Toward Provably Correct Models of Ventricular Cell Function

RL Owen[1], S McKeever[2], J Davies[2], A Garfinkel[3]

[1]Department for Continuing Education, University of Oxford, Oxford, UK
[2]Computing Laboratory, University of Oxford, Oxford, UK
[3]Departments of Medicine (Cardiology), Physiological Science, and Physiology, Cardiovascular Research Laboratory, University of California School of Medicine, Los Angeles, CA, USA

Abstract

Researchers in cardiac mechanics and electrophysiology develop computer models for analyzing complex experimental data. A key issue is model correctness: formally verifying that the model is performing as intended. We present an application of formal software engineering methods to an established electrophysiology model: the Beeler-Reuter (B-R) model of the mammalian ventricular myocyte. A formal specification fragment for the B-R model is developed, which captures the key drivers of the transmembrane potential, including four ionic currents (I_{Na}, I_{s}, $I_{\mathrm{x_1}}$, and $I_{\mathrm{K_1}}$) and a representation for the intracellular calcium ion concentration ($[\mathrm{Ca}]$). Correctness-preserving transformations can be used to refine the specification into executable code, thereby assuring a provably correct implementation. The mathematical and logical tools presented here provide a rigorous approach to proving the correctness of ventricular cell models, thereby improving program implementation and verification.

1. Introduction

The sequencing of the human genome is providing new insights into the structure and function of complex biological systems [1]. These systems, composed of hierarchies of modular components and protocols interconnected by layers of feedback regulation [2, 3], form intricate networks that are robust to imprecise components or perturbations in the environment, but are fragile to trivial mutations or unexpected perturbations for which the system was not optimized. For example, the fraction of possible amino acids sequences that yield functioning proteins is vanishingly small [4], so that a single genetic mutation can be lethal, such as the KCNQ1 single nucleotide polymorphism (SNP) that underlies congenital long-QT syndrome.

Given this complexity, the challenge for cardiac researchers is to understand cardiac structure and function using computational models for analyzing large sets of experimental data across a range of spatial and temporal scales. A current focus is on standard tools and notations for storing and communicating these models. One example is CellML [5], an open XML-based language designed to address model standardization and publishing inaccuracies [6], the latter being of two types: (a) errors introduced in the computer code or its description during the publishing process; or (b) programming errors in the code itself. These errors can lead to a knowledge gap that can propagate to subsequent versions of the code.

Even if the computer code correctly implements the published description given in the paper, improved experiments and data can cause a mismatch between model results and experimental observations. Computational models are therefore under continuous revision, both to improve performance and to match improved experimental results. Many of these revisions are done on an *ad hoc* basis, leading to a second knowledge gap in the form of a library of models whose correctness is largely unknown.

While standardization efforts such as CellML can address the first type of publishing error, the question of model correctness due to programming errors or poorly-documented revisions remains. A method for proving model correctness should address the following: (a) providing a provably correct path between abstract models; (b) providing a provably correct path from the abstract models to executable code; and (c) providing a library of reusable software combinators which can be instantiated to create complex cellular and even whole organ models. This paper represents a first step toward the development of the method, by applying formal software engineering methods to the Beeler-Reuter (B-R) model of the mammalian ventricular myocyte [9]. The modeling and proof approach described here provides a rigorous foundation for the otherwise *ad hoc* specification and design of computational models.

Section 2 presents an overview of the methods, including a review of the Z notation and the B-R model. In Section 3, a formal specification for the B-R model is presented. Section 4 discusses the results and conclusions.

ISSN 0276–6547

2. Methods

Formal software engineering methods use mathematical objects and notation to model complex systems. A formal specification describes system properties at higher levels of abstraction using these mathematical objects and notation [10], allowing rigorous system verification. The formal notation is comprised of: (1) rules for determining the grammatical well-formedness of sentences (syntax); (2) rules for interpreting sentences in a precise, meaningful way within the application domain (semantics); and (3) rules for inferring useful information from the specification (proof theory).

2.1. The Z notation

Z is a formal notation that combines discrete mathematics and logic with macro-like abbreviations called *schemas* [11]. Z includes a schema calculus that allows complex schemas to be constructed from simple schemas, and a refinement calculus for translating schemas into executable code. The basic Z notation is supplemented by a collection of mathematical theorems, axioms and laws describing sets, tuples, relations, functions, sequences and their operators that provide for the formal proof of many system properties. This proof strategy ensures that the data type requirements are consistent and that the operations are applied only within their domains.

The Z notation can support a variety of complex systems and modeling methods. For example, Z can model system behavior as an abstract formal specification, or it can form a design specification comprising modules, data types, procedures, functions, classes and objects.

