Original Paper

A Method to Support Cell Physiological Modelling Using Description Language and Ontology

Takao Shimayoshi,[†] Kazuhiro Komurasaki,[††] Akira Amano,[††]
Takeshi Iwashita,[†††] Tetsuya Matsuda[††]
and Masanori Kanazawa[†††]

The development of physiological cell models to support the understanding of biological mechanisms gains increasingly importance. Due to the complexity of biological systems, whole cell models, which are composed of many imported component models of functional elements, get quite complex, making modifications difficult. Here, we propose a method to enhance structural changes of cell models, employing the markup languages of CellML and our original PMSML (Physiological Model Structure Markup Language), in addition to a new ontology for cell physiological modelling, the Cell Model Ontology. In particular, a method to make references from CellML files to the ontology and a method to assist with manipulation of model structures using PMSML together with the Cell Model Ontology are reported. Using these methods two software utilities, an interactive ontology ID assigner, the CellML Ontologizer, and a graphical cell model editor, the Cell Structure Editor, are implemented. Experimental results proved that the proposed method and the implemented software are useful for the modification of physiological models.

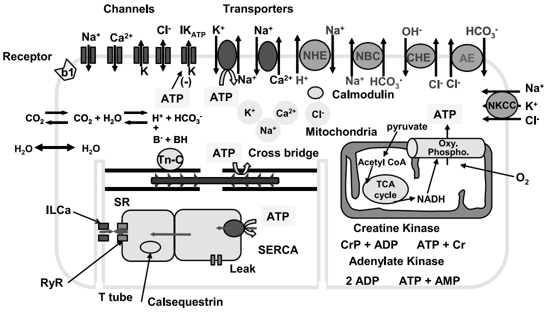
1. Introduction

If the mechanisms of life are to be elucidated, it is necessary to understand the functions of a single cell, the elementary unit of biological activities. In cell physiology, functions of a whole cell are explained as interactions of multiple functional elements, which are sub-functions of cellular functions, e.g., functional protein complexes such as membrane transporters and enzymes, and abstract cellular functions such as background currents and ATP (adenosine triphosphate) production. To represent the dynamic behaviour of each functional element, many models have already been proposed and more are being developed in physiology, especially in the field of electrophysiology. These models are generally formulated as a series of ordinary differential equations.

For the representation of cellular functions, several whole cell models[1)~4)] have recently been developed. As the structure of a cellular function, whole cell models are composed of multiple component models of functional elements. An ordinary differential system of a whole cell model is constructed by connecting equations of component models with global variables such as ion concentrations and the membrane potential. **Figure 1** and **Fig. 2** show schematic diagrams of whole cell models, Kyoto and LRd Model respectively. Model schemas are generally illustrated as diagrams with functional elements arranged in an anatomical hierarchy, which is important for the recognition of the model structure. In respect to a mathematical structure, a whole cell model can be represented as a graph structure of model components, where a model component is defined as either a component model of a functional element or a global variable. **Figure 3** shows such a diagram for the Kyoto Model with model components as nodes and their reference relations as paths.

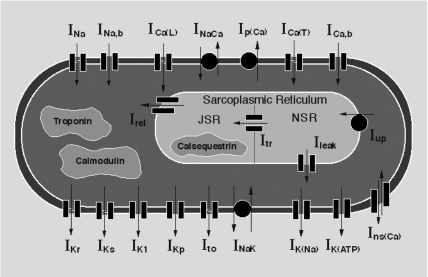
In general, whole cell models are developed using the following procedures. First, an existing model of a functional element is inserted to the whole cell model or updated from the old component model in the whole cell model. The whole cell model is then adjusted to accommodate the new component model. For the insertion or replacement of a component model, The global variables lacking have to be added and new connections have to be made to existing variables if they are referred from the component model. In the next step, the behaviour of the whole cell model has to be validated by several simulations. Depending on the simulation results, the component model

[†] ASTEM Research Institute of Kyoto
[††] Graduate School of Informatics, Kyoto University
[†††] Academic Center for Computing and Media Studies, Kyoto University



from http://www.biosim.med.kyoto-u.ac.jp/research/

Fig. 1 Schematic diagram of Kyoto Model.

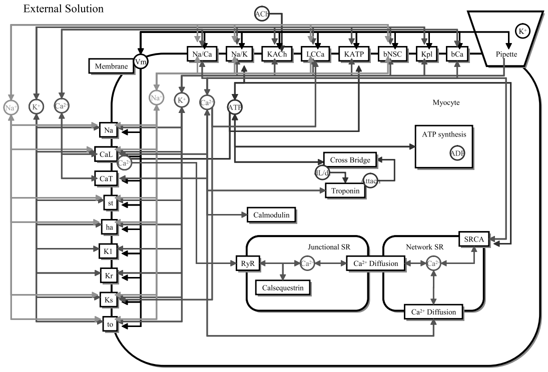


from http://www.cwru.edu/med/CBRTC/LRdOnline/

Fig. 2 Schematic diagram of LRd Model.

is improved or exchanged. Since it is difficult for many physiologists to modify complicated models on a computer, several kinds of software are presently available to support the development of physiological models. However, the interfaces of many existing editors are not intuitive for physiologists. For example, when using COR[5], CESE[6] or simBio[7], programming skills are necessary for the modelling. The CellML Editor[8] treats only the mathematical aspects, despite the fact that the information about the physiological structure of a model is very important for model developers. The Virtual Cell[9] requires extensive manual operations for modifications of model structures when multiple component models are imported.

