

B-09

ISML Java シミュレーター環境の紹介

矢野 憲† 倉智 嘉久†‡

Ken Yano Yoshihisa Kurachi

1. はじめに

複雑な生体機能の解明には、多数の現象の相互作用を考慮した数理モデルの構築が必要である。数理モデルをコンピュータ上に作成するには、数式とデータによるモデルの記述があれば十分であることが多いため、そのモデルを使ってシミュレーションを行う場合、実際にコンピュータ上に実装する作業は論文読者に任されている。一方で複雑な数理モデルの作成に関して、これまでも汎用的なモデル記述言語仕様が提案されてきた（文献[1,2,3]）。いずれの言語仕様においても、モデルは XML によるメタデータとして記述される。一度作成されたモデルは汎用的に利用可能であるため、データベースなどに登録され、研究者間での相互利用が可能となっている。このように汎用モデルの記述文法については詳細に規定されているが、依然として、その数理モデルから実行可能なシミュレーションのソースコードへの変換については、ソフトウェア開発者の責任であると言ってもよい。特定言語仕様で作成されたモデル向けのシミュレーターソフトウェアが提供されている場合もあるが、その内部処理について詳細に記述されていることは多くない。

本論文では、大阪大学の *Physiome* プロジェクト（文献[1]）で使用されている、*insilicoML* で作成されたモデルを実行可能な Java のソースコードに変換するツールの機能概要を説明する。また本ツールを使用して開発を行っている Java による統合的な GUI シミュレーション環境（*ISML J-Simulator*）についても紹介する。

2. クラス構成

insilicoML の詳細な仕様については、参考文献[1]を照会されたい。構築されているモデルは様々な要素がツリー状の階層構造で作成されているため、それらの異なる種類の要素を Java のクラスとして 1 対 1 に対応するように設計することにより、メタデータのツリー構造とクラスの構造がほぼ同一になるようにした。このようにクラスを構成することにより、出来上がったソースコードの可読性が向上し、言語仕様が拡張された場合や、保守性を考えた場合にも有効な選択であると思われる。

モデルのメタデータと同様に、クラス構造の一番外側に、モデルクラスがあり、その内側に複数のモジュールクラスのオブジェクトをメンバーとして持つ。モジュールクラスには、そのモジュールで定義された全ての *Port* と *Physical-Quantity*（以下 *PQ*）がリスト型オブジェクトとして格納されている。モジュール、*PQ*、*Port* には複数の種類があるため、それぞれ抽象型の親クラスを規定し、機能別のクラスをそれぞれ、具象クラスとして実装している。こうすることにより、異なる具象クラスからの作成されたインスタンスオブジェクトを親クラスのインスタンスとして統一的に扱うことが可能になる。

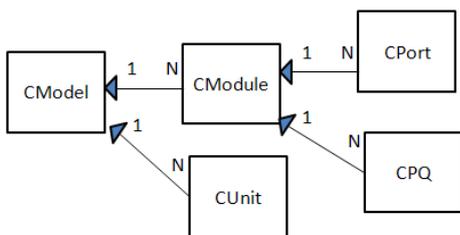


図1 クラスの階層構造

トップレベルにモデルクラス(CModel)が存在し、そのメンバーとして、複数のモジュールクラス(CModule)とユニットクラス(CUnit)が登録されている。各モジュールクラスには複数のポートクラス(CPort)と *Physical-Quantity* クラス (CPQ) が登録されている

