

GON と SBML における XML パスウェイ形式の対応

北風裕教* 松野浩嗣** 宮野悟***

XML Pathway Format Comparison between Genomic Object Net and Systems Biology Markup Language Level 2

Hironori KITAKAZE*, Hiroshi MATSUNO** and Satoru MIYANO***

Abstract

In this paper, two formats of XML description in Genomic Object Net (GON) and System Biology Markup Language Level 2 (SBML Level 2) are compared. First, we discuss about the difference in representations of biological pathways with these two methods. Second, we compare these two formats step-by-step from small models to large models in the following order: (1) initial stage, (2) arrange one continuous place, (3) control variables, (4) arrange one continuous transition, (5) add arc from place to transition, and (6) several connections between elements. Based on these comparisons, we considered about problems which will be arise in performing conversion between these two formats, and discuss the method to improve these problems.

Key words : Genomic Object Net, SBML Level2, Conversion, XML, pathway

1 緒言

生命現象のモデルをコンピュータ上に構築することでシミュレーション分析を可能にするバイオシミュレーションツールの開発が、生命現象の解明や新薬開発など新たな発見の手助けになるという理由から世界中で期待されている。これらの開発者は、シミュレーションの分析方法や保存形式において各々独自のモチベーションに基づくツールを開発し、更に利用者の目的に応じてツールを改良してきた。しかしこの独自性が情報資源の共有化を阻み、連携作業を困難にしてきた。ここ最近になり、ERATO 北野共生システムプロジェクトが System Biology Markup Language (SBML) と呼ばれる中立交換フォーマットの規格化を行い、各システムの保存形式として SBML を取り入れることで相互に互換性を持たせることを提唱した [1, 2]。情報の共有化を望むユーザが増加したことから、SBML は現在多くのシステムで採用されつつある (図 1) [4, 5]。

我々の研究グループは、代謝経路、シグナル伝達、遺伝子制御ネットワークなどの生命パスウェイ (経路図) をペトリネット (PN) を用いて計算機上にモデリング・シミュレーションできる Genomic object Net (GON) と呼ばれるバイオシミュレーションツールの開発を行っている [3]。GON において SBML の両方向 (GON から SBML, SBML から GON) の変換が行えるようになれば、これまで他のツールを用いて研究され、データベース化された世界中の生命パスウェイを GON で利用可能となり、また逆に GON で作成したパスウェイを他のツールで利用することが可能となる。GON は分析において他の多くのツールで用いられるような微分方程式や運動方程式の解法を必要とせず、莫大な計算を必要としないため、計算機の負荷を非常に抑えられる利点がある。また、数学やプログラムの知識に明るくない医者や薬学者においても、生命ネットワークの作成を GUI 環境により効率良く行える。これらの GON の利点と他のツールの利点を組み合わせて、複合的に利用することも可能になる。これらのことから、現在 GON においても SBML の両方向の変換が強く要求されるようになり、最新版に対応した双方向に変換可能なシステムを新たに構築する必要が生じた。

したがって本研究では、GON と SBML の保存形式を比較検討し、相互変換を可能にするための対応について検討を行う。比較は、初期状態、GON の連続プレースを 1 つ配置した場合、変数値を変化した場合、GON の連続トランジションを 1 つ配置した場合、GON のプレースからトランジションへのアークを追加した場合、複数の連結がある場合と SBML Level2 との XML 定義の違いについて段階的に比較を行う。これらの対応から両方向への変換システム開発で起こりうる問題点について考慮し、問題点の改善方法について検討を行う。

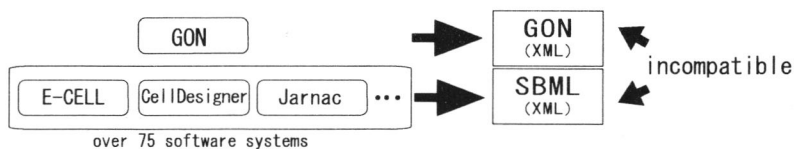


図 1: SBML と各ツールの関係

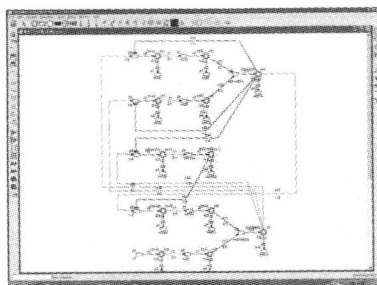


図 2: GON で表現したサーカディアンモデル

	discrete elements	continuous elements
place	○	◎
transition	■	□
token	integer	real number
arc	→ normal	→○ inhibitor

図 3: HPN の基本構成要素

2 GON と SBML との対応

GON と SBML は共に文書やデータの意味や構造を記述するためのマークアップ言語の 1 つである eXtensible Markup Language(XML) を用いて保存形式が記述されるが、タグの記述定義が異なるため互換性はない。表 1 に GON と SBML の保存形式 (タグ) の違いについて例を示す。この章では、これらの各々の生物事象の表現方法について明確にし、それに対応するタグの記述方法について比較し検討する。