Further, for the electronic representation of physiological models, an XML-based description language, CellML[10],[11] was proposed. A CellML file includes a definition and bibliographic metadata of a model. CellML is a useful representation technique for an editor to process and store models. In fact, several software packages[5],[6],[9],[12] in the cell modelling field support CellML. Moreover, a model repository described by CellML has been constructed. However, since the original purpose of CellML was the description of published models, it is not well suitable for addition and replacement of model elements. In CellML, detailed definitions of functional elements and their relations are declared in the same file, and the information about the model structure is described in different sections. Therefore, a careful modification of multiple parts is required to ensure consistency if component models are exchanged.



from http://www.biosim.med.kyoto-u.ac.jp/research/

Fig. 3 Graph structure of Kyoto Model.

To resolve these problems, this paper reports a method to support the development of complex cell physiological models. An original XML-based language for description of the structure of cell models, one that is suitable for the insertion and replacement of model components, and a novel ontology to represent the knowledge of cell models are proposed. As well, for integration of multiple models, a method to assist the identification of model components with the ontology and to aid the modification of cell model structures using the description language and the ontology is introduced.

2. Materials and Methods

2.1 Physiological Model Structure Markup Language

To improve CellML for the usage of model modifications, a description language PMSML (Physiological Model Structure Markup Language) is employed as a supplement to CellML. In a PMSML file, the relations of model components and the anatomical hierarchy of a model are clearly described with separation from the details of the model defined in the referred CellML files.

In a CellML file, a definition of a functional element in a physiological model is described as a '`component`' element, which has a '`name`' attribute as a unique identifier in the file. A '`component`' element contains '`variable`' ele-

```
<?xml version="1.0" encoding="iso-8859-1"?>
<model name="sample" cmeta:id="sample" xmlns="http://w
ww.cellml.org/cellml/1.0#" xmlns:cellml="http://www.ce
llml.org/cellml/1.0#" xmlns:cmeta="http://www.cellml.o
rg/metadata/1.0#">

...

  <!-- Definition of a component-->
  <component name="sodium_channel">

    <!-- Definition of variables -->
    <variable name="i_Na" public_interface="out" units
="microA_per_cm2" />
    <variable name="g_Na" initial_value="120.0" units=
"milliS_per_cm2" />
    <variable name="E_Na" units="millivolt" />
    <variable name="time" public_interface="in" privat
e_interface="out" units="millisecond" />
    <variable name="V" public_interface="in" private_i
nterface="out" units="millivolt" />
    <variable name="E_R" public_interface="in" units="
millivolt" />

    <!-- Definition of mathematical equations -->
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <apply id="E_Na_calculation"><eq />
        <ci> E_Na </ci>
        <apply><plus />
          <ci> E_R </ci>
          <cn cellml:units="millivolt"> 115.0 </cn>
        </apply>
      </apply>

 ...

  <connection>
    <map_components component_1="sodium_channel" compo
nent_2="environment" />
    <map_variables variable_1="time" variable_2="time"
 />
  </connection>

 ...
```

Fig. 4 An example of a CellML file.

ments, which define variables used in the mathematical expressions of the functional element. A 'variable' element has 'name' and 'units' attributes. A 'public_interface' attribute has a value of "in", if the variable refers to a variable of another component, or a value of "out", if the variable is referred from other components. In this paper, a 'variable' element with the 'public_interface' value of "in" or "out" is called a public 'variable' element. The mathematical equations of the functional element are marked up using MathML[13] in a 'mathml:math' element as a child node of the 'component' element. In a part that is separated from the definitions of 'component' elements, the relationships of functional elements on their variables are declared in 'connection' elements. A 'connection' element defines mappings of variables in one component to variables in another. 'Variable' elements with the 'public_interface' value of "in" in one component are connected to 'variable' with the 'public_interface' value of "out" in the other. **Figure 4** shows an example of a CellML file.

```
<!ENTITY % structural "compartment|separator">
<!ENTITY % reactional "reactant|instance">

<!ELEMENT model (%structural;)*>

<!ELEMENT compartment (%structural; | %reactional;)*>
<!ATTLIST compartment
        desc    CDATA   #IMPLIED
        name    NMTOKEN #REQUIRED
>

<!ELEMENT separator (%reactional;)*>
<!ATTLIST separator
        desc    CDATA   #IMPLIED
        name    NMTOKEN #REQUIRED
>

<!ELEMENT reactant EMPTY>
<!ATTLIST reactant
        desc    CDATA   #IMPLIED
        name    NMTOKEN #REQUIRED
>

<!ELEMENT instance (reference)*>
<!ATTLIST instance
        desc    CDATA   #IMPLIED
        name    NMTOKEN #REQUIRED
        type    CDATA   #REQUIRED
>

<!ELEMENT reference EMPTY>
<!ATTLIST reference
        desc    CDATA   #IMPLIED
        name    NMTOKEN #REQUIRED
        link    CDATA   #REQUIRED
>
```

Fig. 5 Entire DTD of PMSML.

