML descrevem como cada componente se comporta no contexto do modelo.

Uma variável é uma entidade nomeada que pertence a um único componente. Seu propósito é representar quantidades usadas nas equações. Um valor inicial e alguns atributos (unidade, interfaces públicas e privadas, etc.) podem ser associados a uma variável. Conexões mapeiam variáveis entre componentes, permitindo a troca de informações entre eles. A especificação CellML [Cuellar 2006] requer que todas as variáveis e números sejam declarados conjuntamente com a unidade utilizada por eles. Esta característica garante a reusabilidade, já que componentes e modelos contendo variáveis com diferentes unidades, mas de mesma dimensão, podem ser conectados. É permitido ao usuário criar novas unidades a partir das unidades existentes.

Freqüentemente os componentes de um modelo são agrupados em uma estrutura hierárquica. Existem dois tipos de agrupamentos pré-definidos: encapsulamento e contenção. Encapsulamento é um tipo lógico de agrupamento que permite usar um componente como interface para um submodelo, impedindo o acesso direto aos componentes do submodelo. A contenção é usada para descrever a organização física ou geométrica do modelo, representando a estrutura dos componentes biológicos. CellML também permite a importação de componentes e unidades de modelos existentes.

3. Cell Component Ontology - CelO

O objetivo específico da definição da ontologia CelO é acrescentar semântica aos modelos. O uso da semântica para expressar o conhecimento intrínseco no modelo visa melhorar sua validação, reutilizar componentes definidos em outros modelos, automatizar alguns processos de composição com base nas características dos componentes e construir repositórios de modelos disponíveis na Web em que as pesquisas possam ser realizadas semanticamente. A definição da estrutura da ontologia foi influenciada pela perspectiva de sua integração com aplicações usadas para execução de modelos CellML, como AGOS [Barbosa et al. 2006] e PCEnv [PCEnv 2008], e, futuramente, em ambientes de workflow científicos.

A associação com CellML pode ser vista sob a perspectiva de uma arquitetura em três níveis de abstração, conforme a Figura 2. No caso dos modelos em eletrofisiologia, a ontologia CelO é utilizada para descrever variáveis e componentes em um nível mais alto de abstração. Por exemplo, um componente pode ser descrito como sendo um “canal iônico de sódio”. O modelo referencia um modelo CellML, onde as variáveis estão representadas sintaticamente, como nas linguagens de programação tradicionais. Um simulador que trabalhe com modelos CellML vai traduzir o modelo para uma implementação específica, a fim de que ele possa ser executado.

A integração da ontologia CelO com CellML permite, portanto, que sejam construídas ferramentas que usem os simuladores existentes, de forma que o

pesquisador modele em um nível alto de abstração e execute computacionalmente o modelo de forma transparente.

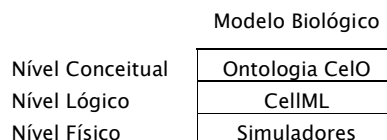


Figura 2. Níveis de abstração no processo de modelagem usando CelO.

3.1. Classes

A ontologia CelO apresenta três classes gerais no nível mais alto (Figura 3). Esta estrutura é motivada pela necessidade de prover três tipos essenciais de conhecimento sobre um modelo:

- **SIEntity**: Grandezas e unidades associadas às variáveis do modelo.
- **DomainEntity**: Conceitos do domínio da Biologia associados ao modelo.
- **ModelEntity**: Componentes do modelo e suas interfaces.

Neste texto, devido à limitação do espaço, apresentamos apenas as subclasses relacionadas a **ModelEntity**, por sua importância no contexto do framework CelOWS. A ontologia completa e os modelos usados nos testes podem ser acessados em <http://celo.mmc.ufjf.br>.

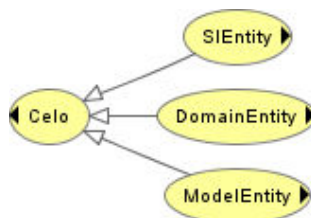


Figura 3. Nível topo da ontologia CelO.

