

Cell Illustrator における視覚化支援ツールの開発と検討

田中寿宜* 北風裕教** 原田勝通*** 松野浩嗣*** 宮野悟****

Tools for Supporting Computational Experiments on Cell Illustrator

Toshinori TANAKA, Hironori KITAKAZE, Katsuyuki HARADA,
Hiroshi MATSUNO and Satoru MIYANO

Abstract

Recently, many tools for biological pathway simulations have developed aiming deeper understanding of molecular interaction in cells. We are developing Cell Illustrator that has Petri net as a basic architecture and also Cell Animator that can output molecular action by intuitive animation. However, it was necessary to make the tag of XML to do the animation of Cell Animator and it was difficult for the user. Then, CI Builder that was making XML only by GUI operation was developed, and effectiveness was examined compared with a past method.

Key words: Petri Net, Cell Illustrator, Cell Animator, CI Builder

1. 緒言

新薬や治療法の開発のために生物学者、薬学者による生物実験が世界中で行われている。しかし生物実験とその解析には、非常に多くの費用・時間・労力、そして生物実験に利用される生命の犠牲が必要となった。現在では研究対象の焦点を絞り込むことで、その負担を軽減するために生命をシステムとして捉えて問題解決をしようとするバイオインフォマティクスの分野が注目されている。このようなシステムの1つとして生命パスウェイ(経路図)を計算機上でモデル記述し、シミュレーション実験から問題の提起や解決、そして応用することを可能にするツールの開発が世界中で行われている。ツールを利用することで、分析対象を絞り込むことができれば、生物実験を最小限に抑え、効率の良い解析が行え、生物分野の急速な発展が期待できる。

開発されるツールは、情報やメカニズムの表し方の違いにより、ある程度それぞれの分野に特化して開発される場合が多い。そのため仕様の異なる様々なツールが存在し、モデル化手法も異なる。モデル化

手法において複雑な微分方程式を用いるツールやプログラムを必要とするツールの場合、数学に長けた研究者には容易に理解できるが、生物学者や薬学者にとっては数学や論理的なモデルを理解するのは困難な作業であった。

そこで我々の研究グループでは、代謝経路、遺伝子制御ネットワーク、シグナル伝達系等の生命パスウェイを表すためにペトリネット(PN)を基本アーキテクチャに持つツール、Cell Illustrator の開発を行っている[1, 5]。Cell Illustrator はユーザーが、自分で思い描くパスウェイについての知識を直感的に記述でき、モデル化やシミュレーションの細かい部分までパラメータの操作ができるように考慮されている。

また我々は、Cell Illustrator のシミュレーション結果をユーザーがアニメーションで直感的に理解できるようにパスウェイの可視化を目的とした環境ツール Cell Animator の開発も行っている。

Cell Illustrator では、プログラミングを必要としなかったが、Cell Animator のアニメーションを定義するためには、XML のタグを手作業で書く(コーデ

ィングする)必要があり, 数学やプログラミングに明るくないバイオリジストにとってアニメーションを作成することは困難な作業であった. そのため具体的な評価に時間を必要とする問題点が残った. したがって本研究では Cell Animator のアニメーション定義ファイル(XML)を GUI の操作だけで作成できるツール, CI Builder[4]の開発を行った. これにより, ユーザーによるモデル化の作業効率が向上したので報告する. 本稿では, まず Cell Illustrator と, それに属する Cell Animator を記述する. 次に Cell Animator に関わる問題点を挙げ, その問題を解決する補助ツール CI Builder の提案と開発について述べ, 従来手法との比較検討を行う.

2. システム構成

本システムは, 生命現象のモデル化を行いシミュレーションを行うソフトウェア Cell Illustrator と, そのデータを元に視覚化を行うソフトウェア Cell Animator の二つのツールが基本システムとなる. それぞれについて説明する.

2.1 Cell Illustrator

Cell Illustrator の概要, 基本アーキテクチャであるペトリネットの基本構造とその拡張, そしてシミュレーション及び問題点について述べる.

2.1.1 Cell Illustrator の概要








	discrete elements	continuous elements
Place		
Transition		
Token	Integer	Real Number
Arc	 normal	 test
		 inhibitor

図1 HPNの基本構成要素

Cell Illustrator は代謝系, 遺伝子制御ネットワーク, シグナル伝達系などの生体内パスウェイの経路図を描く感覚でモデル化しシミュレーションすることができるツールである. Cell Illustrator はモデル化とシミュレーションをするためにペトリネット(Petri net) (PN)を基礎とした方式であるハイブリッド関数ペトリネット(HFPN) [2, 3]を利用している. 以下に PN に拡張を行ったハイブリッドペトリネット(HPN), そしてさらに生物モデルに対応させるた

めさらに拡張が行われた HFPN について述べる.