表 1: GON と SBML Level2 の保存形式 (タグ) の違いの例

GON	SBML Level2
<pre><?xml version="1.0" encoding="UTF-8"?> <ci:model xmlns:ci="http://www.GenomicObject.net/2002/PetriNet"> <ci:net> <ci:place label="e1" name="e1" type="continuous"> <ci:parameter label="m1" initialValue="0.0" minimumValue="0.0" maximumValue="infinite"/> <ci:graphics> <ci:figure> <ci:continuousPlace location="115.0 101.0"/> </ci:figure> </ci:graphics> </ci:place> </ci:net> <ci:simulation enhancedFiring="true" samplingInterval="0.1" simulationTime="2000.0" logUpdateInterval="1.0" plotUpdateInterval="1.0"/> </ci:model></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1"> <model id="Branch"> <listOfCompartments> <compartment id="compartmentOne" size="1"/> </listOfCompartments> <listOfSpecies> <species id="S1" initialConcentration="0" compartment="compartmentOne" boundaryCondition="false"/> </listOfSpecies> </model> </sbml></pre>

2.1 GON : HFPN による生命パスウェイの実現

図 2 に GON を用いて表現した生命パスウェイの例 (サーカディアンモデル) を示す。GON は、生物現象の分析を容易にするために離散値と連続値を同時に扱うことのできるハイブリッド PN(HPN) を、独自に改良し拡張したハイブリッド関数 PN (HFPN) を基本アーキテクチャに持ち、計算機上に生命パスウェイをモデリングすることでシミュレーション分析を可能にすることができるシステムである。以下に HPN から HFPN への拡張について説明する。

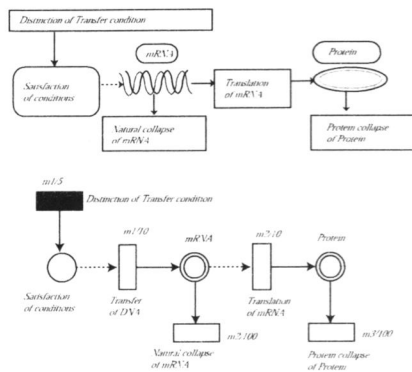


図 4: 転写と翻訳

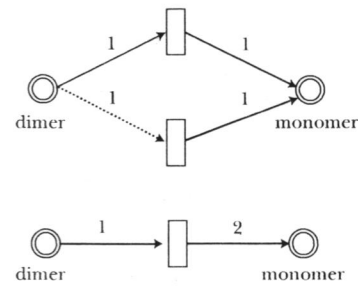


図 5: 2量体から1量体への分解 (上:HPN, 下:HFPN)

2.1.1 HPN の基本構成

HPN 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、トークン、アークの4つの要素から構成される (図3)。トークンの値に対して離散値のみを許す離散PNと連続値を許す連続PNの両方を取り扱うことができ、以下のように生命事象を表現することができる。

- たんぱく質や mRNA など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・崩壊など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。

HPN を用いれば生物事象の模式図に沿って生命パスイを実現できる。図4に、DNA の転写からたんぱく質の合成までの生命事象モデルを HPN を用いて表現した例を示す。

2.1.2 HFPN への拡張

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。よって我々はHPNを独自に改良し、トランジションの発火速度をアークごとに定義付け可能にしたハイブリッド関数PN (HFPN) を生命パスイの表記に利用する。図5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1つ記述するのみで生命パスイを表現することを可能とする。

2.2 SBML: 生命パスイの実現

Systems Biology Markup Language の略である。SBML は XML を記述メカニズムにもつ生物現象表記言語である。SBML は ERATO 北野共生システムプロジェクトによってバイオシミュレーションツールのための世界共通の保存形式として提唱された。また、SBML のバージョンを Level と呼び、現在 SBML Level2 が最新である。

SBML は内部に list of Compartments, list of Species, list of Reactions など幾つかのコンポーネントを持つ。list of Compartments は物質のための有限の入れ物を表現するために用いられる。list of Species は反応に参加する物質を表現するために用いられる。例えば、簡単なイオン (プロテイン、カルシウム) や簡単な分子 (グルコース、ATP)、そして大きな分子 (RNA、プロテイン、polysaccharidies) などに相当する。list of Reactions は運搬や