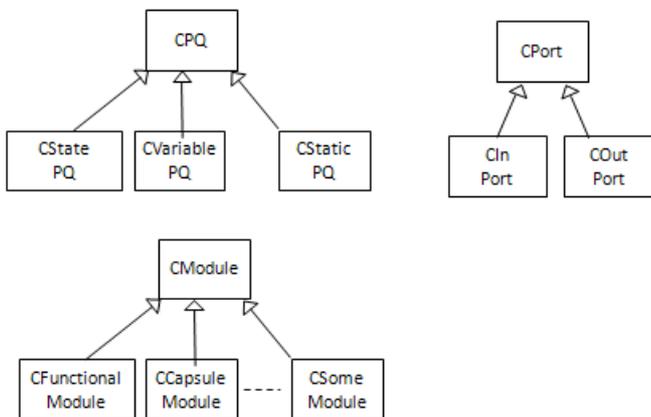


図2 クラスの親子関係

モジュール、*Physical-Quantity*(PQ)、ポートの多様性は、抽象クラス、具象クラスを用いて定義している

†大阪大学臨床医工学融合研究教育センター

‡大阪大学大学院医学系研究科

3. モジュール間のデータ同期処理

insilicoML (文献[1]) のモデルは各モジュールが階層的に相互結合されており、結果として異なるモジュール間の PQ で複雑なデータの依存性が存在する。それらのデータはモジュール間を連結するエッジを使って送受信される。エッジの両端には入力・出力ポートが定義されており、PQ の値が他のモジュールに依存するような場合はそのポートを通じてデータを取得することになる。このように複数のモジュールがネットワークとして複雑に連結しているシステムの場合、モジュール間で共有使用されているデータの同期が重要となる。特に、PQ の値が合成関数式で定義されており、内部の関数が別モジュールの PQ で定義されているような場合には、どのデータを先に計算するかについてシミュレーションを始める前にデータ間の依存性チェックを行う必要がある。もっとも、このようなデータ依存性がないように数理モデルを構築していれば、データの計算順序は問題とはならないが、汎用的に動作するようなツールを作成する場合には、このような特殊なケースも考慮しながら設計を行う必要がある。

シミュレーション途中でモジュール構成が動的に変化しない静的な数理モデルであれば、データ依存性については、静的なデータとして記憶領域に保存でき、以後その記憶領域を参照しながら PQ の値の更新を行えばよいが、マルチエージェントシステムのように、モジュールの構成が動的に変化する場合には、タイムステップ毎にデータ依存性をチェックする必要がある。本ソフトウェアでは、動的にモジュール構成が変更する場合も考慮して、タイムステップ毎にデータ依存性のチェックを行うことにした。以下にデータ同期のアルゴリズムの概略を述べる。

3.1 データ同期アルゴリズム

- ① シミュレーションを始める前に、各モジュール内で使用されている各 PQ について、他のモジュールで定義されるデータに依存していれば、それをモジュール内の記憶領域へ保存しておく
- ② 各タイムステップで全てのモジュールにおいて処理③を行う
- ③ 各タイムステップでモジュール内の PQ を更新する際に、他のモジュールに依存するデータがあれば、そのモジュールにデータの問い合わせを行う。問い合わせを受けたモジュールは、データが更新済みであればそのデータを渡すが、データが更新されていない場合は、自身のモジュール内で計算できれば計算を行うが、さらに他のモジュールのデータに依存していればそのモジュールへ問い合わせる。モジュール内でデータ更新が完了していれば、データが更新されていることを示すフラグをセットすることにより、それ以降に当該データの問い合わせが他のモジュールからある場合は、そのデータを渡す。
- ④ 全てのモジュールのデータの同期が終了していれば、微分方程式により定義される状態変数の値を更新する
- ⑤ 全てのモジュールのデータ更新フラグをリセットして、時間を次のタイムステップに進めて処理②を行う。(シミュレーション途中でモジュール構成が変更されれば処理①を行った後に処理②を行う)
- ⑥ 累積時間がシミュレーション時間になっていれば終了する

4. 統合シミュレーション環境

GUI のシミュレーションフレームワーク部と、上記のシミュレーション実行部とをシミュレーション実行しているモデルに依存しない形式でつなぎ合わせるインターフェース層を設けることで、GUI 部分の開発とシミュレーション実行部の開発を同時並行的に進めることが可能となる。こうすることにより、仮にシミュレーションの実行部のアルゴリズムに変更があった場合でも GUI 部分への影響を最小限にすることが可能となる。インターフェースには図 3 に示すように 3 つのメソッドのみが規定され、GUI 部はそのインターフェースを通じて、モデルのパラメータ値の取得・変更、シミュレーション実行を行う。