2.1.2 HPNの基本要素

PN は, グラフ表現の一種であり大別してプレース, トランジション, アーク, トークンの4つの要素から構成される. 最も基本的な PN は, トークンの値に対して離散値のみを許す離散 PN であるが, 生物事象は物質の濃度変化のような連続値で表現した方が都合の良い場合がある. 離散値と連続値の両方を扱うことができる PN に hybrid Petri net (HPN) がある(図1). このモデルを利用し Cell Illustrator では生命事象のパスウェイを構築する.

生命事象を作成する際, 次の定義に従い HPN でパスウェイを作成する. HPN を用いれば生物事象の模式図に沿って生命パスウェイを実現できる.

- タンパク質や mRNA 等の遺伝子産物の濃度を「連続プレース」で表す.
- 細胞の形状・状態・条件を「離散プレース」で表す.
- リン酸化・結合・崩壊等の現象を「連続トランジション」で表す. その反応速度をトランジションの発火速度で表す.
- シグナルや外部刺激, 形態変化のタイミングや条件の判断に要する時間は, 「離散トランジション」で表す.
- 各信号が影響を与えるまでの時間を離散トランジションの遅延時間で表す.
- 反応の方向, 影響の種類(状態の移行, 活性化, 抑制化)をアークで表す.

2.1.3 HFPN への拡張

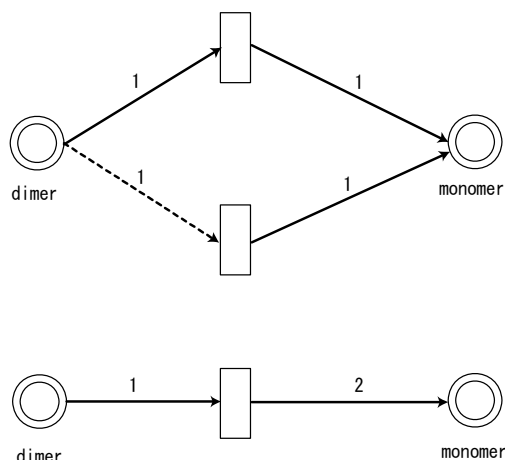


図2 分解(上: HPN, 下: HFPN)

HPN は連続トランジションの発火から消費されるトークン数と出力されるトークン数の比が一定であ

るため、自然にモデル化できない生命パスウェイがある。例えば 1 分子の 2 量体が 2 分子の 1 量体になる事象や、2 分子の 1 量体が 1 分子の 2 量体になる事象である。よって HPN を改良し、トランジションの発火速度をアークごとに定義付けを可能にしたハイブリッド関数 PN(HFPN) を生命パスウェイの表記に利用する。図 2 に 1 分子 2 量体から 1 量体への分解の様子を HPN と HFPN によって表現したものを示す。このように 1 分子 2 量体が 2 分子 1 量体に分解される場合や、逆に 2 分子の 1 量体が 1 分子の 2 量体に結合する場合において、HFPN はトランジションをひとつ記述するのみで生命パスウェイを表現することを可能にする。






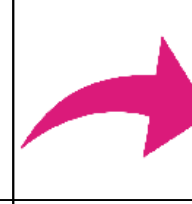
		
タンパク質	mRNA	活性化
		
切断	転写	結合

図 3 絵付きエレメント

2.1.4 Cell Illustrator の HFPN の変更点

HFPN でエレメントの名前が、プレース、トランジション、アークと呼ばれていたのに対し Cell Illustrator ではエンティティ (Entity)、プロセス (Process)、コネクター (Connector) と呼ぶ。呼称が違っただけで役割に変化は無い。Cell Illustrator ではさらにこれら基本的なエレメントに絵を付けたエレメント (図 3) (エンティティとプロセス) を用意しており、これらを利用することで物質や現象を直感で理解でき、パスウェイのモデルをより綺麗に分かりやすく作成することができる。

Cell Illustrator はこれらの優れた GUI 環境を提供することにより、生物学者や薬学者のユーザーにとって難解な微分方程式やプログラミング等に振り回されることなく、直感的なパスウェイ構築とシミュレーションが実現できる。

2.1.5 Cell Illustrator で生命モデルを表す

生体内では、結合、分裂、生成、崩壊など様々な現象が複雑に絡み合うことで構成されている。ここでは、

結合について一例を挙げて説明する。

生体内で、あるタンパク質 A と B が結合 (Binding) して複合体を作るという生命事象を Cell Illustrator でモデル化すると図 4 のようになる。それぞれのエンティティがタンパク質を示し、プロセスによって複合体の形成というイベントを表現している。またエンティティ中のトークン数 m_1, m_2, m_3 がタンパク質濃度を示している。プロセス p_1 の発火により、エンティティ e_1, e_3 のトークン数は同時に失われ、複合体を表すエンティティ e_2 のトークン数 m_2 が増加する。