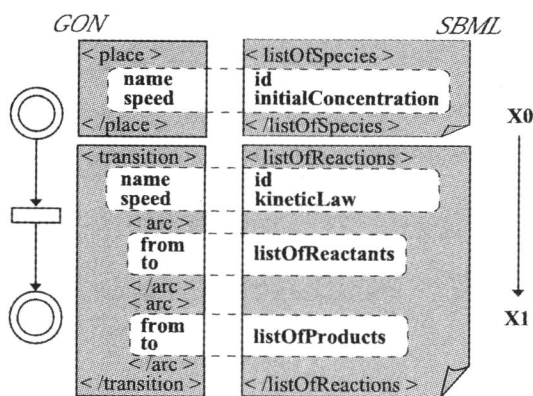


図 6: GON と SBML の対応

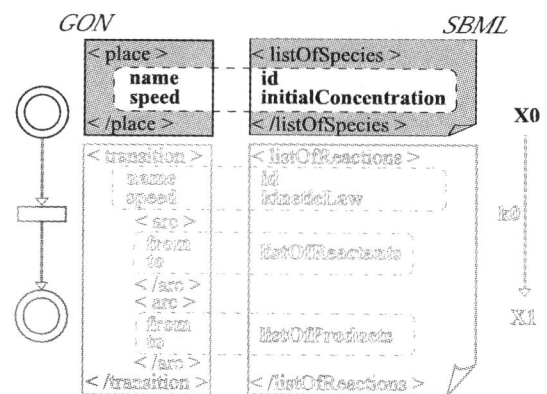


図 7: GON と SBML の対応 (プレースの場合)

種の量を変更することができるプロセスを表すために用いられる。SBML はこれらの XML 構成要素により生物現象を表現する。

現在 75 近くのプロジェクトが SBML の対応に取り組んでおり、共有化が実現しつつある。

3 GON と SBML の対応

GON と SBML の XML の両方向の変換を考えるために、GON から SBML への変換を順方向の変換とし、SBML から GON への変換を逆変換と定義する。GON と SBML の XML は非常に良く似た構造を有しているが、全ての情報を 1 対 1 で定義することはできない。GON の XML 表現は SBML の XML 表現よりも複雑であり、全ての情報を表現することはできず、SBML は GON の部分集合として考えられる。よって順方向の変換においては多くの情報を削除する必要があり、また逆方法の変換では多くの情報を補わなければならない。

図 6 に、GON の XML 保存形式と SBML Level2 の XML 保存形式の対応図を示す。大別すると GON の <place> は、SBML の <listOfSpecies> に対応し、GON の <transition> と <arc> は、SBML の <listOfReactions> に対応する。各々について、部分的に詳しく説明する。

3.1 プレースとの対応

図 7 に GON と SBML の対応 (プレースの場合) について示す。GON の XML において <place> は、place のラベル (label), place の名前 (name), place の型 (type), トークン情報の parameter のラベル (label), parameter の初期値 (initialValue), parameter の最小値 (minimumValue), parameter の最大値 (maximumValue), 図の位置情報 (location) が定義される。この中で SBML に対応するのが、<place> の name と initialValue であり、各々 <listOfSpecies> の id と initialConcentration に対応する。ここで、<place> の name と <listOfSpecies> の id は物質の名前であり、<place> の initialValue と <listOfSpecies> の initialConcentration では、物質の初期量に相当する。この定義以外のパラメータについては、SBML において注釈行 (notes, annotation) に記述する。表 2 に <place> と <listOfSpecies> の例を示す。

3.2 トランジションとの対応

図 8 に GON と SBML の対応 (トランジションの場合) を示す。GON の XML において <transition> は、transition のラベル (label), transition の名前 (name), transition の型 (type), 優先値 (priority value), 速度値 (speed value), transition の位置情報 (location) が定義される。この中で SBML に対応するのが、<transition> の name と speed value であり、各々 <listOfReaction> の id と kineticLaw に対応する。ここで、<transition> の name と <listOfReaction> の id はトランジションの名前であり、<transition> の speed value と kineticLaw は、物質の発火速度に相当する。SBML に速度情報を記述する際は、WWW Consortium(w3c) によって提唱された数式を XML 形式で記述するための形式である MathML の定義に従って行わなければならない。したがって、保存形式の変換を行う際に数式の変換を行う必要がある。この定義以外のパラメータについては、SBML において注釈行 (notes, annotation) に記述する。表 3 に <transition> と <listOfReaction> の例を示す。

表 2: <place> と <listOfSpecies>:initialConcentration のタグの例

GON	SBML Level2
<pre> ... <ci:place label="e1" name="e1" type="continuous"> <ci:parameter label="m1" initialValue="0.0" minimumValue="0.0" maximumValue="infinite"/> <ci:graphics> <ci:figure> <ci:continuousPlace location="157.0 129.0"/> </ci:figure> </ci:graphics> </ci:place> ... </pre>	<pre> ... <listOfSpecies> <species id="e1" initialConcentration="10" compartment="compartmentOne" boundaryCondition="false"/> </listOfSpecies> ... </pre>

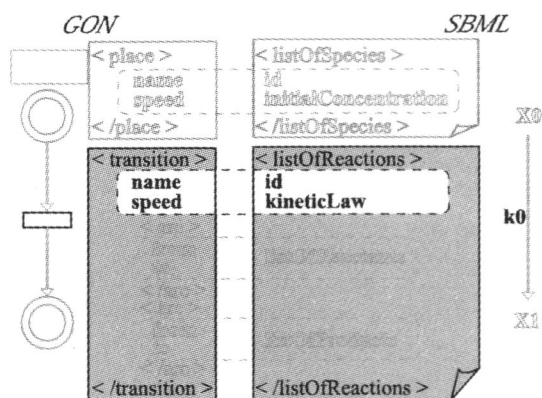


図 8: GON と SBML の対応 (トランジションの場合)

