

膜人工化学に基づくネットワークモデルの構築

茂 呂 智 大^{†1} 富 永 和 人^{†2}

人工化学は人工生命の一研究手法であり、化学反応系を抽象的に表現する手段である。我々は膜の概念を取り入れた独自の膜人工化学を提案している。膜人工化学を用いることで、複数の領域における化学反応によって成り立つ生命現象や、領域の構造変化を伴う生命現象を表現できる。この膜人工化学を応用することで、生物に見られるような相互作用をネットワークモデルに取り入れられる可能性がある。本研究では膜人工化学に基づくネットワークモデルを構築し、その動作を実現するソフトウェアを実装した。このソフトウェアに計算主体の振舞いを与え、期待通りに動作することを確認した。

A network model based on an artificial chemistry with membranes

TOMOHIRO MORO^{†1} and KAZUTO TOMINAGA^{†2}

Artificial chemistries are formalisms that describe virtual chemical systems. They are used in the research field of artificial life. We propose an artificial chemistry that can deal with membranes. It models a system of nested membranes like a living cell, and therefore can describe life phenomena comprising chemical reactions in multiple compartments and/or structural changes of the system of membranes. We speculate that a network model incorporating interactions observed in living systems can be build using the artificial chemistry. In this study, we designed a network model based on the artificial chemistry, and developed a software system to execute the model. We confirmed the system operates as expected with a preliminary example.

^{†1} 東京工科大学大学院 バイオ・情報メディア研究科

Graduate School of Bionics, Computer and Media Sciences, Tokyo University of Technology

^{†2} 東京工科大学 コンピュータサイエンス学部

School of Computer Science, Tokyo University of Technology

1. はじめに

人工化学は人工生命の一研究手法であり、化学反応系を抽象的に表現する手段である¹⁾。人工化学の目的のひとつは生命現象の表現である。これまでに様々な人工化学が提案されてきた。我々の研究室では、膜の概念を取り入れた独自の膜人工化学を提案している²⁾。膜人工化学は、複数の領域における化学反応によって成り立つ生命現象や、領域の構造変化を伴う生命現象の表現が可能である。

本研究では、ネットワーク上の計算機間を移動する計算主体を生物に見立て、複数の計算主体が生物に見られるような相互作用を行うネットワークモデルを提案する。このネットワークモデルでは、ネットワーク上に存在する情報を仮想的な分子に変換し、この分子間の化学反応によってその情報を処理する。計算主体の振舞いを膜人工化学の化学反応で定義する。計算主体は情報を表す分子を反応させながら、ネットワーク上の計算機間を移動していく。

本研究では、この膜人工化学に基づくネットワークモデルを構築し、その応用を考える。

2. 膜人工化学

膜人工化学は、膜構造とその構造の動的変化を表現できる人工化学である²⁾。膜人工化学を用いることで、複数の領域における化学反応によって成り立つ生命現象や、領域の構造変化を伴う生命現象を表現できる。この節ではこの膜人工化学を簡単に説明する。

2.1 原体と分体

自然界における原子にあたるものをこの人工化学においては原体と呼ぶ。原体は、英大文字ではじまり、それに 0 個以上の英小文字や数字が続く文字列で表現する。

自然界における分子にあたるものを分体と呼ぶ。分体は図 1 のようにひとつ以上の原体列を積み重ねたものである。原体列は 0 個以上の原体の列である。分体が持つ各原体列には整数で表す開始位置があり、この開始位置をずれと呼ぶ。ずれは最初の列の開始位置を基準とする。最初の列のずれを 0 とする。ずれの尺度はひとつの原体を 1 とする。図 1 の分体はふたつの原体列を積み重ねたものである。この分体は 0#ABC/1#DEF/ と表す。この分体の 2 行目の原体列は、1 行目の原体列と比べて原体ひとつ分右にずれている。分体は、その分体が持つ原体列とそのずれを順に並べて表記する。記号 (#) は原体列のずれと原体列の区切りを表し、記号 (/) は行の区切りを表す。

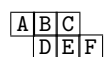


図 1 分体の例

Fig. 1 An example virtual molecule.