A classe **ModelEntity** e suas sub-classes (Figura 4) definem os conceitos usados no modelo biológico representado. O objetivo é ter uma descrição de alto nível, fazendo referência ao modelo CellML para as questões de simulação.

Uma premissa na arquitetura do framework CelOWS é tratar um modelo como um serviço web, que possui interfaces e pode ser “executado” (através da simulação). Assim, a classe **ModelService** provê uma forma de organizar a visão de um modelo como um serviço. Sua estrutura é inspirada na ontologia OWL-S [Martin 2004], usada para descrição semântica de serviços web. Uma instância de **ModelService** existe para cada modelo e vai estar associada a uma instância de **ModelProfile**, **ModelGrounding** e **ModelProcess**.

ModelProfile informa sobre o que o modelo trata: quais são os componentes do modelo, se está associado a algum compartimento específico da célula e quais são as entidades biológicas (instâncias da classe **BiologicalEntity**) associadas ao modelo. **ModelGrounding** especifica qual o modelo “lógico” associado, no caso de uma simulação. A classe **CellMLModel** é usada para armazenar a URI do modelo CellML associado. **ModelProcess** indica como o modelo pode ser usado, ou seja, quais são os parâmetros de entrada (**ModelParameterIn**) e de saída (**ModelParameterOut**) que

estão associados à interface (**ModelInterface**) do modelo. Esses parâmetros estão diretamente associados às variáveis do modelo e podem ser usados no processo de simulação ou na composição com outros modelos. **ModelType** é uma classe genérica, usada para caracterizar o problema biológico tratado pelo modelo e o tipo de modelagem matemática utilizada.



Figura 4. Classe ModelEntity da ontologia CelO.

ModelObject agrupa os objetos que compõe um modelo, estando diretamente relacionado com o tipo de modelo subjacente. **Equation** relaciona as equações que implementam matematicamente o modelo, expressas em MathML. **Variable** representa as variáveis do modelo. **ModelVariable** descreve como as variáveis participam nas equações do modelo e **ComponentVariable** relaciona as variáveis de cada componente do modelo. Essas variáveis podem ser usadas na interface do componente (**InterfaceVariable**) ou serem locais (**LocalVariable**). **DomainVariable** associa as variáveis do modelo com conceitos expressos pelas classes de **DomainEntity**. **Model** representa o modelo propriamente dito, que pode ser atômico (contém um único componente) ou composto (contém dois ou mais componentes). **Component** representa os componentes do modelo, que podem estar descritos no próprio modelo (**InternalComponent**) ou podem fazer referência a modelos externos (**ExternalComponent**). Um modelo pode ser composto por modelos internos e externos simultaneamente.

3.2. Regras Semânticas

As regras semânticas são usadas na ontologia CelO com o propósito básico de inferir conhecimento que está implícito em modelos CellML existentes. Isto é feito definindo as propriedades associadas às variáveis dos componentes com indivíduos da ontologia, a partir de detalhes obtidos no modelo CellML, considerando também o conhecimento pré-definido (por exemplo, variáveis sem dimensão não podem estar associadas a

elementos químicos). Eventualmente as regras podem ser usadas também para validar semanticamente o modelo em relação à consistência e completude; um exemplo de validação é a verificação se o agrupamento dos componentes não fere a hierarquia anatômica.

A seguir apresentamos algumas regras definidas para a ontologia CelO, expressas em SWRL, com uma breve explicação.