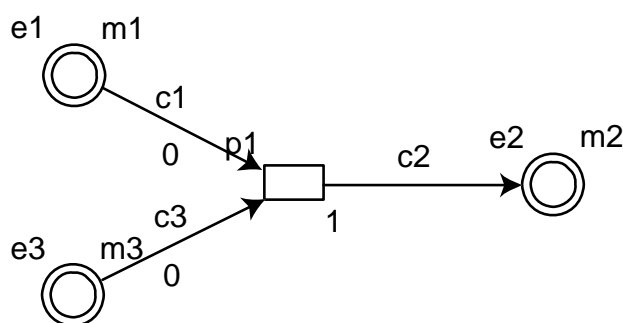


図 4 複合体の形成 (結合)

2 種類以上の物質による複合体も同様に表現できる。プロセスは、全ての入力エンティティにトークンが存在していないと発火可能とならない。このことを利用して、同じプロセスに対して、必要となる物質を表すエンティティ全てからのコネクターを入力すればよい。

2.1.6 Cell Illustrator シミュレーション実験

Cell Illustrator のシミュレーションはグラフに出力でき、CSV 形式で保存することができる。CSV には、エンティティの濃度量とプロセスの発火状態及びエンティティ、プロセス、コネクターの一覧が保存できる。

図 4 で作成したタンパク質の結合のモデルを Cell Illustrator でシミュレーションを行った。エンティティの初期濃度量を $e_1=100, e_3=70$ 、プロセスの結合速度は $p_1=(m_1*m_3)/1000$ として、実験を行った結果が図 5 である。縦軸が濃度量、横軸が時間である。シミュレーションの変遷を CSV 形式で出力した結果を図 6 に示す。ここで Time は時間を表し、 e_1, e_2, e_3 はエンティティの濃度量を表している。 e_1, e_3 のトークンが少なくなるとプロセスの反応速度は減少し、結合速度が遅くなる。時間が経つに連れてエンティティの濃度量は $e_1=30, e_2=70, e_3=0$ に収束することが分かる。

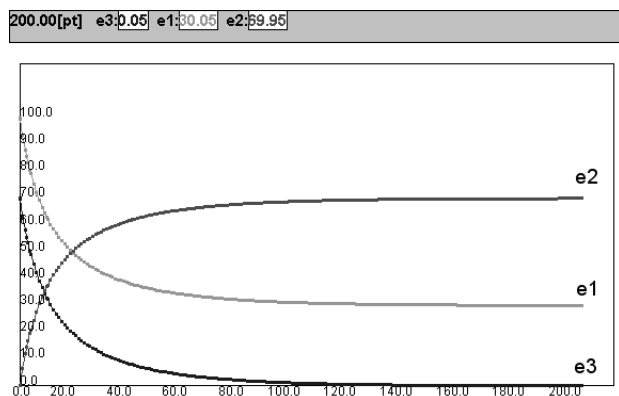


図5 複合体形成のシミュレーション結果

Time	e3	e2	e1	Time	e3	e2	e1
0	70	0	100	10	32.283311	37.716689	62.283311
1	63.536658	6.4633418	93.536658	11	30.361392	39.638608	60.361392
2	58.017098	11.982902	88.017098	12	28.60652	41.39348	58.60652
3	53.250579	16.749421	83.250579	13	26.99851	43.00149	56.99851
4	49.094526	20.905474	79.094526	14	25.520304	44.479696	55.520304
5	45.440224	24.559776	75.440224	15	24.15738	45.84262	54.15738
6	42.203324	27.796676	72.203324	16	22.897301	47.102699	52.897301
7	39.317382	30.682618	69.317382	17	21.729356	48.270644	51.729356
8	36.729343	33.270657	66.729343	18	20.644264	49.355736	50.644264
9	34.396329	35.603671	64.396329	19	19.633951	50.366049	49.633951

図6 複合体形成のシミュレーション結果(CSV)

2.1.7 Cell Illustrator の問題点

シミュレーションの状況はグラフ化したもので確認することができるが、数値の変化だけでは、結合の状況や物質が集中及び移動の様子を捉えにくい。つまり Cell Illustrator のシミュレーション結果だけでは生命現象のイメージが付きにくいという問題があった。そこで、開発されたのが視覚化ツール Cell Animator である。

2.2 Cell Animator

2.2.1 Cell Animator の概要

Cell Animator は、Cell Illustrator からシミュレーション結果である CSV ファイルをアニメーションで表現することのできる視覚化ツールである [5]。Cell Animator は XML の技術をベースにして作成されているため、多少のプログラミングの知識があればホームページの HTML のタグを書くような感覚でシミュレーションの視覚化を行うことができる。Cell Animator によるアニメーション作成に必要なのはシミュレーションによる CSV ファイルと物質や反応を表す画像ファイル (PNG 形式) とプログラム能力である。いくつかのアニメーションサンプルやマニュアル [5] も準備されており、それを利用すればそれほど困難なく目的のアニメーションを作成できる。Cell Animator を用いることで Cell Illustrator でのシミュレーションをアニメーションで表現できるので生命現象の直感的な理解が容易にできる。