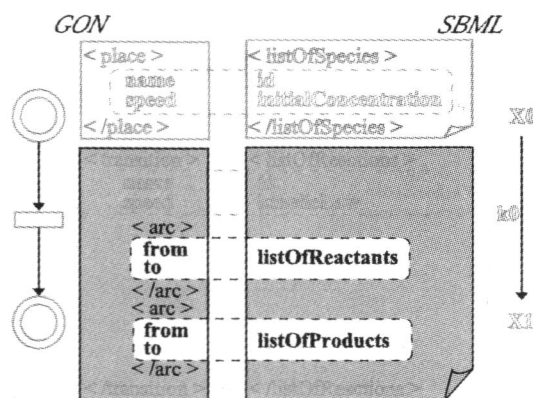


図 9: GON と SBML の対応 (アークの場合)

3.3 アークの対応

図 9 に GON と SBML の対応 (トランジションの場合) を示す。GON の XML において <arc> は、arc のラベル (label), arc の名前 (name), arc の型 (type), arc の連結元 (from), arc の連結先 (to), スピード (speed), 重み (weight), および位置情報 (Arc points) などからなる。この中で、SBML に対応するのが、<arc> の from, to であり、各々 <listOfReactions> の id, listOfReactants, listOfProducts である。ここで <arc> の from, to と <listOfReactions> の listOfReactants, listOfProducts は物質が進む方向に相当する。これ以外のパラメータについては、SBML の注釈行に記述する。表 4 に <transition> と <listOfReaction> の例を示す。

4 GON と SBML Level2 の保存形式の比較

GON と SBML Level2 の保存形式の比較を、初期状態、連続プレースを 1 つ配置した場合、連続トランジションを 1 つ配置した場合、プレースとトランジションをアークで結んだ場合、そしてこれらの組み合わせと、段階を追って行く。これにより、SBML Level2 の XML 記述方法および変換方法を示す。GON は HPN のアーキテクチャを有することから離散 PN の記述も可能であるが、SBML において離散概念が存在しないので、離散 PN については以降記述を省略する。

4.1 初期状態の場合

表 5 は GON と SBML の初期状態における XML である。GON や SBML に要素を追加すると、GON 側は ci:net 内、SBML 側は model 内に XML タグが追加される。以下に GON の要素を追加した場合における各々の XML の記述を示す。

4.2 連続プレースを 1 つ配置

表 6 は GON において図 10 で示される連続プレースをひとつ配置したときの XML と、それに対応して species をひとつ配置した SBML の記述を示したものである。GON で連続プレースをひとつ配置した場合、

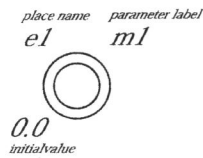


図 10: 連続ブレース

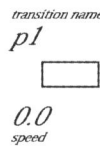


図 11: 連続トランジション

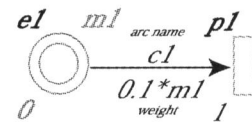


図 12: アーク

表 3: <transition> と <listOfReactions>:kineticLaw のタグの例

GON	SBML Level2
<pre> ... <ci:transition label="p1" name="p1" type="continuous"> <ci:firingCondition> <ci:priority value="0"/> <ci:speed value="0.0"/> </ci:firingCondition> <ci:function/> <ci:graphics> <ci:figure> <ci:continuousTransition location="227.0 162.0"/> </ci:figure> </ci:graphics> </ci:transition> ... </pre>	<pre> ... <listOfReactions> <reaction id="p1" reversible="false"> <kineticLaw> <math xmlns="http://www.w3.org/1998/Math/MathML"> <apply> <cn>1.0</cn> </apply> </math> </kineticLaw> </reaction> </listOfReactions> ... </pre>

```

<ci:place label="e1" name="e1" type="continuous">
...
</ci:place>

```

で示されるタグが初期状態 (表 5) に追加され, また SBML においても

```

<listOfCompartments>
<compartment id="compartmentOne" size="1"/>
</listOfCompartments>
<listOfSpecies>
<species id="e1" compartment="compartmentOne"/>
</listOfSpecies>

```

のタグが新しく追加される。place name は GON を利用するユーザがブレースを認識するための名前であり、一般的に XML 内部の認識指標としてこれを利用することはない。これは place の他に、transition や arc においても同様のことが言える。GON はブレース自体を示す place label とブレースの中に格納されている値を参照するための parameter label の 2 つの名前を持つ。SBML は 1 つの名前でこの 2 つの機能を兼ね備えている。そのため引き起こされる幾つかの問題点が存在する。これについては後述に示す。