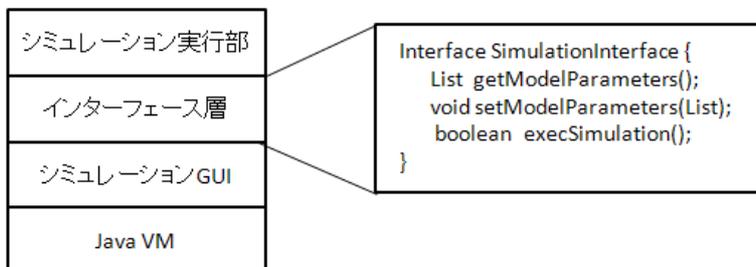


図 3 統合シミュレーション階層
シミュレーション GUI 部とシミュレーション実行部はインターフェース層を通じて通信し、モデルパラメータの取得・設定、シミュレーション起動を行う

4.1 統合シミュレーション実行例

例として Hodgkin-Huxley モデル(文献[5])と心筋細胞モデルでよく使用される Luo Rudy モデル (文献[6]) のシミュレーション実行した結果の例を図4に示す。使用しているモデルは公開されているデータベース (文献[4]) より取得可能である。

機能は限定されているが、本論文で紹介した ISML Java シミュレーターを Google 社より無償で開放されているサーバー環境にインストールしており、ユーザーは JavaWebStart を使用してシミュレーター起動することが可能である。(文献[7])

5. 謝辞

本ソフトウェアの開発にあたり、多くの方にご協力をいただきました。特に大阪大学臨床医工学融合研究教育センターの野村泰伸様、浅井義之様、大阪大学大学院の基礎工学研究科バイオダイナミクス研究室の弓倉陽介君には多大なサポートを頂きました。ここに、心より感謝の意を表します。

6. 参考文献

1. Physiome.jp ホームページ <http://www.physiome.jp/>
2. SBML http://sbml.org/Main_Page
3. CellML <http://www.cellml.org/>
4. InsilicoModel DB <http://www.physiome.jp/modeldb/index1.php>
5. L. Hodgkin, F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve." J.Physiol.,117,500-544,1952
6. C.H.Luo, Y. Rudy, "A model of the ventricular cardiac action potential. Depolarization, repolarization, and their interaction." Circ Res,68,1501-1526,1991
7. ISML J-Simulator <http://isml-j-simulator.appspot.com/>