2.2. Beeler-Reuter model

An early cardiac cell model was developed by G.W. Beeler and H. Reuter [9], using four individual, discrete ionic currents and six Hodgkin-Huxley type gating variables. In contrast to the earlier Purkinje fiber models of Noble [7] and R.E. McAllister, D. Noble and R.W. Tsien [8], the B-R model focused on the mammalian ventricular myocyte. The currents of the B-R model include: (1) an initial fast inward sodium current I_{Na}, similar to the one used by Hodgkin and Huxley; (2) a time-dependent outward current I_{x_1}; (3) a time-independent potassium outward current I_{K_1}: and (4) a secondary slow inward current I_{s}, the latter carried primarily by calcium ions which is responsible for cardiac cell contraction and the action potential (AP) plateau. Thus, the total ionic current in the B-R model is given by four currents and uses eight variables: membrane potential, six ionic gates (m, h, j, x_1, f and j) and the intracellular calcium concentration ([Ca]).

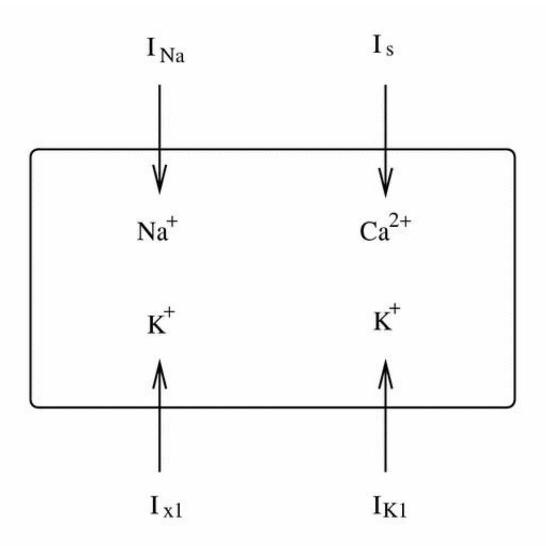

Figure 1. A schematic diagram describing the current flows across the cell membrane that are captured in the Beeler-Reuter model (after [5]).

3. Results

3.1. Abstract specification

A basic excitable cell, such as a cardiac myocyte, is modeled as a relation between types representing currents and potentials:

$$[\mathit{CURRENT}]$$

$$\mathit{CHANNELS} == \mathbb{P}\,\mathit{CURRENT}$$

$$\begin{array}{|l} \mathit{POTENTIAL} : \mathbb{P}\,\mathbb{Z} \end{array}$$

As shown in Figure 1, these currents and potentials, representing continuous, time-dependent signals, provide the minimum functionality required of any excitable cell.

The cell model (system state) is comprised of a transmembrane potential and a set of ionic currents:

$$\begin{array}{|l} \hline \mathit{Cell} \\ \mathit{transmembrane} : \mathit{POTENTIAL} \\ \mathit{currents} : \mathit{CHANNELS} \\ \hline \end{array}$$

The transmembrane potential is defined as the potential drop across the inner and outer surfaces of the cell membrane. The ionic currents are mediated by sodium (Na^+), potassium (K^+), and calcium (Ca^{2+}) ions.

An initialization schema defines the state of the cell after it has been initialized:

$$\begin{array}{|l}
\textit{CellInit} \\
\hline
\textit{Cell}' \\
\hline
\textit{transmembrane}' = 0 \land \textit{currents}' = \emptyset \\
\hline
\end{array}$$

where $\emptyset$ denotes the empty set.

An action is a temporal change in transmembrane potential triggered by membrane depolarization. Following depolarization, the absolute and relative refractory periods are modeled by a gate function that maps a pair of potentials and an ionic channel to an ionic channel:

$$\begin{array}{|l}
\textit{gate} : (\textit{POTENTIAL} \times \textit{POTENTIAL} \\
\qquad\qquad \times\ \textit{CHANNELS}) \pfun \textit{CHANNELS}
\end{array}$$

An action function models the response of the cell to depolarization by mapping the ionic currents to a potential:

$$\begin{array}{|l}
\textit{action} : \textit{CHANNELS} \pfun \textit{POTENTIAL}
\end{array}$$