4.3 連続ブレースの初期量を変更した場合

表 7 左に、GON の連続ブレースの初期量を 10 にしたときに出力される XML を示す。この場合、GON の initialValue の値のみが 10 に置き換えられる。また表 7 右に GON に対応した SBML を示す。初期量をあらゆる initialConcentration の値が 10 に変更される。初期量のような変数の設定及び定義は、place のみならず transition や arc においても同様に、対応する変数の置き換えに相当するため、以降変数の記述については省略する。

4.4 連続トランジションを 1 つ配置した場合

表 8 は GON において図 11 で示される連続トランジションをひとつ配置したときの XML と、それに対応した SBML の記述を示したものである。GON で連続トランジションを 1 つ配置した場合は初期状態 (表 5) の XML に対して以下のタグの追加が行われる。

```

<ci:transition label="p1" name="p1" type="continuous">
...
</ci:transition>

```

表 4: <arc> と <listOfReactions>:listOfReactants のタグの例

GON	SBML Level2
<pre> ... <ci:transition label="p1" name="p1" type="continuous"> ... <ci:arc label="c1" name="c1" type="normal" from="e1" to="p1"> <ci:speed value="1.0"/> <ci:weight value="0.0"/> <ci:updater type="Script" value=""/> <ci:graphics> <ci:figure> <ci:normalArc points="151.0 180.0 219.06836563565966 171.0436361005711"/> </ci:figure> <ci:targetArrow> <ci:normalArcArrow points="0.0 0.0 0.0 6.0 8.0 3.0" pointsInAbsoluteCoordinate="218.6770020979455 168.06927321394346 219.4597291733738 174.01799898719872 227.0 170.0"/> </ci:targetArrow> </ci:graphics> </ci:arc> <ci:arc label="c2" name="c2" type="normal" from="p1" to="e2"> <ci:speed value="1.0"/> <ci:weight value="0.0"/> <ci:updater type="Script" value=""/> <ci:graphics> <ci:figure> <ci:normalArc points="249.0 169.0 331.0701767945389 179.94269023927185"/> </ci:figure> <ci:targetArrow> <ci:normalArcArrow points="0.0 0.0 0.0 6.0 8.0 3.0" pointsInAbsoluteCoordinate="331.46666795481195 176.96900653722395 330.67368563426584 182.91637394131976 339.0 181.0"/> </ci:targetArrow> </ci:graphics> </ci:arc> ... </ci:transition> ... </pre>	<pre> ... <listOfReactions> <reaction id="p1" reversible="false"> <listOfReactants> <speciesReference species="e1"/> </listOfReactants> <listOfProducts> <speciesReference species="e2"/> </listOfProducts> <kineticLaw> ... </kineticLaw> </reaction> </listOfReactions> ... </pre>

また SBML においては、次の XML タグの追加が行われる。

```

<listOfReactions>
<reaction id="p1" reversible="false">
<kineticLaw>
...
</kineticLaw>
</reaction>
</listOfReactions>

```

4.5 プレースからトランジションへのアークを追加した場合 (トランジションの速度変更あり)

表 9 左は、GON において図 12 で示される連続プレースから連続トランジションへ連結するアークを追加したときの XML である。また、表 9 右は GON に対応した SBML の XML である。GON はプレースからトランジションの間またはトランジションからプレースの間に必ずアークが接続される。したがって、arc のみを配置することはできない。GON はアークを追加することによって、表 6 左と表 8 左に反映された記述の組み合わせに

```

<ci:function>
  <ci:arc label="c1" name="c1" type="normal" from="e1" to="p1">
...
  </ci:arc>
</ci:function>

```

が新たに追加される。SBML 側も同様に、表 6 右と表 8 右に反映された記述の組み合わせに

表 5: 初期状態

GON	SBML
<pre><?xml version="1.0" encoding="UTF-8"?> <ci:model xmlns:ci="http://www.GenomicObject.net/2002/PetriNet"> <ci:net/> <ci:simulation enhancedFiring="true" samplingInterval="0.1" simulationTime="2000.0" logUpdateInterval="1.0" plotUpdateInterval="1.0"/> </ci:model></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1"> <model/> </sbml></pre>

表 6: 連続プレースを1つ配置

GON	SBML
<pre><?xml version="1.0" encoding="UTF-8"?> <ci:model xmlns:ci="http://www.GenomicObject.net/2002/PetriNet"> <ci:net> <ci:place label="e1" name="e1" type="continuous"> <ci:parameter label="m1" initialValue="0.0" minimumValue="0.0" maximumValue="infinite"/> <ci:graphics> <ci:figure> <ci:continuousPlace location="129.0 171.0"/> </ci:figure> </ci:graphics> </ci:place> </ci:net> <ci:simulation enhancedFiring="true" samplingInterval="0.1" simulationTime="2000.0" logUpdateInterval="1.0" plotUpdateInterval="1.0"/> </ci:model></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1"> <model id="Sample"> <listOfCompartments> <compartment id="compartmentOne" size="1"/> </listOfCompartments> <listOfSpecies> <species id="e1" compartment="compartmentOne"/> </listOfSpecies> </model> </sbml></pre>

```
</listOfReactants>
<speciesReference species="e1"/>
</listOfReactants>
```