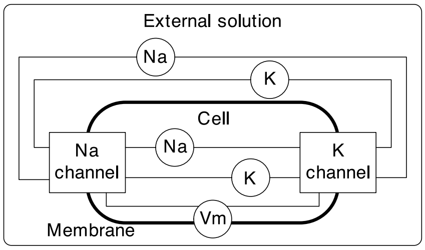


Fig. 6 An example of a model structure.

PMSML describes an anatomical hierarchy as an XML tree composed of 'compartment' tags, such as an intracellular space and a sarcoplasmic reticulum, and 'separator' tags, such as cell membranes and virtual boundaries of compartments. A declaration of a functional element is marked up with 'instance' tag in an anatomically appropriate node of the tree, while that of a global variable is marked up with 'reactant' tag. An 'instance' tag has a 'link' attribute with an XLink[14] value to a 'component' element in a CellML file which defines the model detail of the functional element. The referred variables of a functional element are specified with 'reference' tags in the 'instance' tag. In **Fig. 5**, the entire DTD (Document Type Definition) of PMSML is shown. **Figure 6** illustrates an example of a simple cell model, and **Fig. 7** depicts the description of the model in PMSML. In this example, a ventricular cell exists in an external solution. The cell and the external solution are separated by a cell membrane. In the cell

```
<?xml version="1.0"?>
<model>
<compartment name="xsol" desc="external solution">
 <reactant name="Na" desc="sodium concentration"/>
 <reactant name="K" desc="potassium concentration"/>

 <compartment name="cell" desc="ventricular cell">
  <reactant name="Na" desc="sodium concentration"/>
  <reactant name="K" desc="potassium concentration"/>

  <separator name="membrane" desc="cell membrane">
   <reactant name="Vm" desc="membrane potential"/>

   <instance type="km2003.xml#sodium_current"
             name="sodiumChannel">
    <reference name="Vm" link="[@name=’Vm’]"/>
    <reference name="Na_i" link="../[@name=’Na’]"/>
    <reference name="Na_o" link="../../[@name=’Na’]"/>
    <reference name="K_i" link="../[@name=’K’]"/>
    <reference name="K_o" link="../../[@name=’K’]"/>
   </instance>
   <instance type="km2003.xml#potassium_current"
             name="potassiumChannel">
    <reference name="Vm" link="[@name=’Vm’]"/>
    <reference name="Na_i" link="../[@name=’Na’]"/>
    <reference name="Na_o" link="../../[@name=’Na’]"/>
    <reference name="K_i" link="../[@name=’K’]"/>
    <reference name="K_o" link="../../[@name=’K’]"/>
   </instance>
  </separator>
 </compartment>
</compartment>
</model>
```

Fig. 7 An example of a PMSML file.

membrane, there are two functional elements, sodium and potassium channels, which refer to sodium and potassium concentrations in the external solution and the cell as well as the membrane potential.

2.2 Cell Model Ontology

To modify the structure of a whole cell model, frequent routine works must be done. These demanding tasks induce errors such as an inappropriate location of a functional element or a wrong connection of a functional element to a global variable. However, these routine works can be aided by using knowledge about cell models. A sensible solution for the electrical representation of knowledge is ontology. An ontology semantically defines a terminology and includes a taxonomy and relations of the terms. For model structure modifications, ontology is powerful for not only identifying model components, but also for allocating model components to an anatomical hierarchy and integrating information about major associations of functional elements with global variables. Although many ontologies have been constructed in biology, these ontologies are not suitable for our purposes here. For example, Gene Ontology[15)], a major ontology, includes a database with numerous protein molecules. The cellular component ontology of the Gene Ontology describes anatomical structures of cellular substances. On the other hand, physiological models often include abstract functional elements, which represent cellular functions but are not associated or not yet identified with molecular substances. These abstract functional elements are not registered in the cellular component ontology. Moreover, the Gene Ontology does not describe associations in physiological mechanisms. For these reasons, the Gene Ontology is inadequate for cell physiological modelling.

Here, a novel ontology for cell physiological modelling, the Cell Model Ontology is introduced. This ontology describes the knowledge about identifications, general relations and hierarchical locations of model components common in physiological models. The Cell Model Ontology is built as a set of entries, which are subtypes of a single implicit super concept, ‘model component’. Each entry of a model component has six attributes: ‘id’, ‘unit’, ‘keywords’, ‘location’ and ‘associations’. The last two attributes, ‘location’ and ‘associations’ describe relationships of concepts. An ‘ID’ attribute is the identification number, and a ‘name’ attribute is the standard name for this entry. A ‘unit’ attribute has strings of the base unit if this entry can be treated as a variable. A ‘keywords’ attribute includes a list of the terminological keywords of this entry. A ‘location’ attribute indicates another concept, which anatomically contains this concept. An ‘associations’ attribute holds a list of other concepts that have common major relationships with this concept. Currently, the Cell Model Ontology is stored in a proprietary text format.