2.2.2 Cell Animator のシミュレーション

実際に複合体形成のアニメーションの XML を作成しシミュレーションを行う。予め使用する画像をグラフィック作成ソフトで準備をしておく。XML 作成の手順を次に示す。

1. 利用する PNG ファイルと CSV ファイルの参照先, ID の設定をする。
2. アニメーションウィンドウを作成する。
3. アニメーションウィンドウの子供としてアニメーション対象物を設定する。座標の設定, 範囲の設定, 配置の割合, 角度の設定等を行う。
4. アニメーション対象物の中にアニメーションを設定する。画像やテキストの大きさを設定する。
5. アニメーションイベントを設定する。CSV に従って画像の移動, 色変化のイベントに閾値, 周期等を調整する。

作成した Cell Animator の XML を図 7 に、Cell Animator のアニメーションを図 8 に示す。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE test SYSTEM "./GonVisTool.dtd">
<test>
  <simulation animationClock="0.2" physicalClock="0.05" />
  <!-- image -->
  <imageFile id="blue-image" file="./bluePlotein.png" />
  <imageFile id="red-image" file="./redPlotein.png" />
  <imageFile id="bind-image" file="./rbBinding.png" />
  <imageFile id="frame" file="./empty.png" />
  <!-- CSV -->
  <tableFile file="./Binding.csv">
    <placeName id="blue-entity" name="e1" />
    <placeName id="red-entity" name="e3" />
    <placeName id="bind-entity" name="e2" />
  </tableFile>

  <!-- animation -->
  <animation title="Binding Animation" backColor="1 1 1">
    <!-- set Frame -->
    <animeSingleObject position="25 100 -1">
      <animeImage image="frame" size="60 60"/>
    </animeSingleObject>
    <animeSingleObject position="25 25 -1">
      <animeImage image="frame" size="60 60"/>
    </animeSingleObject>
    <animeSingleObject id="Bind_Position" position="125 75 -1">
      <animeImage image="frame" size="60 60"/>
    </animeSingleObject>
    <animeMultiObject lowerBound="0 0 0" upperBound="50 50 0">
      <animeImage image="red-image" size="5 5"/>
      <animeMultiVisibility event="red-entity" period="-1"
      discretization="2" increasingOnly="false" />
    </animeMultiObject>