Define a grandeza medida por uma variável:

```
celo:Variable(?x1) ^ celo:hasDetail(?x1, ?x2) ^ celo:DefinedUnit(?x3) ^
celo:hasName(?x3, ?x4) ^ swrlb:stringEqualIgnoreCase(?x2, ?x4) ^ celo:measures(?x3,
?x5) → celo:hasMeasure(?x1, ?x5)
```

Se uma variável x1 possui uma propriedade string detalhe x2 que é igual ao nome x4 de uma unidade x3 que mede uma grandeza x5, então a variável x1 mede a grandeza x5

Associa uma variável a um elemento químico (exceto as variáveis sem dimensão):

```
celo:Variable(?x1) ^ celo:hasDetail(?x1, ?x2) ^ celo:ChemicalEntity(?x3) ^
celo:hasName(?x3, ?x4) ^ swrlb:stringEqualIgnoreCase(?x2, ?x4) ^
celo:hasVariableUnit(?x1, ?x5) ^ celo:hasName(?x5, ?x6) ^ swrlb:notEqual(?x6,
"dimensionless") → celo:hasDomainEntity(?x1, ?x3)
```

Se uma variável x1 possui uma propriedade string detalhe x2 que é igual ao nome x4 de um elemento químico x3 e essa variável não é adimensional, ela é associada ao elemento químico através da propriedade hasDomainEntity

Associa uma variável a uma instância da classe BiologicalEntity:

```
celo:Variable(?x1) ^ celo:hasDetail(?x1, ?x2) ^ celo:BiologicalEntity(?x3) ^
celo:hasName(?x3, ?x4) ^ swrlb:stringEqualIgnoreCase(?x2, ?x4) →
celo:hasDomainEntity(?x1, ?x3)
```

Se uma variável x1 possui uma propriedade string detalhe x2 que é igual ao nome x4 de uma entidade biológica x3, então a variável x1 é associada à entidade biológica através da propriedade hasDomainEntity

3.3. Exemplo de um Modelo CelO

A seguir são apresentados fragmentos do modelo CelO referente ao componente “membrana” do modelo “A Modification of the Hodgkin-Huxley Equations Applicable to Purkinje Fibre Action and Pace-Maker Potentials” [CellML Repository 2008].

```
model:model_membrane
  rdf:type celo:Model , celo:ModelService ;
  celo:describedBy model: ModelInterface ;
  celo:hasName "membrane"^^xsd:string ;
  celo:presents model:_ModelProfile ;
  celo:refersTo model:_CellMLModel .
.....
model:_ModelInterface
  rdf:type celo:ModelInterface ;
  celo:hasInterface model:membrane_i_Na , model:membrane_V , model:membrane_i_Leak ,
model:membrane_i_K , model:membrane_time .
.....
model:_CellMLModel
  rdf:type celo:CellMLModel ;
  celo:hasURI "http://...ufjf.br/models/noble_1962/membrane.cellml"^^xsd:string .
.....
model:membrane_V
  rdf:type celo:ModelParameterOut , celo:ActionPotential , celo:DependentVariable ,
celo:InterfaceVariable ;
  celo:hasDetail "millivolt"^^xsd:string , "private"^^xsd:string ,
"membrane"^^xsd:string , "V"^^xsd:string , "-"^^xsd:string , "public_out"^^xsd:string ;
  celo:hasDomainEntity celo:Membrane ;
  celo:hasInitialValue "-87"^^xsd:string ;
  celo:hasMeasure celo:Voltage ;
  celo:hasName "V"^^xsd:string ;
  celo:hasPublicInterface "out"^^xsd:string ;
  celo:hasVariableUnit celo:millivolt ;
  celo:isInterfaceVariableOf model:membrane .
```


4. CelOWS: uma infraestrutura para aplicações em e-science

A especificação da ontologia CelO é um passo importante para a modelagem no domínio de fisiologia cardíaca. No entanto, a ontologia está inserida em um contexto mais amplo que é a especificação de uma infraestrutura para aplicações em e-science, para prover suporte computacional a projetos de pesquisa e pesquisadores que desejam compartilhar experiências e resultados. Esta infraestrutura visa apoiar a busca por artefatos de pesquisa relacionados a um dado domínio de aplicação.