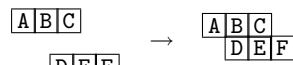


図 2 反応の例

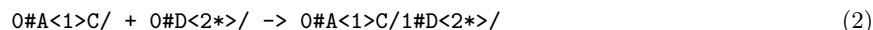
Fig. 2 An example of virtual reaction.

2.2 組み換え規則

組み換え規則とは、我々の人工化学における分体間の反応を表す式である。組み換え規則は化学反応式のように表記し、左辺の項に一致する分体群を右辺の項に一致する分体群に組み換える。例えば、図 2 のような反応を表す組み換え規則は、式 1 のように表記する。



組み換え規則の項の表記には鬼札を用いてもよい。鬼札には原体鬼札と列鬼札の 2 種類がある。原体鬼札は任意の原体ひとつと一致する鬼札である。原体鬼札は識別子となる整数を角括弧 (<>) で囲んだ文字列で表す。列鬼札は 0 個以上の原体で構成される原体列に一致する鬼札である。列鬼札は分体の行の先頭または末尾にのみ使える。行の先頭に現れる列鬼札は、識別子となる整数の前に星印 (*) をつけて <*1> のように表記する。行の末尾に現れる列鬼札は、識別子となる整数の後に星印をつけて <1*> のように表記する。鬼札を含む組み換え規則の例を式 2 に示す。



組み換え規則の左辺第 1 項 $0\#A<1>C/$ は分体 $0\#ABC/$ に一致する。また、組み換え規則の左辺第 2 項 $0\#D<2*>/$ は分体 $0\#DEF/$ に一致する。そのため、式 2 の組み換え規則が分体 $0\#ABC/$ と $0\#DEF/$ に適用されると、分体 $0\#ABC/1\#DEF/$ に組み換えられる。

2.3 小部屋と膜

自然界における細胞質にあたるものをこの人工化学においては小部屋と呼ぶ。小部屋は反応集合と組み換え規則群を持つ仮想的な空間である。小部屋が持つ反応集合中の分体は、その小部屋が持つ組み換え規則によって組み換えられる。

自然界における細胞膜にあたるものをこの人工化学においては膜と呼ぶ。膜は小部屋の包みであり、隣接する小部屋は膜によって仕切られる。膜も小部屋と同様に反応集合と組み換え規則群を持つ。膜が持つ反応集合中の分体を膜分体と呼ぶ。膜分体は膜蛋白質などをモデル化するものである。膜分体は、膜分体を持つ膜の組み換え規則やその膜が仕切っている小部屋の組み換え規則によって組み換えられる。

膜分体には甲向きと乙向きのふたつの向きがある。向きを区別するために、甲向きの膜

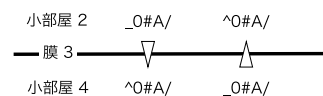


図 3 膜分体の向き

Fig. 3 Directions of membrane virtual molecules.

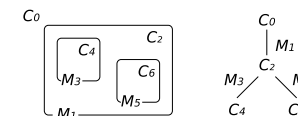


図 4 系の構造と木

Fig. 4 Structure and tree of system.