2.3 Identifier Assignment to Model Components

Integration of multiple models is further complicated through the fact that a physiologically the same model component is often assigned different names in different models, such as the name of its function or the name of its protein complex. Especially in CellML files, model components defined in ‘`component`’ or ‘`variable`’ elements do not have standardized identification names.

Therefore, it is necessary to associate each element of a CellML file with an entry in the Cell Model Ontology. This is achieved by assigning an ontology ID number to each CellML element. An ID number is appended as a ‘`cmo:id`’ attribute to a ‘`component`’ tag or a public ‘`variable`’ tag of CellML. However, an ID is not allocated to model-specific components and variables, such as mathematical expressions and parameters without physiological meaning. **Figure 8** shows an example of

```
<component name="fast_sodium_current" cmo:id="520">
  <variable name="i_Na" public_interface="out" units="
microA_per_microF" cmo:id="220"/>
  <variable name="E_Na" public_interface="out" units="
millivolt" cmo:id="203"/>

  <variable name="g_Na" initial_value="16.0" units="mi
lliS_per_microF"/>

  <variable name="time" public_interface="in" private_
interface="out" units="millisecond" cmo:id="010"/>
  <variable name="V" public_interface="in" private_int
erface="out" units="millivolt" cmo:id="202"/>
  <variable name="R" public_interface="in" units="joul
e_per_kilomole_kelvin" cmo:id="006"/>
  <variable name="F" public_interface="in" units="coul
omb_per_mole" cmo:id="008"/>
  <variable name="T" public_interface="in" units="kelv
in" cmo:id="007"/>
  <variable name="Nai" public_interface="in" units="mi
llimolar" cmo:id="211"/>
  <variable name="Nao" public_interface="in" units="mi
llimolar" cmo:id="101"/>

  <variable name="m" private_interface="in" units="dim
ensionless"/>
  <variable name="h" private_interface="in" units="dim
ensionless"/>
  <variable name="j" private_interface="in" units="dim
ensionless"/>
```

Fig. 8 A fragment of a CellML file with ontology IDs.

a CellML file containing ontology ID numbers. In addition, '`cmo:id`' attributes are appended to '`instance`' and '`reactant`' tags in PMSML files.

For assistance in appending ontology ID numbers to CellML files, a method to estimate correspondences between model components in a CellML file and the instances of entries in the Cell Model Ontology is employed. In this method, all '`component`' elements in a CellML file are first estimated followed by all public '`variable`' elements. For target XML elements, '`component`' or '`variable`' elements, estimation scores of correspondences of all ontology entries are first evaluated by a score function using lexical analysis for attribute strings and structural analysis for the relations of XML elements. If the estimation reliability calculated from score values is higher than a certain threshold, then each target XML element is estimated as an intance of the first candidate entry. The detailed algorithm is described below (**Fig. 9**).

(1) Score values of all ontology entries for each target XML element are calculated by

$$s(e_i, o_j) = f_{\text{score}}(e_i, o_j)$$

where e_i is an element of a set E, and o_j is an element of a set O. E is the set of all '`component`' elements in the target CellML file for component estimations, or a set of all public '`variable`' elements for variable estimations; however, '`variable`' elements mapped in a '`connection`' tag are equated. O is the set of all entries in the ontology.

```
/* calculation of scores of each element for all entries */
for( $e_i \in E$ ) {
    for( $o_j \in O$ ) {
        $s(e_i, o_j)$ = calculate_score($e_i$, $o_j$);
    }
}

/* setting of initial candidates */
$C = E$;
for( $e_i \in E$ ) {
    $O_{\text{cand}}(e_i) = O$;
}

do {
    $fixed$ = false;

    /* calculation of reliability for all elements */
    for( $e_i \in C$ ) {
        $r_{e_i} = s(e_i, o_{b1}(e_i)) \ / \ s(e_i, o_{b2}(e_i))$;
    }

    /* fixing of the estimated entry for elements */
    for( $e_i \in C$ ) {
        if( $r_{e_i} > threthold$ ) {
            fixed_entry[ $e_i$ ] = $o_{b1}(e_i)$;
            $C = C - e_i$;
            for( $e_k \in C$ ) {
                $O_{\text{cand}}(e_k) = O_{\text{cand}}(e_k) - o_{b1}(e_i)$;
            }
            $fixed$ = true;
        }
    }
} while( $fixed$ );
```

Fig. 9 Algorithm for the identification of model components.

(2) The set C of target XML elements is initialized by E, and the set $O_{\text{cand}}(e_i)$ of candidate entries for each XML element e_i by O.