が追加される。

また表 9 は、arc の追加以外に、トランジションの速度変更を記述している。表 9 左は、GON のトランジションの速度を $0.1 \times m1$ に変更し出力された XML である。また表 9 右は GON に対応して SBML の速度情報を変更したものである。GON の transition speed の記述が変更され

```
<ci:speed value="1.0"/>
<ci:weight>
<ci:script><![CDATA[0.1*m1]]>
</ci:script>
</ci:weight>
```

に置き換わっているのがわかる。SBML 側も同様に

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
<apply>
<times/>
<ci>k0</ci>
<ci>e1</ci>
</apply>
```


表 7: 連続プレースの初期量を 10 にした場合

GON	SBML
<pre> <?xml version="1.0" encoding="UTF-8"?> <ci:model xmlns:ci="http://www.GenomicObject.net/2002/PetriNet"> <ci:net> <ci:place label="e1" name="e1" type="continuous"> <ci:parameter label="m1" initialValue="10.0" minimumValue="0.0" maximumValue="infinite"/> <ci:graphics> <ci:figure> <ci:continuousPlace location="129.0 171.0"/> </ci:figure> </ci:graphics> </ci:place> </ci:net> <ci:simulation enhancedFiring="true" samplingInterval="0.1" simulationTime="2000.0" logUpdateInterval="1.0" plotUpdateInterval="1.0"/> </ci:model> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1"> <model id="Sample"> <listOfCompartments> <compartment id="compartmentOne" size="1"/> </listOfCompartments> <listOfSpecies> <species id="e1" initialConcentration="10" compartment="compartmentOne"/> </listOfSpecies> </model> </sbml> </pre>

表 8: 連続トランジションを 1 つ配置した場合

GON	SBML
<pre> <?xml version="1.0" encoding="UTF-8"?> <ci:model xmlns:ci="http://www.GenomicObject.net/2002/PetriNet"> <ci:net> <ci:transition label="p1" name="p1" type="continuous"> <ci:firingCondition> <ci:priority value="0"/> <ci:speed value="0.0"/> </ci:firingCondition> <ci:function/> <ci:graphics> <ci:figure> <ci:continuousTransition location="185.0 190.0"/> </ci:figure> </ci:graphics> </ci:transition> </ci:net> <ci:simulation enhancedFiring="true" samplingInterval="0.1" simulationTime="2000.0" logUpdateInterval="1.0" plotUpdateInterval="1.0"/> </ci:model> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1"> <model id="Sample"> <listOfReactions> <reaction id="p1" reversible="false"> <kineticLaw> <math xmlns="http://www.w3.org/1998/Math/MathML"> <apply> <cn>1.0</cn> </apply> </math> </kineticLaw> </reaction> </listOfReactions> </model> </sbml> </pre>

```

</math>
<listOfParameters>
<parameter id="k0" value="0.1"/>
</listOfParameters>

```

が追加される。加わった情報を比較すると GON 側は $0.1 * m1$ という式が 1 行記述されているのに対し、SBML では MathML 形式で $k0 * e1$ という式が記述される。ここで $k0$ は定数であり、 0.1 が格納されている。先に述べたように GON のプレースには 2 つの名前が存在し、プレースに格納されている値を参照する場合、つまり速度を表す数式の中では parameter label ($m1$) を用いる。またプレース自体を示す場合、つまりアークの from, to には place label ($e1$) を用いる。しかし SBML では GON で使われるこの 2 つの変数を 1 つの変数で表さなければならないことに加えて定数を基本的に parameter ($k0$) で扱うという仕様の影響で、速度情報を比較すると数式が一致していないように見える。ここで、GON から SBML への順方向の対応では、place label のみを使うといった変数の統一が必要となり、また、SBML から GON への逆方向の対応では、新たな変数名 parameter label を定義する必要がある。

表 9: 連続トランジションの速度を 0.1*m1 に変更した場合

GON	SBML
<pre> <?xml version="1.0" encoding="UTF-8"?> <ci:model xmlns:ci="http://www.GenomicObject.net/2002/PetriNet"> <ci:net> <ci:place label="e1" name="e1" type="continuous"> <ci:parameter label="m1" initialValue="0.0" minimumValue="0.0" maximumValue="infinite"/> <ci:graphics> <ci:figure> <ci:continuousPlace location="73.0 185.0"/> </ci:figure> </ci:graphics> </ci:place> <ci:transition label="p1" name="p1" type="continuous"> <ci:firingCondition> <ci:priority value="0"/> <ci:speed value="1.0"/> </ci:firingCondition> <ci:function> <ci:arc label="c1" name="c1" type="normal" from="e1" to="p1"> <ci:speed value="1.0"/> <ci:weight> <ci:script><![CDATA[0.1* m1]]> </ci:script> </ci:weight> <ci:updater type="Script" value=""/> <ci:graphics> <ci:figure> <ci:normalArc points="95.0 196.0 177.0 196.0"/> </ci:figure> <ci:targetArrow> <ci:normalArcArrow points="0.0 0.0 0.0 6.0 8.0 3.0" pointsInAbsoluteCoordinate="177.0 193.0 177.0 199.0 185.0 196.0"/> </ci:targetArrow> </ci:graphics> </ci:arc> </ci:function> <ci:graphics> <ci:figure> <ci:continuousTransition location="185.0 190.0"/> </ci:figure> </ci:graphics> </ci:transition> </ci:net> <ci:simulation enhancedFiring="true" samplingInterval="0.1" simulationTime="2000.0" logUpdateInterval="1.0" plotUpdateInterval="1.0"/> </ci:model> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1"> <model id="sample"> <listOfCompartments> <compartment id="compartmentOne" size="1"/> </listOfCompartments> <listOfSpecies> <species id="e1" compartment="compartmentOne" initialConcentration="0"/> </listOfSpecies> <listOfReactions> <reaction id="p1" reversible="false"> <listOfReactants> <speciesReference species="e1"/> </listOfReactants> <kineticLaw> <math xmlns="http://www.w3.org/1998/Math/MathML"> <apply> <times/> <ci>k0</ci> <ci>e1</ci> </apply> </math> <listOfParameters> <parameter id="k0" value="0.1"/> </listOfParameters> </kineticLaw> </reaction> </listOfReactions> </model> </sbml> </pre>

4.6 プレース→トランジション→プレース 2 つの場合

表 10 左は GON で連続プレースから連続トランジション, そして連続プレース 2 つにそれぞれアークで結んだときの XML である. また, 表の右側は GON に対応した SBML の XML である.

GON 側はプレース, アークの数が増えたことから, XML タグが