Para validação, a ontologia CelO está sendo usada no framework CelOWS, desenvolvido para pesquisa e composição de modelos. O framework estende um trabalho anterior do grupo no contexto de infraestrutura para e-science, denominado MathWS [Matos et al. 2007]. CelOWS possibilita localizar semanticamente modelos em repositórios distribuídos, executar estes modelos como serviços web e realizar sua composição para gerar novos modelos. O tratamento de modelos como serviços que possuem uma descrição semântica e podem ser simulados traz uma grande flexibilidade para a tarefa de modelagem no ambiente de e-science. Cada modelo pode ser executado remotamente (de forma independente ou composto com outros modelos) ou pode, por exemplo, fornecer seu código (através de sua URI) para ser composto localmente.

4.1. Arquitetura

Uma visão geral da arquitetura do framework CelOWS é apresentada na Figura 5. Na figura *URI CelO* representa a URI do modelo CelO. O framework implementa conceitos de Arquitetura Orientada a Serviços e considera três camadas (*tiers*) distintas.

- **CelOWS:** representa o próprio framework. Implementado como um serviço web, o framework pode ser instalado em *sites* distintos, permitindo a distribuição dos repositórios, além de ser independente da interface que vai acessar os seus serviços, facilitando a integração com ferramentas existentes.
- **Backend:** camada de serviços usados pelo framework, como o acesso a banco de dados e a execução de simulações de modelos CellML.
- **Client:** implementa uma interface com o usuário do framework, podendo ser desenvolvida em qualquer linguagem com acesso a serviços web.

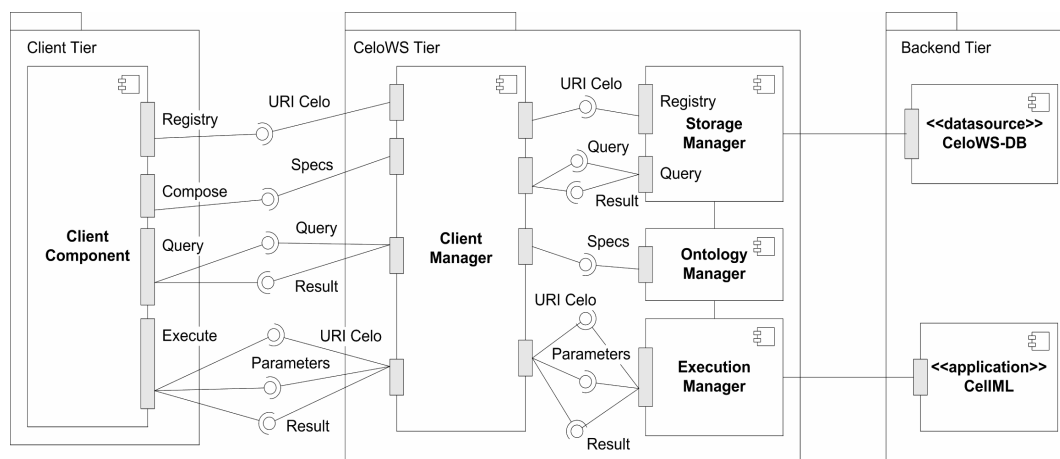


Figura 5. Visão geral da arquitetura do broker CelOWS.

4.2. Serviços

O framework oferece quatro serviços para seus usuários:

- **Registry:** Registro de um modelo no sistema. A partir da URI fornecida pelo usuário, o modelo é recuperado, um mecanismo de inferência é aplicado e o modelo é armazenado em um banco de dados.
- **Compose:** Composição de modelos. O usuário fornece um arquivo XML com a especificação de quais modelos devem ser compostos e a arquitetura dessa composição (como os componentes serão conectados entre si).
- **Query:** Pesquisa de modelos. O usuário fornece uma consulta codificada em SPARQL [SPARQL 2008] e recebe como resultado as URIs dos modelos, componentes ou variáveis, que atendem à consulta.
- **Execute:** Execução de modelos. O usuário fornece a URI do modelo a ser executado e os parâmetros a serem utilizados (os parâmetros para a execução podem ser obtidos por uma consulta SPARQL prévia). A partir do modelo é recuperada a URI do modelo CellML correspondente e invocada uma aplicação que faça a simulação do modelo, retornando os resultados obtidos.