(3) A reliability $r(e_i)$ of an estimation for each XML element e_i is calculated by an empirical function:

$$r(e_i) = s\left(e_i, o_{b1}(e_i)\right) / s\left(e_i, o_{b2}(e_i)\right)$$

where $o_{b1}(e_i)$ is an entry with the largest $s(e_i, o)$ value in all elements of the set $O_{\text{cand}}(e_i)$, and $o_{b2}(e_i)$ with the second-largest value.

(4) If the reliability $r(e_i)$ for an XML element e_i in the set C is greater than a threshold value, then the element e_i is removed from the set C, and $o_{b1}(e_i)$ is removed from the set $O_{\text{cand}}(e_k)$ of each element e_k in the set C.

(5) If there is no reliability $r(e_i)$ greater than the threshold value, then this algorithm is terminated, or else the process is returned to step 3.

Here, f_{score} is a function $f_{\text{score}_\text{c}}$ for component estimations or $f_{\text{score}_\text{v}}$ for variable estimations. $f_{\text{score}_\text{c}}$ is defined as

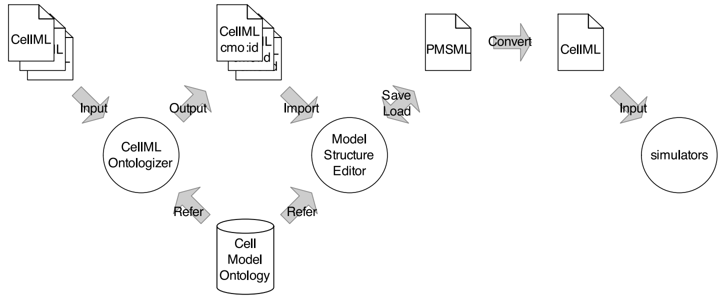


Fig. 10 Data flow of implemented tools.

$$f_{\mathrm{score_c}}(e_i, o_j) = w_{cn} \cdot f_{\mathrm{kwd}}(e_i, o_j) + w_{cd} \sum_{o_k \in O_{\mathrm{asc}}(o_j)} f_{\mathrm{kwd}}(e_i, o_k)$$

and $f_{\mathrm{score_v}}$ as

$$f_{\mathrm{score_v}}(e_i, o_j) = w_{vn} \cdot f_{\mathrm{kwd}}(e_i, o_j) + w_{vu} \cdot f_{\mathrm{unit}}(e_i, o_j) + w_{vi} \sum_{c_k \in C_{\mathrm{asc}}(o_j)} f_{\mathrm{cnt}}(e_i, c_k)$$

where w_{cn}, w_{cd}, w_{vn}, w_{vu} and w_{vi} are constants, $O_{\mathrm{asc}}(o_j)$ is a set of entries listed in the 'associations' attribute of the instance o_j, and $C_{\mathrm{asc}}(o_j)$ is a set of '`component`' elements identified to entries that include the entry o_j in the 'associations' attribute. The value of $f_{\mathrm{kwd}}(e, o)$ is calculated by the count of matched keywords to the name of e in the 'keyword' attribute of o. The value of $f_{\mathrm{unit}}(e, o)$ is 1 if the unit of e corresponds to the 'unit' attribute of o, or otherwise 0. The value of $f_{\mathrm{cnt}}(e, c)$ is 1 if the '`component`' element c contains the '`variable`' element e, or otherwise 0.

2.4 Assistance of Model Structure Manipulation

To make model structure modifications easier, a method to intelligently assist manipulations are employed. This method supports the following tasks by using information of the CellML file and the Cell Model Ontology: restricted allocation of an inserted model component to the anatomical hierarchy, intelligent addition of variables required for an inserted functional element, automatic connection of an inserted functional element to variables referred from the functional element, and extraction of functional elements that have major associations with a specified model component.

If a new functional element is inserted into an existing model structure, the expected anatomical location is determined by the 'location' attribute of the ontology entry referred from the '`component`' element of the functional element in the CellML file. Variables which need to be connected are recognized through the ontology ID numbers of the '`variable`' elements contained in the '`component`' element. If some variables do not exist in the model structure, these variables can be automatically added. For example, if a membrane transporter with ATP and ion concentrations as public variables is inserted into a model, the location is restricted to the cell membrane. The missing ATP variable is added first, and then all public interfaces are connected automatically.

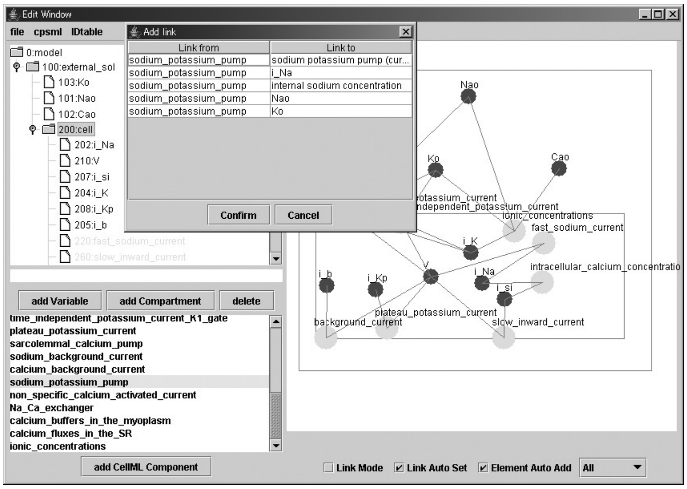
Furthermore, model components which have a major association with a specified model component can be discriminated by searching components identified to ontology entries that have the entry of the specified model component in the 'associations' attributes. If attention is limited to calcium dynamics, it is possible to extract only the model components that are associated with calcium concentrations.