    <animeMultiObject lowerBound="0 0 0" upperBound="50 50 0">
      <animeImage image="red-image" size="5 5"/>
      <animeMultiVisibility event="red-entity" period="-1"
      discretization="4" increasingOnly="false" />
      <animeMultiMotion event="red-entity"
      reference="Bind_Position" type="repeat" motionPeriod="10"/>
    </animeMultiObject>
  </animation>
</test>
```

図7 手作業で作成した Cell Animator の XML

表1 Cell Animator アニメーション定義 XML Element 一覧

エレメントの役割	エレメント名	設定内容
アニメーション設定文書	test	ルートエレメント
アニメーション速度	simulation	アニメーション速度
画像ファイルの取り込み	imageFile	使用する画像ファイル
数値データの取り込み	sequenceFile	Cell Illustrator 出力ファイル(CSV) (1つのエンティティ値のみの場合)
	tableFile	Cell Illsutrator 出力ファイル(CSV) (複数のエンティティ値の場合)
	placeName	tableFile のプレース
アニメーションウィンドウ	animation	アニメーションを表示するウィンドウ
アニメーション対象物	animeSigleObject	画像を1つ表示するオブジェクト
	animeMultiObject	画像を複数表示するオブジェクト
	animeStraightSequence	複数の画像を一直線に並べる
	animeRingSequence	複数の画像を円環状に並べる
	animeStructureElement	Straight, Ring において使用する画像
アニメーション	animeText	テキスト
	animeImage	使用する画像
	animeSingleVisibility animeMultiVisibility	対象物の表示/非表示イベント
	animeSingleColor animeMultiColor	対象物の色変化イベント
	animeSingleMotion animeMultiMotion	対象物の移動イベント
	プロッターウィンドウ	plotter
プロットデータ	plotSeries	エンティティ値への参照

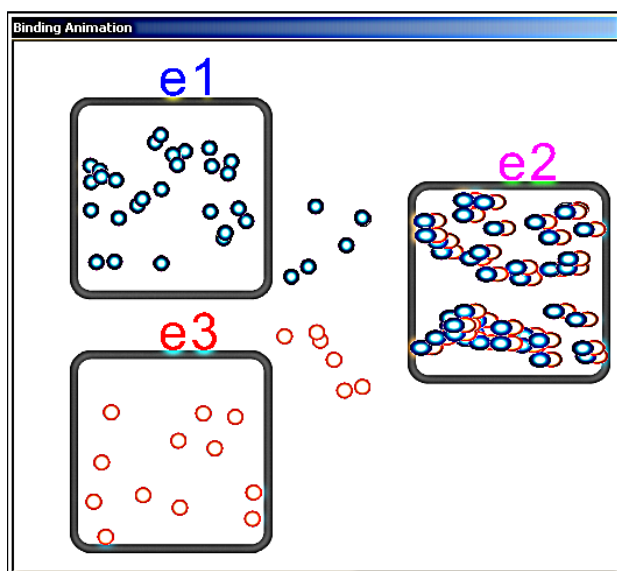


図8 複合体形成のアニメーション

2.2.3 Cell Animator の問題点

Cell Animator の XML をコーディングするには変数の定義, パラメータの設定, エラーの修正を反復しな

がらプログラミングを行わなければならない. 生物学者や薬学者にとってプログラミングは困難な作業であり, ユーザー自身がシミュレーションを行うための効率を損ねていた. そこで HTML を作成するときの Homepage Builder のように, GUI を利用した絵を貼り付けて簡単な設定をするだけで Cell Animator の XML を自動作成できるツールがあると便利であると考えられる.

本研究では, これらの問題を解決する XML 自動変換ツール CI Builder を作成した. 以下に作成した CI Builder について詳しく説明する.

3. CI Builder の開発

Cell Animator の XML をコーディングする場合の問題点を挙げ, 対処法を考え, 具体的な図を設計し, プログラムに実装する.

3.1 Cell Animator の XML 構造解析

Cell Animator における XML の書式は GonVisTool.dtd に定義される. 図9に DTD ファイルの構造とデータの参照関係を示す. この図でアニメーション設定ファイル内の長方形アイコンの上はエ

表2 手作業によるXML作成時の問題点

作成時の問題点	関連事項
座標やサイズ, 色, 時間など数値を設定する場合, 視覚的に判断できないため, 何度も Animator で実行しながら調整する必要がある.	<ul style="list-style-type: none"> simulation の時間 各オブジェクトの座標, サイズ 各ウインドウやプロッター用ラインの色, オブジェクトの色変化の設定
id 名の設定が非効率的, 参照名が一致する必要がある.	<ul style="list-style-type: none"> imageFile や数値データ読み込み用エレメント, オブジェクトの id アニメーション設定エレメントからの参照名
アニメーション設定時, CSV データの数値だけだと変化が理解し難い.	<ul style="list-style-type: none"> アニメーション設定エレメントの各属性

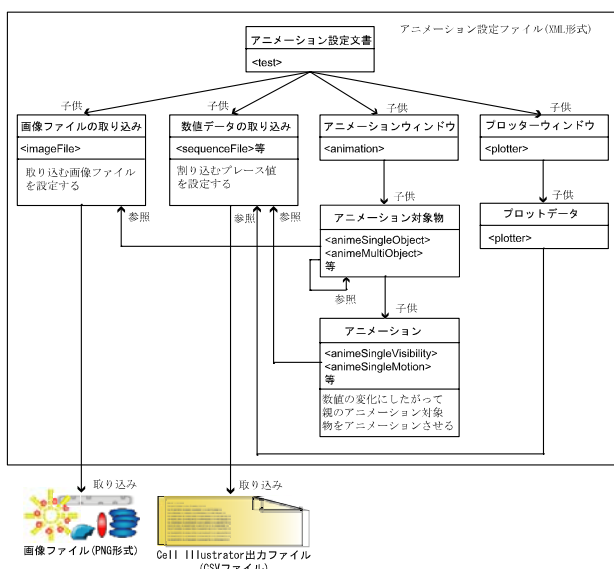


図9 GONVISTOOL.DTD の構造

レメントの役割, 下は XML のエレメントの名前を表している. また表1に Cell Animator のXMLのエレメントの一覧を種類ごとに分けて示す.

testエレメントはアニメーションのXMLのルートである. その子供としてPNGやCSVの設定を行うエレメント (imageFile, SequenceFile) とアニメーションとグラフ設定用のエレメント (animation, plotter) がある. さらに animation と plotter の子供は画像やシミュレーション結果を参照し, 具体的なアニメーションを行う設定を行うエレメントがある. このXMLのエレメントの組み合わせによって様々なアニメーションを実現できる.

3.2 コーディングの問題

表2にXMLをコーディングする時の問題点を挙げる. これらの問題を解決すれば様々な部分を自動化し簡略化することができる.

3.3 開発言語について

本システムは開発言語としてマルチプラットフォームで動作可能である Java を用いる. インターネットを通してソフトウェアを配布することが Java は可能であり, どの OS でも使用できるという特徴はユーザを選ばない. また, プログラミングスタイルや言語仕様を C 言語から多く受け継いでいるため, 多くのプログラマーが Java を理解しやすいこと.