Os serviços da camada CelOWS estão distribuídos por quatro módulos gerenciais:

- **Client Manager:** responsável por toda interação com as aplicações clientes do framework. Sua finalidade é impedir que os clientes tenham acesso à estrutura interna do framework, além de fornecer um ponto único de entrada/saída no sistema.
- **Storage Manager:** responsável pelos processos de armazenamento e recuperação de ontologias na base de dados, além do processamento das consultas (*queries*) realizadas pelo usuário. Encapsula o acesso ao banco de dados, permitindo que o framework seja independente do Sistema Gerenciador de Banco de Dados.
- **Ontology Manager:** responsável pelos processos de inferência sobre os modelos, bem como por encapsular a API (*Application Program Interface*) que acessa a ontologia CelO.
- **Execution Manager:** responsável pela execução do modelo CellML associado ao modelo CelO. Encapsula o acesso às aplicações de simulação, permitindo que o framework não dependa de uma aplicação específica.

4.3. Detalhes de Implementação

Toda a manipulação da ontologia CelO, bem como dos modelos descritos em OWL, é feita através do Protégé-OWL 3.4 (Editor e API) [Protege 2008]. Para inferência são usados o Pellet Reasoner [Pellet 2008] e o Jess Rule Engine [Jess 2008]. O desenvolvimento do framework CelOWS foi feito com uso do Eclipse e as linguagens Java e PHP5. Para o armazenamento das ontologias em um banco de dados relacional e a realização das consultas SPARQL é usado o SOR (*Scalable Ontology Repository*) [Lu et al. 2007]. A manipulação e execução dos modelos CellML é feita com AGOS e PCEnv.

Todos os serviços descritos estão implementados e funcionais. Em relação aos módulos gerenciais, apenas **Ontology Manager** está em desenvolvimento, sendo usada atualmente a API gerada automaticamente pelo Protégé para acesso à ontologia.

Como exemplo de uso, é apresentado o código XML usado para especificar a composição de modelos, gerando um novo modelo. Neste exemplo é feita a composição de componentes do modelo *“A Modification of the Hodgkin-Huxley Equations Applicable to Purkinje Fibre Action and Pace-Maker Potentials”* com componentes do modelo *“A quantitative description of membrane current and its application to conductance and excitation in nerve”*, disponíveis em [CellML Repository 2008].

```
<composition>
<global uri="http://celo.ufjf.br/owl/noble_1962/environment.owl"/>
<model uri="http://celo.ufjf.br/owl/noble_1962/membrane.owl" uses="global" >
  <model uri="http://celo.ufjf.br/owl/noble_1962/leakage_current.owl"uses="global"/>
  <model uri="http://celo.ufjf.br/owl/noble_1962/potassium_channel.owl" uses="global" >
    <model uri="http://celo.ufjf.br/owl/noble_1962/potassium_channel_n_gate.owl" />
  </model>
  <model uri="http://celo.ufjf.br/owl/noble_1962/sodium_channel.owl" uses="global" >
    <model uri="http://celo.ufjf.br/owl/hh_1952/sodium_channel_h_gate.owl" />
    <model uri="http://celo.ufjf.br/owl/hh_1952/sodium_channel_m_gate.owl" />
  </model>
</model>
</composition>
```

4.4. Cenários de Uso

Os seguintes cenários apresentam, de forma resumida, as possibilidades de utilização do framework CelOWS pelos pesquisadores:

- A partir de um conceito ou fenômeno biológico de interesse (por exemplo, “canal iônico de sódio” ou “potencial da membrana”), o pesquisador busca na ontologia CelO a sua representação semântica. De posse desta representação, ele pesquisa em um repositório quais modelos estão de alguma forma associados com o conceito e com o fenômeno. Obtendo uma lista de modelos candidatos, ele escolhe um que descreva o sistema desejado e pode executar simulações ou obter o código CellML para análise.
- O pesquisador pode obter um dado modelo CellML, executar simulações e, com base em sua experiência ou análise, enriquecer semanticamente o modelo CelO associado ao modelo CellML.
- O pesquisador pode localizar um modelo existente, possivelmente composto por componentes atômicos, e fazer simulações substituindo estes componentes por outros semelhantes. Os componentes semelhantes, tanto em termos de descrição de um sistema, quanto em relação aos parâmetros de entrada e saída, podem ser localizados através de uma busca semântica.
- O pesquisador pode construir um novo modelo, a partir de um modelo vazio ou de um modelo existente, através da composição de componentes atômicos existentes e descritos semanticamente. Um novo modelo CellML é gerado, permitindo sua simulação ou seu uso na composição de modelos mais complexos.

5. Desafios e Perspectivas para Pesquisa

Os desafios enfrentados na área de pesquisa relacionada à proposta apresentada podem ser classificados em dois grupos: Biologia Sistêmica e Ciência da Computação.

Biologia Sistêmica (*Systems Biology*) é uma disciplina emergente que envolve a integração do conhecimento biológico através de diversas escalas e domínios, a fim de compreender a dinâmica dos diversos processos biológicos que interagem entre si como um sistema integral [Finkelstein et al. 2004]. O painel apresentado em [Sauro et al. 2006] discute os principais desafios para modelagem e métodos de simulação em Biologia Sistêmica. Destacamos a questão da modelagem multi-escala, que envolve a integração de modelos genéticos e modelos de órgãos, incluindo modelos celulares e de tecidos. O uso de ontologias é fortemente considerado neste contexto pela possibilidade de representar o conhecimento dos diversos níveis de maneira integrada e formal.

Por outro lado, a SBC (Sociedade Brasileira de Computação) apresentou a modelagem computacional de sistemas complexos naturais como um dos Grandes Desafios da Pesquisa em Computação [SBC 2006]. A proposta apresentada está associada aos desafios específicos relacionados à modelagem computacional em pelo menos três pontos:

- Desenvolvimento de ferramentas para a construção colaborativa de modelos: implementado como um serviço web, o framework CelOWS permite o acesso por pesquisadores distribuídos geograficamente, que podem gerar modelos de forma colaborativa e compartilhar instantaneamente resultados de simulações, aumentando a interação nas pesquisas compartilhadas;
- Criação de novas técnicas em visualização científica: a representação semântica dos modelos permite, com o desenvolvimento e uso de ferramentas adequadas, uma representação gráfica e textual mais próxima do domínio do pesquisador, em um nível de abstração que oculte detalhes de implementação como, por exemplo, a sintaxe XML;
- Problemas de dimensionalidade e escalabilidade: o uso de uma arquitetura orientada a serviços, como proposto, associado à execução em um ambiente de grade computacional, acessível via web, permite que modelos mais complexos sejam experimentados por um número maior de pesquisadores, favorecendo a pesquisa.

As principais questões a serem tratadas a longo prazo se referem à integração das diversas ontologias existentes e à representação dos modelos. Em Biologia existem atualmente dezenas de ontologias, englobando as áreas Biomédica, Celular e Genética. Estas ontologias, no entanto, estão associadas a domínios específicos e não ao processo de modelagem. Linguagens para modelagem, como CellML e SBML [SBML 2008], ainda estão em um nível mais baixo de abstração. Um grande esforço de pesquisa e padronização é necessário para que os níveis biológico, matemático e computacional de um modelo possam ser explorados de maneira eficaz.

6. Trabalhos Correlatos

A discussão sobre o uso de ontologias em CellML vem ocorrendo na comunidade de CellML há alguns anos [Nielsen 2003]. Durante o CellML Workshop 2007 o tema foi

novamente discutido [Halstead 2007]. A ontologia CelO é uma contribuição para esta discussão por representar os modelos integrando linguagens de ontologias e CellML.