2.5 Implementation

Two kinds of software were implemented. The first is an interactive utility to assign ontology ID numbers to a CellML file, the CellML Ontologizer. The second is a graphical editor for model structures, the Cell Structure Editor. The data flow of these tools is shown in **Fig. 10**.

The CellML Ontologizer supports appending attributes of ontology ID numbers to elements of a CellML file by using the estimation method described in 2.3. This utility first provides a list of multiple candidates for components. Then, an estimated list of variables is presented. The user can interactively correct in both steps.

The Cell Structure Editor supports modifications of complicated physiological models including imports of multiple component models. The GUI-based editor visualizes the model structure. Manipulations of the model structure are possible, such as an addition of a global



0:model
100:external_sol
103:Ko
101:Nao
102:Cao
200:cell
202:i_Na
210:V
207:i_si
204:i_K
208:i_Kp
205:i_b
220:fast_sodium_current
260:slow_inward_current

Link from	Link to
sodium_potassium_pump	sodium potassium pump (cur...
sodium_potassium_pump	i_Na
sodium_potassium_pump	internal sodium concentration
sodium_potassium_pump	Nao
sodium_potassium_pump	Ko

time_independent_potassium_current_K1_gate
plateau_potassium_current
sarcolemmal_calcium_pump
sodium_background_current
calcium_background_current
sodium_potassium_pump
non_specific_calcium_activated_current
Na_Ca_exchanger
calcium_buffers_in_the_myoplasm
calcium_fluxes_in_the_SR
ionic_concentrations

□ Link Mode ☑ Link Auto Set ☑ Element Auto Add All

Fig. 11 Screen shot of the Cell Structure Editor.

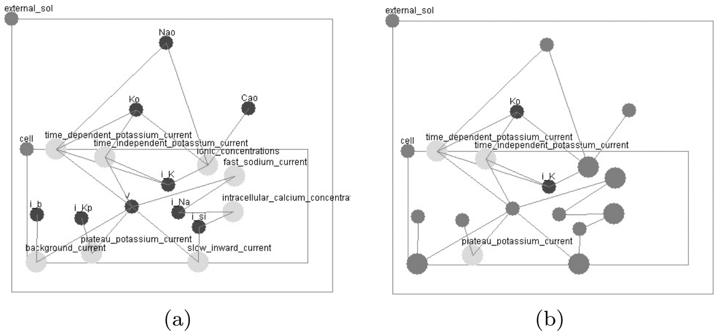

external_sol
Nao
Ko
Cao
cell
time_dependent_potassium_current
time_independent_potassium_current
ionic_concentrations
fast_sodium_current
i_K
V
i_Na
intracellular_calcium_concentra
i_si
i_b
i_Kp
plateau_potassium_current
background_current
slow_inward_current
external_sol
Ko
cell
time_dependent_potassium_current
time_independent_potassium_current
i_K
plateau_potassium_current
(a)
(b)

Fig. 12 Extraction of focused model components.

variable or a connection between a functional element and a global variable. The user can import functional elements to the model structure from CellML files of existing models. The user operations are assisted with confirmations, using the proposed method in 2.4. **Figure 11** shows an example of a screen shot. In addition, the editor can display a subset of model components, which are associated with a model component specified by the user. **Figure 12**-a shows a screen shot where all model components in a model are displayed, and Fig. 12-b shows an example where only model components associated with the membrane potential are displayed. An edited model structure can be saved to and loaded from a PMSML file.

3. Experiments

3.1 Estimation of CellML Elements

Experimental results for the estimation of ontology ID numbers for CellML elements are shown in **Table 1**. Public CellML files of the LRd,[☆1] Kyoto,[☆2] Jafri,[☆3], and ten Tusscher[☆4] Models are estimated using an ontology that is constructed from model components of just the LRd and Kyoto Models. Estimations of variables are performed after assignments of the correct ontology ID numbers to all components. Here, an estimation is defined as correct if the '`component`' or '`variable`' elements of CellML are fixed to the correct ontology entry, and as valid if one of the candidates with the highest score is the correct instance.

The ratio of valid estimations for all estimations is over 95%. In addition, the difference between models used for the construction of the ontology, i.e., the LRd and Kyoto Models,

[☆1] http://www.cellml.org/examples/models/LR_II_model_1994.xml

[☆2] http://www.cellml.org/examples/models/matsuoka_ventricular_cell_model_2003.xml

[☆3] http://www.cellml.org/examples/models/jafri_rice_winslow_model_1998.xml

[☆4] http://www.cellml.org/examples/models/ten_tusscher_model_2004.xml

Table 1 Estimations for CellML elements.