Java 実行環境がメモリ開放, 参照渡し of the management を行ったり, 例外処理, アクセス制限の機構によってプログラムの安全性が保たれること. 通信, XML, グラフィックなどの規定されたクラスライブラリを持っていること. そして, オブジェクト指向言語の概念で設計されており, ソフトウェア開発効率と保守性を高めていることが挙げられる.

3.4 問題の解決案と実装

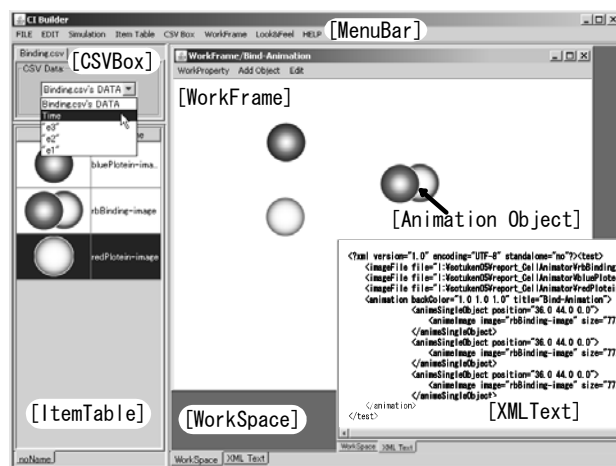


図10 CI Builder

表3に問題を解決するための解決案をエレメントの単位で示し, 実際に Java でプログラミングを行う. 操作のほとんどを GUI 化して, ユーザーがコードを直接書く操作を極力減らすようにする.

表3の解決案を元に作成した CI Builder の概観を図10に示し, CI Builder のクラス構造を図11に示す. 図10と図11における [Animation Object],

表 3 問題点の解決案

問題の種類	解決案
アニメーション速度	GUI の設定ダイアログを準備する。
画像ファイルの取り込み	格納, 表示テーブル(ItemTable)を作成し, 読み込んだ画像と名前を表示する。画像を読み込むダイアログを作成する。
数値データの取り込み	格納・表示用コンボボックスを作成し, 読み込んだ CSV のエンティティ名を表示する。ID はエンティティ名をそのまま割り当てる。数値をグラフで表示する機能があると便利になる。
アニメーションウィンドウ	ウィンドウを複数表示できる領域(WorkFrame)を作成し, アニメーション対象物の追加, 変更, 削除を行なう。ウィンドウの設定ダイアログ (数値と色の設定ができるダイアログ) を準備する。
アニメーション対象物	ItemTable からドラッグ&ドロップで画像を配置する。WorkFrame の画像をクリックするとポップアップメニューを表示する。WorkFrame に置かれた画像は, ドラッグで移動できるようにする。一直線や円環状, 複数配置のオブジェクトを配置するときには, 専用のダイアログを準備する。
アニメーション	アニメーションに必要な要素を入力できるダイアログを作成する。XMLText で変更したソースはWorkFrame や XML の保存がそのままできるようにする。
ソースコードに関して	XML のソースを表示する領域(XMLText)を作成し, 実際のソースを確認できるようにする。
メニュー	各種ファイル(PNG, CSV, XML)の読み込み, XML の保存, 印刷, 編集等を行なうメニューを作成する。

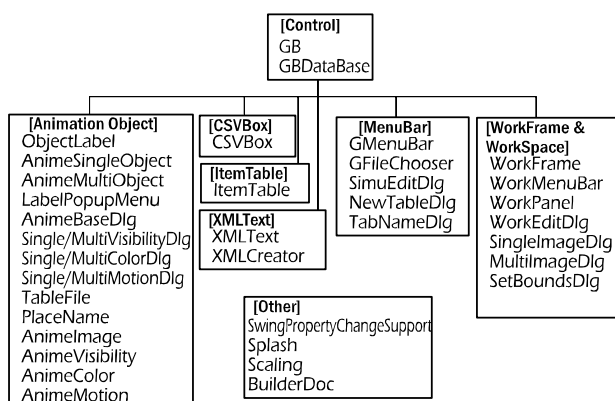


図 11 CI Builder のクラス関係

[CSVBox], [ItemTable], [MenuBar], [WorkFrame], [WorkSpace], [XMLText]の GUI とクラスが対応している。[Control]は全体のデータの管理や受け渡しが行われている。

3.5 従来の手法との比較

作成した CI Builder を用いて従来の手法との比較検討を行った。手作業で用いた XML と同等のアニメーションを CI Builder で作成する。CI Builder での XML 作成の手順を示す。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><test>
<simulation animationClock="0.2" physicalClock="0.05"/>
<imageFile file="I:\sotuken05\report_CellAnimator\bluePlotein.png"
id="bluePlotein-image"/>
<imageFile file="I:\sotuken05\report_CellAnimator\redPlotein.png"
id="redPlotein-image"/>
<imageFile file="I:\sotuken05\report_CellAnimator\rbBinding.png"
id="rbBinding-image"/>
<imageFile file="I:\sotuken05\report_CellAnimator\empty.png"
id="empty-image"/>

<tableFile file="Binding.csv">
<placeName id="e1" name="e1"/>
<placeName id="e3" name="e3"/>
<placeName id="e2" name="e2"/>
</tableFile>

<animeSingleObject position="-150.0 19.0 0.0">
<animeImage image="empty-image" size="50.0 50.0"/>
</animeSingleObject>
<animeSingleObject position="-90.0 40.0 0.0">
<animeImage image="empty-image" size="50.0 50.0"/>
</animeSingleObject>
<animeSingleObject position="-150.0 74.0 0.0">
<animeImage image="empty-image" size="50.0 50.0"/>
</animeSingleObject>
<animation backColor="1.0 1.0 1.0" title="Binding_Animation">
<animeMultiObject lowerBound="-166.0 58.0 0.0" upperBound="-133.0
89.0 0.0">
<animeImage image="bluePlotein-image" size="5.0 5.0"/>
<animeMultiVisibility discretization="2" event="e1"
increasingOnly="False" period="-1" threshold="0"/>
</animeMultiObject>
<animeMultiObject lowerBound="-165.0 5.0 0.0" upperBound="-134.0
33.0 0.0">
<animeImage image="redPlotein-image" size="5.0 5.0"/>
<animeMultiVisibility discretization="2" event="e3"
increasingOnly="False" period="-1" threshold="0"/>
</animeMultiObject>
```