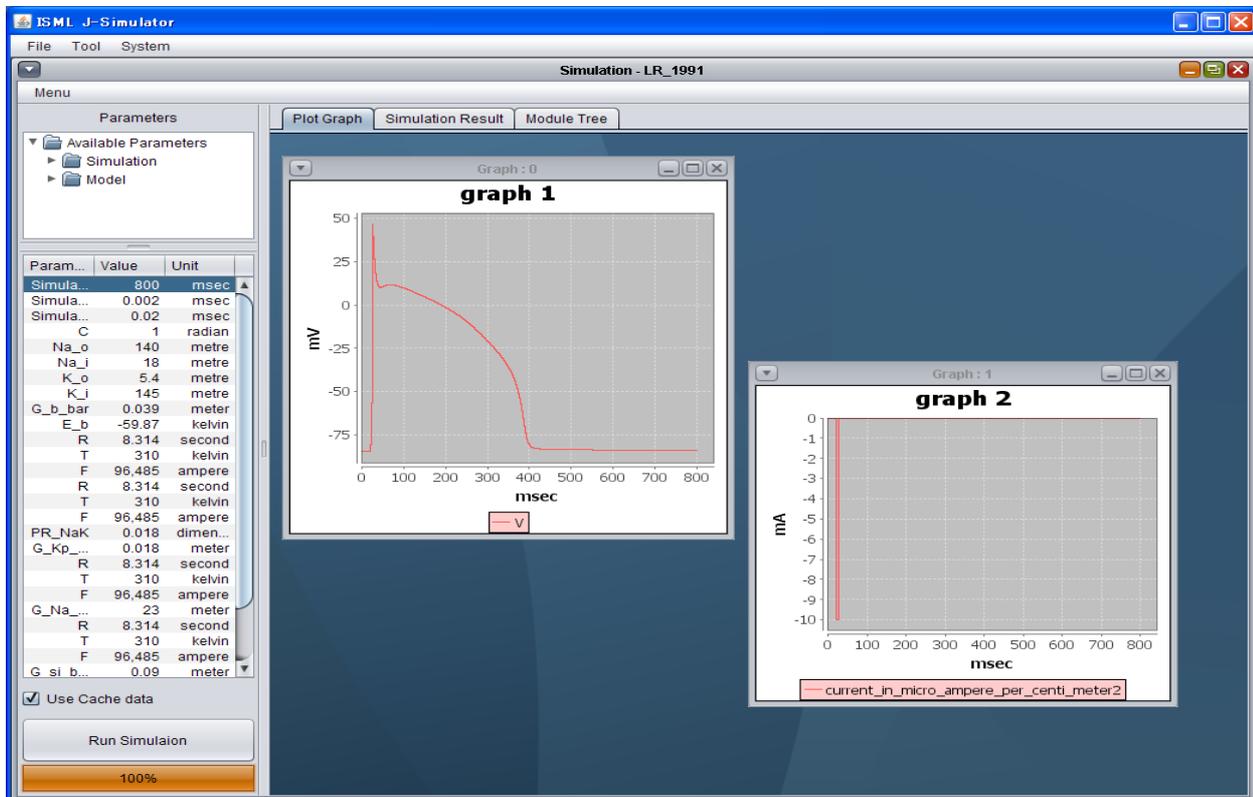


図 4-1 Luo Rudy 心筋モデルのシミュレーション実行画面

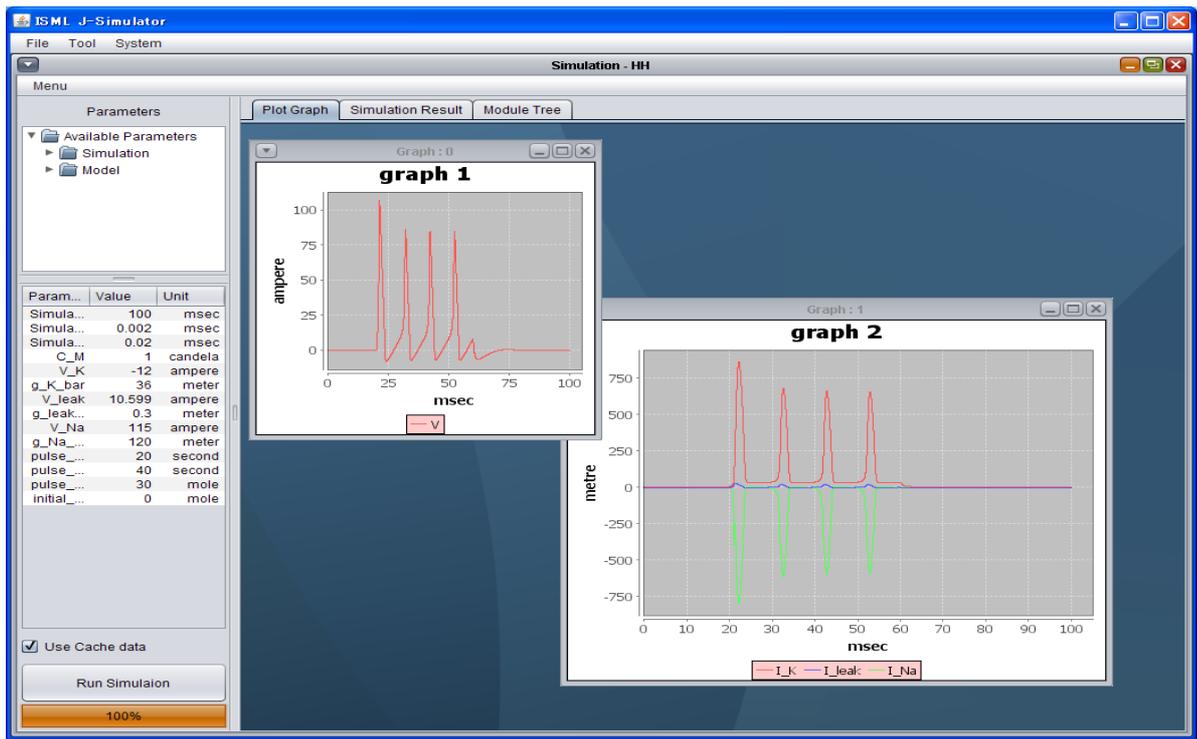


図 4-2 Hodgkin-Huxley モデルのシミュレーション実行画面