```

<ci:place label="e3" name="e3" type="continuous">
  ...
</ci:place>
  <ci:arc label="c3" name="c3" type="normal" from="p1" to="e3">
  ...
</ci:arc>

```

のように追加されていることがわかる.

また SBML 側も同様に, species, listOfProducts が

```
<listOfProducts>
```

表 10: プレース→トランジション→プレース 2つの場合

GON	SBML
<pre> <?xml version="1.0" encoding="UTF-8"?> <ci:model xmlns:ci="http://www.GenomicObject.net/2002/PetriNet"> <ci:net> <ci:place label="e1" name="e1" type="continuous"> <ci:parameter label="m1" initialValue="10.0" minimumValue="0.0" maximumValue="infinite"/> <ci:graphics> <ci:continuousPlace location="73.0 185.0"/> </ci:figure> </ci:place> <ci:place label="e2" name="e2" type="continuous"> <ci:parameter label="m2" initialValue="0.0" minimumValue="0.0" maximumValue="infinite"/> <ci:graphics> <ci:figure> <ci:continuousPlace location="381.0 87.0"/> <ci:text name="value" location="368.5 112.0" arrangement="None" offsets="-14.0 23.0"/> </ci:figure> </ci:place> <ci:place label="e3" name="e3" type="continuous"> <ci:parameter label="m3" initialValue="0.0" minimumValue="0.0" maximumValue="infinite"/> <ci:graphics> <ci:figure> <ci:continuousPlace location="283.0 269.0" size="25.0 25.0"/> </ci:figure> </ci:place> <ci:transition label="p1" name="p1" type="continuous"> <ci:firingCondition> <ci:priority value="0"/> <ci:speed value="1.0"/> </ci:firingCondition> <ci:function> <ci:arc label="c1" name="c1" type="normal" from="e1" to="p1"> <ci:speed value="1.0"/> <ci:weight value="1.0"/> <ci:updater type="Script" value=""/> </ci:arc> <ci:graphics> <ci:figure> <ci:normalArc points="95.0 196.0 177.0 196.0"/> <ci:targetArrow> <ci:normalArcArrow points="0.0 0.0 0.0 6.0 8.0 3.0" pointsInAbsoluteCoordinates="177.0 193.0 177.0 199.0 185.0 196.0"/> </ci:targetArrow> </ci:graphics> </ci:arc> <ci:arc label="c2" name="c2" type="normal" from="p1" to="e2"> <ci:speed value="1.0"/> <ci:weight value="0.0"/> <ci:updater type="Script" value=""/> </ci:arc> <ci:graphics> <ci:figure> <ci:normalArc points="207.0 191.0 374.85276938454047 106.59403596663108"/> <ci:text name="name" location="279.92638469227023 130.29701798331553" arrangement="None" offsets="-3.5 -9.5"/> <ci:targetArrow> <ci:normalArcArrow points="0.0 0.0 0.0 6.0 8.0 3.0" pointsInAbsoluteCoordinates="373.5050058970538 103.91382448583376 376.2005328720271 109.27424744742841 382.0 103.0"/> </ci:targetArrow> </ci:graphics> </ci:arc> <ci:arc label="c3" name="c3" type="normal" from="p1" to="e3"> <ci:speed value="1.0"/> <ci:weight value="0.0"/> <ci:updater type="Script" value=""/> </ci:arc> <ci:graphics> <ci:figure> <ci:normalArc points="203.0 202.0 279.92078003856744 267.7997034064854"/> <ci:text name="name" location="233.96039001928372 214.8998517032427" arrangement="None" offsets="0.0 -11.0"/> <ci:targetArrow> <ci:normalArcArrow points="0.0 0.0 0.0 6.0 8.0 3.0" pointsInAbsoluteCoordinates="281.8708912611354 265.51999592094813 277.97066881599943 270.07941089202257 286.0 272.99999999999994"/> </ci:targetArrow> </ci:graphics> </ci:arc> </ci:transition> </ci:net> <ci:simulation enhancedFiring="true" samplingInterval="0.1" simulationTime="2000.0" logUpdateInterval="1.0" plotUpdateInterval="1.0"/> </ci:model> </pre>	<pre> <?xml version="1.0" encoding="UTF-8"?> <sbml xmlns="http://www.sbml.org/sbml/level2" level="2" version="1"> <model id="sample"> <listOfCompartments> <compartment id="compartmentOne" size="1"/> </listOfCompartments> <listOfSpecies> <species id="e1" compartment="compartmentOne" initialConcentration="0"/> <species id="e2" compartment="compartmentOne" initialConcentration="0"/> <species id="e3" compartment="compartmentOne" initialConcentration="0"/> </listOfSpecies> <listOfReactions> <reaction id="p1" reversible="false"> <listOfReactants> <speciesReference species="e1"/> </listOfReactants> <listOfProducts> <speciesReference species="e2"/> <speciesReference species="e3"/> </listOfProducts> <kineticLaw> <math xmlns="http://www.w3.org/1998/Math/MathML"> <apply> <cn>1.0</cn> </apply> </math> </kineticLaw> </reaction> </listOfReactions> </model> </sbml> </pre>

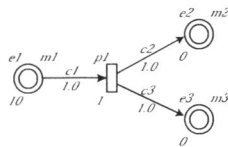


図 13: SBML と各ツールの関係