図 12 CI Builder による Cell Animator の XML

1. 利用する PNG ファイルと CSV ファイルを読み込む。
2. 画像を ItemTable から WorkFrame にドラッグ&ドロップ等で貼り付ける。
3. 絵を右クリックしアニメーションイベントのパラメータを設定する。
4. 細かい部分は XMLText で実際のソースを修正する。

以上の手順で自動作成した Cell Animator の XML を図 12 に、アニメーションを図 13 に示す。

手作業で作成した XML と比較すると、画像のパスが絶対パスである事や、自動生成された id 名の違い、イメージの座標、インデントされていない等の細かい部分の違いがあるもののその他文法は忠実に再現できていることが分かる。

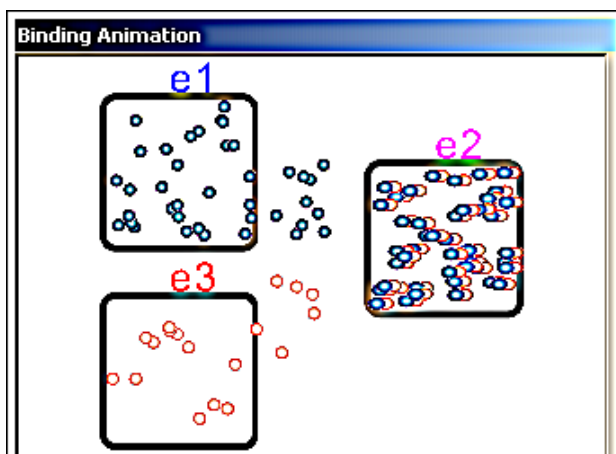


図 13 CI Builder で作成した複合体形成動画

4. 被験者による利用性能評価実験

Cell Illustrator 及び Cell Animator を利用経験のある被験者、または生命モデルに関して知識のある被験者に対し、CI Builder の評価実験を行った。利用する生命モデルはアポトーシス（管理された細胞死）とする。図 14 と図 15 に Cell Illustrator により記述したアポトーシスのモデルと Cell Animator のサンプルのアニメーションを示す。評価方法は被験者がモデルを完成するまでの時間とする。従来の手法である XML を手作業で作成する場合と、提案手法の CI Builder を用いた場合で作成時間をどれだけ短縮できるか比較した。Cell Illustrator のシミュレーション結果の CSV ファイルは事前に準備しておく。