The following operation schemas uses *gate* and *action* to depolarize the cell:

$$\begin{array}{|l}
\textit{DepolarizeCell}_0 \\
\hline
\Delta \textit{Cell} \\
v? : \textit{POTENTIAL} \\
\hline
v? \geq 0 \\
(v?, \textit{transmembrane}, \textit{currents}) \in \operatorname{dom}(\textit{gate} \fatsemi \textit{action}) \\
\textit{transmembrane}' = \\
\qquad (\textit{gate} \fatsemi \textit{action})\ (v?, \textit{transmembrane}, \textit{currents}) \\
\hline
\end{array}$$

$$\begin{array}{|l}
\textit{DepolarizeCell}_0 \\
\hline
\Delta \textit{Cell} \\
v? : \textit{POTENTIAL} \\
\hline
v? \geq 0 \\
(v?, \textit{transmembrane}, \textit{currents}) \mapsto \\
\qquad\qquad \textit{transmembrane}' \in \textit{gate} \fatsemi \textit{action} \\
\hline
\end{array}$$

The predicates check that the action is positive and in the domain of the *gate* function. The function $\textit{gate}\ (v?, \textit{transmembrane}, \textit{currents})$ updates the ionic currents according to the input potential and the present transmembrane potential and ionic currents. The function $\textit{action}\,\textit{currents}'$ updates the transmembrane potential corresponding to the updated ionic currents.

3.2. Concrete design

Using the abstract specification, we now derive a concrete design which removes non-determinism and enriches the model, by adding structure to the ionic currents and including a representation for the intracellular calcium concentration. As shown in Figure 2, the concrete design represents the minimum components and functionality of the B-R cell model [9].

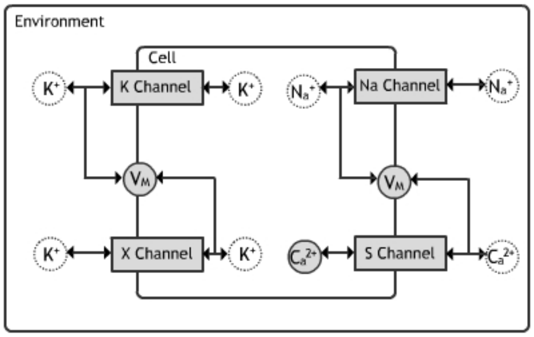

Figure 2. A schematic diagram of the ionic channels and ion concentrations for a four-channel cell model. The shaded regions denote the components of the Beeler-Reuter model.

The BR model includes a type for the intracellular calcium ion concentration:

$$\begin{array}{|l}
\textit{CONCENTRATION} : \mathbb{P}\,\mathbb{N}
\end{array}$$

As with the currents and potentials, the concentration may be continuous and time-dependent. The BR cell is comprised of a transmembrane potential *vm*, four ionic channels *channels*, and an intracellular calcium concentration:

$$\begin{array}{|l}
\textit{BRCell} \\
\hline
\textit{vm} : \textit{POTENTIAL} \\
\textit{channels} : \operatorname{seq} \textit{CURRENT} \\
\textit{cai} : \textit{CONCENTRATION} \\
\hline
\#\textit{channels} = 4 \\
\hline
\end{array}$$

The *BRCell* is initialized as follows:

$$\begin{array}{|l}
\textit{BRCellInit} \\
\hline
\textit{BRCell}' \\
\hline
\textit{cai}' = 0 \land \textit{vm}' = 0 \land \textit{channels}' = \langle\,\rangle \\
\hline
\end{array}$$

where $\langle\,\rangle$ denotes the empty sequence. Note that the predicates of *BRCell* and *BRCellInit* ensure that the intracellular calcium concentration cannot take negative values.

Next, we define a function that updates the intracellular calcium concentration:

$$\begin{array}{|l}
\textit{calcium} : (\operatorname{seq} \textit{CURRENT} \times \textit{CONCENTRATION}) \\
\qquad\qquad \pfun \textit{CONCENTRATION}
\end{array}$$

and a new gate function:

$$\begin{array}{|l}
\textit{brgate} : (\textit{POTENTIAL} \times \textit{POTENTIAL} \\
\qquad\qquad \times \operatorname{seq} \textit{CURRENT}) \pfun \operatorname{seq} \textit{CURRENT}
\end{array}$$

The *DepolarizeBRCell* operation schema is now given by:

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{DepolarizeBRCell}_0} \\
\hline
\Delta \textit{BRCell} \\
v? : \textit{POTENTIAL} \\
\hline
v? \geq 0 \\
(v?, \textit{vm}, \textit{channels}) \mapsto \textit{channels}' \in \textit{brgate} \\
\operatorname{ran} \textit{channels}' \mapsto \textit{vm}' \in \textit{action} \\
(\textit{channels}', \textit{cai}) \mapsto \textit{cai}' \in \textit{calcium} \\
\hline
\end{array}
$$

3.3. Forward simulation

We now have two models or "views" of the B-R ventricular myocyte, however, the question remains: are these two views mathematically and logically consistent, both internally and with each other? A formal relation between *Cell* and *BRCell* is given by a retrieve relation that is functional from the concrete view to the abstract view:

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{BRCellRetrieveCell}} \\
\hline
\textit{Cell} \\
\textit{BRCell} \\
\hline
\textit{currents} = \operatorname{ran} \textit{channels} \wedge \textit{transmembrane} = \textit{vm} \\
\hline
\end{array}
$$

The retrieve relation *BRCellRetrieveCell* may be used to formally prove that $\textit{DepolarizeBRCell}_0$ is correct with respect to its corresponding abstraction. This involves proving that the retrieve relation is a total function, and that the initialization and depolarization operations are correct:

$$
\begin{array}{l}
\forall \textit{BRCell} \bullet \exists_1 \textit{Cell} \bullet \textit{BRCellRetrieveCell} \\
\forall \textit{Cell}';\ \textit{BRCell}' \bullet \textit{BRCellInit} \wedge \\
\qquad\qquad \textit{BRCellRetrieveCell}' \Rightarrow \textit{CellInit} \\
\forall \textit{Cell};\ \textit{BRCell} \bullet \operatorname{pre} \textit{DepolarizeCell} \wedge \\
\qquad\qquad \textit{BRCellRetrieveCell} \Rightarrow \operatorname{pre} \textit{DepolarizeBRCell} \\
\forall \textit{Cell};\ \textit{Cell}';\ \textit{BRCell};\ \textit{BRCell}' \bullet \operatorname{pre} \textit{DepolarizeCell} \wedge \\
\qquad\qquad \textit{BRCellRetrieveCell} \wedge \textit{DepolarizeBRCell} \wedge \\
\qquad\qquad\quad \textit{BRCellRetrieveCell}' \Rightarrow \textit{DepolarizeCell}
\end{array}
$$

Once we have proved the correctness of our concrete design with respect to the abstract specification, we may either proceed with another round of design and refinement, using our first concrete design as the abstract specification, or translate the concrete design into executable code.

4. Discussion and conclusions

We have developed a formal specification fragment for the B-R mammalian ventricular myocyte. From a theoretical perspective, our method improves the formulation and proof of computational models and may be extended to more complex biological systems.

One advantage of the formal specification is that implementation details can be ignored in order to focus on the most important aspects of the problem. This is particularly important for computational models that are that too novel or complex to develop by intuition or by modifying existing code. A second advantage is that it provides two very important proof opportunities [11], which demonstrate that the data type requirements are consistent and that the operations are applied only within their domains. These proofs are available prior to code development, testing and publication, and have the potential to reduce model development time and improve model documentation.

The formal specification and modeling process described here can be readily extended to more complex cell models, as demonstrated here by the addition of the intracellular calcium concentration. In addition, we are developing methods to formally evolve computational models in response to improved or more accurate experimental data. By abstracting these evolutions, along with refinements and translations, within a common mathematical structure, we hope to demonstrate tools and methods that can simplify the design, development, and maintenance of computational models of the heart.

References

[1] Lesk AM. Introduction to Bioinformatics. UK:Oxford, 2001

[2] Csete ME, Doyle J. Reverse engineering of biological complexity. Science 2002;295:1664–9.

[3] Stelling J, Sauer U, Szallasi Z, Doyle FJ, Doyle J. Robustness of cellular functions. Cell 2004;118:675–85.

[4] Carlson JM, Doyle J. Complexity and robustness. Proc Natl Acad Sci USA 2002;99:2538–45.

[5] See `www.cellml.org`.

[6] Lloyd C, Halstead MDB, Nielsen PF. CellML: its future, present and past. Prog Biophys Mol Biol 2004;85:433–50.

[7] Noble D. A modification of the Hodgkin-Huxley equations applicable to Purkinje fibre action and pacemeker poitentials. J Physiol 1962;160:317–52.

[8] McAllister RE, Noble D, Tsien RW. Reconstruction of the electrical activity of cardiac Purkinje fibres. J Physiol 1975;251:1–59.

[9] Beeler GW, Reuter H. Reconstruction of the action potential of ventricular myocardial fibres. J. Physiol. 1977;268:177–210.

[10] van Lamsweerde A. Formal Specification: a Roadmap. In: Finkelstein A, editor. The Future of Software Engineering. Limerick Ireland: ACM Press, 2000;149–59.

[11] Woodcock J, Davies J. Using Z: Specification, Refinement and Proof. Essex UK: Prentice Hall 1996.

Address for correspondence:

Name: Randall L. Owen
Full postal address: 11523 Treeview Ct, Moorpark, CA 93021
E-mail address: randall.owen@new.ox.ac.uk