(a) Estimations for Components

	Targets	Correct estimations	Valid estimations
LRd	15	13 (87%)	15 (100%)
Kyoto	24	18 (75%)	21 (88%)
Jafri	14	12 (86%)	14 (100%)
ten Tusscher	13	12 (92%)	13 (100%)

(b) Estimations for Variables

	Targets	Correct estimations	Valid estimations
LRd	35	27 (77%)	33 (94%)
Kyoto	50	38 (76%)	48 (96%)
Jafri	27	24 (89%)	27 (100%)
ten Tusscher	24	21 (88%)	22 (92%)

Table 2 Operation counts for model manipulations.

Task	Assisted	Non-assisted
A	5	13
B	11	17
C	10	16

A: exchange of a time independent potassium current.
B: exchange of a sodium current.
C: insertion of a transient outward current.

and those which are not used, i.e. the Jafri and ten Tusscher Models, is negligible small. Nevertheless, three components of the Kyoto Model are estimated as wrong entries, because the names of these components include abbreviations of physiological terms, for instance the "non-selective cation current" is named as 'background_NSC_current'. For such a case, it is extremely difficult to determine the correct entry. Moreover, two cases of wrong variable estimations are detected. In one case, the variable belongs to a component that is not registered in the ontology. This problem can be resolved by expanding the ontology. In the other case, the name of the variable cannot be distinguished from others by keyword matching and structural analysis.

3.2 Efficiency of the Manipulation Assistance

The numbers of assisted and non-assisted user operations for three different model manipulations using the Cell Structure Editor are shown in **Table 2**. All tasks are applied to the structure of the LRd Model. In task A and B, the time independent potassium current and the sodium channel of the LRd Model are respectively exchanged with those of the Kyoto Model. In task C, the transient outward current of the Kyoto Model is inserted into the LRd Model.

The operation counts for all tasks with assistance are reduced by more than one-third compared to those without assistance. The required variables are inserted and connected automatically expect for model-specific variables, which are not registered in the ontology. These variables have to be inserted and connected manually. Therefore, the insertion of a functional element that contains many model-specific variables, such as a sodium channel or a transient outward current of the Kyoto Model, as used in task B and C, requires a comparatively large number of operations.

4. Discussion

The high rate of validity of the experimental results for estimations of CellML elements confirms that the proposed method is effective in appending appropriate ID numbers to CellML files. However, the ratio of correct estimations is not so high, and a major improvement of the estimation method is necessary to resolve this issue. Nevertheless, perfect estimations for all model components are impossible due to several lexical problems, and current valid rates are considered practical because the user of the CellML Ontologizer can choose the correct entry from listed candidates. Furthermore, almost the same result is obtained for models not registered to the ontology as for those that are registered. This indicates that the schema of the Cell Model Ontology and the estimation method are generally valid for all physiological cell models.

The assistance of model manipulations using PMSML and the Cell Model Ontology is proved to be efficient. In addition, the possibility to display a subset of focused model components using the ontology improves the visibility of complex cell models. These features further facilitate the manipulation of complex cell models, compared to the abilities of existing graphical model editors such as the CellML Editor and the Virtual Cell.

Moreover, a PMSML file edited with the Graphical Model Editor can be easily converted to a new CellML file by merging all referred CellML files. The created file can be handled by CellML-compliant software. This means that an edited model can be readily simulated for validation. Therefore, implemented software utilities can accelerate repeated modifications and simulations of a model, as well as with an integrated modelling environment such as CESE and the Virtual Cell.

5. Conclusion

This paper reports methods that support the development of the structures of complex cell models using a markup language, PMSML, and an ontology, the Cell Model Ontology. Experimental results for estimations of CellML elements show that the proposed ontology and the estimation method are effective for physiological cell models. Furthermore, the assistant method for model structure manipulations is shown to be practical. The implemented software facilitates the development process for cell models. Future work includes the expansion of the proposed ontology. An efficient development of complex cell models using these methods and implementations will promote further progress in cell physiological research.

Acknowledgments The authors wish to thank Prof. Akinori Noma and members of Kyoto University Biosimulation Centre for valuable comments during the study. We also thank Dr. Natalie Schneider for improving the English of this paper. This work was supported by the Leading Project for Biosimulation from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1) Luo, C.-H. and Rudy, Y.: A dynamic model of the cardiac ventricular action potential I. Simulations of ionic currents and concentration changes, *Circulation Research*, Vol.74, No.6, pp.1071–1096 (1994).

2) Jafri, M.S., Rice, J.J. and Winslow, R.L.: Cardiac Ca^{2+} Dynamics: The Roles of Ryanodine Receptor Adaptation and Sarcoplasmic Reticulum Load, *Biophysical Journal*, Vol.74, No.3, pp.1149–1168 (1998).

3) Matsuoka, S., Sarai, N., Kuratomi, S., Ono, K. and Noma, A.: Role of individual ionic current systems in ventricular cells hypothesized by a model study, *Japanese Journal of Physiology*, Vol.53, No.2, pp.105–123 (2003).