Um trabalho com ontologias aplicadas aos modelos de célula é descrito em [Shimayoshi 2006]. Os autores apresentam métodos para suporte ao desenvolvimento de estruturas de modelos complexos das células, usando uma linguagem de marcação (PMSML) e uma ontologia chamada Cell Model Ontology. A ontologia CelO tem um objetivo similar, mas usa linguagens mais padronizadas, como OWL e SWRL. Isto permite o uso de uma grande variedade de ferramentas, além da possibilidade de um trabalho colaborativo no desenvolvimento da ontologia.

Uma abordagem usando a associação de ontologias e serviços web para apoiar a modelagem em sistemas biológicos é descrita em [Sun et al. 2007]. É apresentada uma ontologia para representar o meta-modelo em OWL. OWL-S é usado para especificar a parametrização e composição semi-automática de serviços web voltados para a execução do modelo. Este trabalho é similar em finalidade ao framework CelOWS. A ontologia CelO, no entanto, abrange mais que o meta-modelo, incluindo conceitos do domínio biológico em estudo.

7. Considerações Finais

O crescente volume e a distribuição de dados e processos em Bioinformática tornam cada vez mais fácil a descoberta de novas informações biológicas. Para gerenciar esses dados e processos de forma automatizada e escalável, o uso de workflows científicos torna-se imprescindível. Por outro lado, os modelos desenvolvidos na área da Biologia possuem diversas representações possíveis, das quais destacam-se as representações conceitual, matemática e computacional.

Embora seja desejável a associação das áreas de workflows e modelagem, a literatura não apresenta propostas específicas nessa direção. Acreditamos que isto acontece porque geralmente os modelos são considerados apenas no seu aspecto de “representações”, enquanto workflows lidam com componentes de software que podem ser “executados”.

Nossa pesquisa propõe a aproximação destas áreas usando mecanismos inovadores, embora baseados em padrões bem estabelecidos. No aspecto de “representação” é apresentada a ontologia Cell Component Ontology – CelO, para acrescentar semântica aos modelos biológicos. No aspecto de “processo”, foi desenvolvido o framework CelOWS para armazenamento, pesquisa, composição e execução desses modelos. O framework segue a abordagem de arquitetura orientada a serviços, sendo implementado ele próprio como um serviço web.

Algumas questões como escalabilidade e desempenho ainda não foram tratadas. Trabalhos futuros terão como foco estas questões, assim como preocupações relacionadas à integração com outras ontologias do domínio da biologia.

Referências

- Barbosa, C. et al. (2006) “A Transformation Tool for ODE based models”. Lecture Notes in Computer Science 3991, pp. 69–75.
- Bechhofer, S. et al. (2004) “OWL web ontology language 1.0 reference”. Disponível em: <http://www.w3.org/TR/owl-ref/>.