4.1 XML を手作業で作成した場合

Cell Animator のタグファイルマニュアルとアニメーションサンプルを渡し、XML の作成を行ってもらった。質問を受けた場合は、詳細な説明を行った。

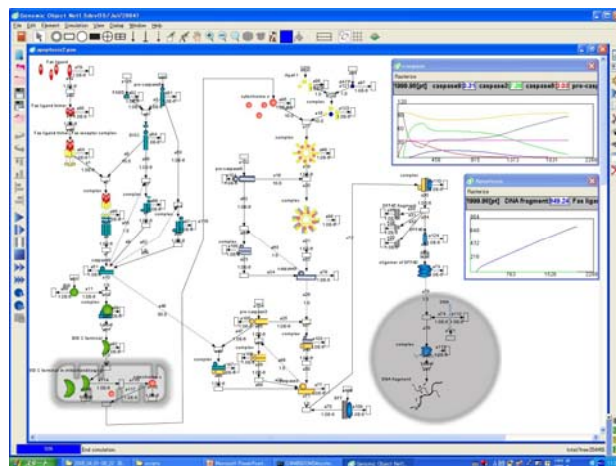


図 14 アポトーシスモデル

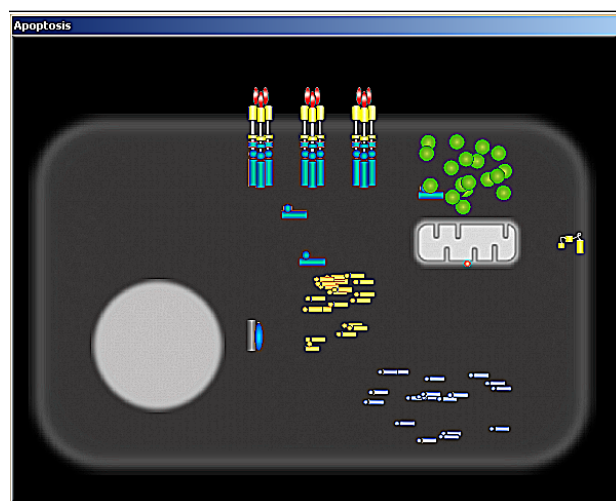


図 15 アポトーシスの動画

またどうしても上手くいかないエラーに関しては、補足を行った。

4.2 CI Builder を用いて XML を自動生成した場合

CI Builder の起動方法、利用法を纏めたマニュアルを作成し、CI Builder を操作して XML を作成してもらった。手作業の場合と同様に、質問やエラーに関する補足を行った。

4.3 考察

被験者が作成方法を理解し、目的の XML を記述するまでに約 3 日かかったのに対して、CI Builder を用いると半日程度に短縮することができた。

手作業で一番長くかかった作業が、文法ミスやプロパティの値の不適合によるデバッグの処理だったのに対して、CI Builder の作業ではこのような人為的ミスを自動化により無くすことができた為、作業時間を短縮することができたと推測される。

しかし CI Builder の GUI や機能は、Homepage Builder のような HTML 作成ツールに比べれば、未完

成な部分や改善点が多い上,ほとんどユーザーの意見等を取り入れて作成されていないため被験者が戸惑っている点がたくさん見られた.改善点に対する意見を得る事ができた.CI Builder を標準システムとするためには,エディタの機能(編集状態の保存,画像の大きさの変更等)を有していたり,多くの被験者の意見を反映する必要があることが今後の課題として残った.被験者による利用性能評価実験によりCI Builder が Cell Animator のアニメーションを作成するに当たって「求められているツール」であることを認識できた.

5. 結言

従来,Cell Illustrator のシミュレーション結果の可視化ツールである Cell Animator を利用するには,手作業でアニメーションの XML を作成しなければならず,プログラミングに明るくない研究者たちは困難な作業であった.そこで,Cell Animator のアニメーション定義をする XML を,GUI を用いた容易な操作で作成できるツール,CI Builder の開発を行った.そして手作業で XML を作成する場合と CI Builder で作成する場合の作成効率の比較検討を行った.比較には実際のユーザーに近い人に,手作業での作成と,CI Builder での自動作成をしてもらい,検討を行った.

その結果,CI Builder を用いれば Cell Animator のアニメーションを作成するのに費やしていた時間と負担をかなり軽減することが確認できた.またこのツールを用いることで生命モデルの分析や検討の効率化が図れることが確認できた.

参考文献

- [1] M.Nagasaki, A.Doi, H.Matsuno, S.Miyano, Genomic Object Net: I. A platform for modeling and simulating biopathways, Appl. Bioinformatics, 2:181-184, 2004.
- [2] H.Matsuno, Y.Tanaka, H.Aoshima, A.Doi, S.Miyano, Biopathways representation and simulation on hybrid functional Petri net, In Silico Biology, 3(3):389-404, 2003.
- [3] A.Doi, M.Nagasaki, S.Fujita, H.Matsuno, S.Miyano, Genomic Object Net: II. Modeling biopathways by hybrid functional Petri net with extension, Appl. Bioinformatics, 2:185-188, 2004.
- [4] 原田 勝通, 生命パスウェイシミュレーション

視覚化支援ツールの開発, 山口大学理学部, 2005, CellAnimator タグファイルマニュアル, 株式会社パイケーキ

- [5] Cell Illustrator Projects :
<http://www.genomicobject.net/>