```
<speciesReference species="e2"/>
<speciesReference species="e3"/>
</listOfProducts>
```

のようにそれぞれ追加されていることがわかる。このように GON で作成した生物パスウェイの HFPN 記述が拡張された場合においても、XML のタグ記述が単純に組み合わせられることで簡単に表現することができる。

5 結言

GON と SBML の保存形式を比較し、相互変換を可能にするための対応について検討を行った。GON から SBML の変換を順方向の変換、SBML から GON への変換を逆変換として定義した。比較は、初期状態、GON の連続プレースを 1 つ配置した場合、変数値を変化した場合、GON の連続トランジションを 1 つ配置した場合、GON のプレースからトランジションへのアークを追加した場合、複数の連結がある場合など、GON と SBML Level2 との XML 定義の違いについて段階的に検討を行った。

GON から SBML への順方向の変換に対しては、プレース・トランジション・アーク・トークン等の位置情報、グラフ定義など多くのパラメータが、SBML では記述することができず、SBML の注意書き部 (notes, annotation) にこれらの未使用パラメータを保存する必要があることが分かった。SBML Level2 では注意書きの記述方法にはルールがなく各ツールごとに独自の記述方法を用いて定義しているため、他ツール間のやり取りが困難となり SBML の本来の目的を達成していない。そのため、他のツールを考慮しながら注意書き部もなるべく他ツールにあわせた記述をとるようにしていく必要があるといえる。その他の問題として、SBML は Species 1 つに対して 1 つの名前が定義される。それに対応し、GON のプレースにおいては place label と place name の 2 つの変数名を持つ。トランジションの速度設定においては、place label を利用し、アークの結合情報を設定する際は place name を利用することが GON では定義されているので、今後相互変換を可能にするためには、名前の分割を正しく行うように注意する必要がある。

また、SBML から GON への逆変換に対しては、SBML の注釈に何も記述がない場合、GON のパラメータの必要条件を全て定義することはできないことが明らかとなった。特に、GON は生命パスウェイを GUI を用いて容易に作成することができるが、パスウェイの連結は正しく行われていても、位置情報がないことにより、見栄えの悪い状態で出力されることは致命的であると考えられる。この問題を解決するためには、位置情報を含め、初期パラメータや要素の型などの多くの情報を補う必要が生じるといえる。

今後は SBML のバージョンの向上により、これらのパラメータ値 SBML が記述可能になることを期待する。さらに、GON から SBML Level2 と SBML Level2 から GON への両方向への変換が可能なツールの開発を行い、現在の GON システムの標準システムに導入したい。

参考文献

- [1] Andrew Finney and Michael Hucka, Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions, 28 June 2003.
- [2] <http://sbml.org/>
- [3] <http://www.GenomicObject.Net/>
- [4] <http://www.e-cell.org/>
- [5] <http://www.gepasi.org/>