- Beeler, G. W., Reuter, H. (1977) “Reconstruction of the action potential of ventricular myocardial fibres”. *Journal of Physiology* , 268, 177-210.
- Berners-Lee, T. et al. (2001) “The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”. Scientific American.
- Borst, W.N. (1997) “Construction of Engineering Ontologies”. Phd Thesis. Disponível em: <http://www.ub.utwente.nl/webdocs/inf/1/t0000004.pdf>.
- CellML (2008). “Cell Markup Language”. Disponível em <http://www.cellml.org>.
- CellML Repository (2008) “Cell Markup Language”. Disponível em <http://www.cellml.org>.
- Cuellar, A. A. et al. (2006) “CellML Specification 1.1 ”. Disponível em: http://www.cellml.org/specifications/cellml_1.1.
- Finkelstein, A. et al. (2004) “Computational Challenges of Systems Biology” IEEE Computer, Vol. 37 No. 5 (2004) 26–33.
- Garny A. et al. (2007) “CellML and Associated Tools and Techniques”. Paper submitted to Elsevier Ireland Ltd.
- Gruber, T. R. (1994) “Towards principles for the design of ontologies used for knowledge sharing”. In N. Guarino and R. Poli (Eds.), Formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer.
- Halstead M. (2007) “Ontologies and Repositories”. CellML Workshop 2007. Disponível em : <http://www.cellml.org/workshop2007>.
- Horrocks, I. et al. (2004) “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”. W3C Member Submission 21 May 2004. <http://www.w3.org/Submission/SWRL/>.
- Jess (2008) “JESS, the Rule Engine for the Java Platform”. Disponível em: <http://herzberg.ca.sandia.gov>.
- Lu, J. et al. (2007) “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, 1402-1405.
- Luo, C., Rudy, Y. (1991) “A Model of the Ventricular Cardiac Action Potential - Depolarisation, Repolarisation and Their Interaction”, *Circulation Research* , 68, 1501-1526.
- Luo, C., Rudy, Y. (1994) “A Dynamic Model of the Cardiac Ventricular Action Potential - Simulations of Ionic Currents and Concentration Changes”, *Circulation Research* , 74, 1071-1097.
- Malmivuo, J., Plonsey, R. (1995) “Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields”. Oxford University Press, New York.
- Martin, D. et al. (2004) “OWL-S: Semantic Markup for Web Services”. W3C Member Submission 22 November 2004. Disponível em: <http://www.w3.org/Submission/OWL-S/>.

- Matos, E. et al. (2007) “MathWS: Broker de Serviços Web para e-Science”. 1st Brazilian e-Science WorkShop. in conjunction with 22nd Brazilian Symposium on Data Base, João Pessoa, Brazil.
- Nelson, M. E. (2005) “Electrophysiological Models” In: Databasing the Brain: From Data to Knowledge. S. Koslow and S. Subramaniam (eds.) pp. 285-301, Wiley, New York.
- Nickerson D, Hunter P. (2006) “The Noble cardiac ventricular electrophysiology models in CellML”. Prog Biophys Mol Biol, Jan-Apr;90(1-3):346-59.
- Nielsen P. et al. (2003) “Ontologies in CellML: A Versatile Method to Describe Cellular Models”. IIth International conference on Intelligent Systems for Molecular Biology (ISMB). International Society for Computational Biology(ISCB). (2003)
- PCEnv (2008) “Physiome CellML Environment”. Disponível em: <http://www.cellml.org/downloads/pcenv>.
- Pellet (2008) “Pellet: The Open Source OWL DL Reasoner”. Disponível em: <http://pellet.owldl.com>.
- Protégé (2008) “Protege OWL – Ontology Editor for Semantic Web”. Disponível em: <http://protege.stanford.edu/plugins/owl/>.
- Sauro, H. M. et al. (2006) “Challenges for modeling and simulation methods in systems biology”. In Proceedings of the 38th Conference on Winter Simulation (Monterey, California, December 03 - 06, 2006). L. F. Perrone, B. G. Lawson, J. Liu, and F. P. Wieland, Eds. Winter Simulation Conference. Winter Simulation Conference, 1720-1730.
- SBC (2006) “Grandes Desafios de Pesquisa em Computação no Brasil: 2006 – 2016 – Relatório Final”. Disponível em: <http://www.sbc.org.br/index.php?language=1&content=downloads&id=231>.
- SBML (2008) “Systems Biology Markup Language”. Disponível em: http://sbml.org/Main_Page.
- Shimayoshi, T. et al. (2006) “A Method to Support Cell Physiological Modelling Using Description Language and Ontology”. IPSJ Digital Courier, 2:726-735, 2006.
- SPARQL. (2008) “SPARQL Query Language for RDF”. W3C Recommendation 15 January 2008. Disponível em: <http://www.w3.org/TR/rdf-sparql-query/>
- Sun, Z. et al. (2007) “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.