4) ten Tusscher, K.H., Noble, D., Noble, P.J. and Panfilov, A.V.: A model for human ventricular tissue, *American Journal of Physiology Heart and Circulatory Physiology*, Vol.286, No.4, pp.1573–1589 (2004).

5) Garny, A., Kohl, P. and Noble, D.: Cellular Open Resource (COR): A public CellML based environment for modelling biological function, *International Journal of Bifurcation and Chaos*, Vol.13, No.12, pp.3579–3590 (2003).

6) Missan, S. and McDonald, T.F.: CESE: Cell electrophysiology simulation environment, *Applied Bioinformatics*, Vol.4, No.2, pp.155–156 (2005).

7) Sarai, N., Matsuoka, S. and Noma, A.: simBio: A Java package for the development of detailed cell models, *Progress in Biophysics and Molecular Biology*, Vol.90, No.1–3, pp.360–377 (2006).

8) cellml.org: CellML Tools, http://www.cellml.org/tools.

9) Loew, L.M. and Schaff, J.C.: The Virtual Cell: A software environment for computational cell biology, *Trends in Biotechnology*, Vol.19, No.10, pp.401–406 (2001).

10) Cuellar, A. A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P. and Hunter, P.J.: An Overview of CellML 1.1, a Biological Model Description Language, *SIMULATION*, Vol.79, No.12, pp.740–747 (2003).

11) Cuellar, A., Nielsen, P., Halstead, M., Bullivant, D., Nickerson, D., Hedley, W., Nelson, M. and Lloyd, C.: CellML 1.1 Specification, http://www.cellml.org/specifications/cellml_1.1 (2006).

12) Christie, G.R., Bullivant, P.D., Blackett, A.S. and Hunter, P.J.: Modelling and visualising the heart, *Computing and Visualization in Science*, Vol.4, No.4, pp.227–235 (2002).

13) Carlisle, D., Ion, P., Miner, R. and Poppelier, N.: Mathematical Markup Language (MathML) Version 2.0 (Second Edition), http://www.w3.org/TR/2003/REC-MathML2-20031021/ (2003).

14) DeRose, S., Maler, E. and Orchard, D.: XML Linking Language (XLink) Version 1.0, http://www.w3.org/TR/2001/REC-xlink-20010627/ (2001).

15) Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. and Sherlock, G.: Gene Ontology: tool for the unification of biology, *Nature Genetics*, Vol.25, No.1, pp.25–29 (2000).

(Received July 15, 2006)
(Accepted September 11, 2006)

(Communicated by *Toshiyuki Amagasa*)

Takao Shimayoshi was born in 1974. He received his M.E. degree from Kyoto University in 1999. He had worked in Mitsubishi Electric Coop. since 1999 and engaged in research on network systems. Since 2003 he has been a researcher in ASTEM Research Institute of Kyoto. His current research interests are biological simulation. He is a member of IPSJ, IEEJ and IEEE-EMBS.

Kazuhiro Komurasaki was born in 1981. He received his Master of Informatics degree from Kyoto University in 2006. He has been working in IBM Japan since 2006.

Akira Amano was born in 1965. He received his Ph.D. degree from Kyoto University in 1995. He had worked in Hiroshima City University as an associate professor since 1995 and has been in Kyoto University as an associate professor since 2002. His current research interests are biological simulation and image processing. He is a member of IPSJ, IEICE, JSMBE, JSAI, IEEE-EMBS and IEEE-CS.

Takeshi Iwashita was born in 1971. He received the M.E. degree and Ph.D. degree from Kyoto University in 1995 and 1998 respectively. From 1998 to 1999, he worked as a research associate in Kyoto University. In 2000, he joined the Data Processing Center, Kyoto University. Since 2003 he has worked as an associate professor in the Academic Center for Computing and Media Studies, Kyoto University. From 2003 to 2004, he was a visiting fellow of Griffith University, Australia. His research interests include high performance computing, linear iterative solvers, and electromagnetic field simulations. He is a member of IPSJ, JSAEM, JSCES and IEEE.

Tetsuya Matsuda was born in 1956. He received his M.D. and Ph.D. degree from Kyoto University in 1981 and 1988 respectively. He had worked in Kyoto University Hospital as a cardiologist from 1988. He was an associate professor, Department of Medical Informatics, Kyoto University Hospital from 1997, and has been a Professor, Graduate school of Informatics, Kyoto University since 2000. His current research interests are cardiovascular Magnetic Resonance imaging and biological simulation of cardiac motion. He is a member of ISMRM, SCMR, and IEEE-EMBS.

Masanori Kanazawa was born in 1946. He received M.E. from Kyoto University in 1971. From 1972 he was an assistant professor, associate professor and professor of Data Processing Center, Kyoto University. He is now a professor of Academic Center for Computing and Media Studies, Kyoto University. He was also obtained a Ph.D. degree in engineering from Kyoto University. His current research interests include high performance computing and distributed systems. He is a member of ACM, IEICE and JSIAM.