分体には山印 (^) を、乙向きの膜分体には下線 (˘) を頭につけて表記する。例えば膜分体 $\wedge 0\#A/$ と $\underline{0\#A/}$ があつたとき、前者は甲向きであり、後者は乙向きである。膜分体が小部屋から参照される場合には、その膜分体を持つ膜と小部屋の位置関係によって向きの扱いが異なる。図 3 の膜 3 にあるふたつの膜分体を小部屋 2 から参照する時は、左側の膜分体は乙向きとして扱い、右側の膜分体は甲向きとして扱う。反対に、膜 3 にあるふたつの膜分体を小部屋 4 から参照する時は、左側の膜分体は甲向きとして扱い、右側の膜分体は乙向きとして扱う。このように、小部屋から膜分体を参照する場合は相対的な向きを用いる。これは、膜蛋白質などが向きを持つためである。

2.4 系

系は小部屋とそれを包む膜の組をひとつ以上持つ仮想的な世界である。系は、大部屋と呼ばれる、膜に包まれない特別な小部屋をただひとつ持つ。系は膜を辺、小部屋を節とする木を構成する。大部屋は系が構成する木の根となる。例えば、図 4 の左側に示す系は、右側に示す木を構成する。図 4 の系における大部屋は C_0 である。小部屋 C_2 は親小部屋に C_0 を持ち、子小部屋に C_4, C_6 を持つ。膜 M_1 は小部屋 C_0 と小部屋 C_2 の仕切りであり、膜 M_3 は小部屋 C_2 と小部屋 C_4 の、膜 M_5 は小部屋 C_2 と小部屋 C_6 の仕切りである。

2.5 動作

この人工化学の系は以下の手順で非決定的に動作する。

- (1) 系の全ての小部屋や膜が持つ反応集合を初期化する。
- (2) 系が持つひとつの小部屋または膜を選択する。
- (3) 選択した小部屋または膜が持つひとつの組み換え規則を選択する。
- (4) 選択した組み換え規則がその規則の適用範囲内にある反応集合中のある分体群に適用可能な場合は適用する。
- (5) 手順 2 に戻る。

2.6 人工化学シミュレータ

人工化学系の動作を計算機上で計算するソフトウェアを人工化学シミュレータという。我々

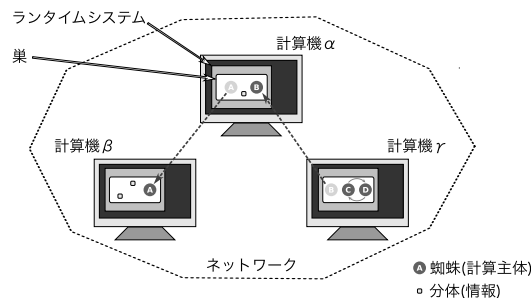


図5 蜘蛛圏の概念図
Fig.5 Basic concept of Kumonoi.

の研究室では膜人工化学のための人工化学シミュレータを開発してきた。本研究ではそのシミュレータを用いて、計算主体が行う情報の処理を実現する。

3. 膜人工化学に基づくネットワークモデル

本研究では、膜人工化学をネットワークモデルに取り入れ、複数の計算主体が生物に見られるような相互作用を行うネットワークモデル^{くものい}を構築した。蜘蛛圏の概念図を図5に示す。この節ではネットワークモデル蜘蛛圏について説明する。

3.1 蜘蛛圏の用語と概念

蜘蛛圏における用語と概念について説明する。

蜘蛛 蜘蛛は他の計算機に移動して処理を継続できる計算主体である。蜘蛛は他の蜘蛛と相互作用する。蜘蛛は自身の行動を定めるひとつの膜人工化学系を持つ。この系といくつかの情報によって蜘蛛は構成される。蜘蛛が持つ膜人工化学系やその他の情報は、蜘蛛圏の利用者が自由に定義できる。

巣 巣は蜘蛛が生息する仮想空間である。巣は蜘蛛圏のネットワークを構成する各計算機にひとつずつ存在し、膜人工化学系によって表現される。巣には0匹以上の蜘蛛が存在でき、それぞれの蜘蛛が持つ膜人工化学系を統合した系が巣となる。例えば、図6に示す巣には蜘蛛Aと蜘蛛Bが生息している。蜘蛛Aと蜘蛛Bはそれぞれ人工化学系 S_A , S_B を持つ。この時、図6の巣を表現する人工化学系 S_{nest} は図7のように S_A , S_B を取り込んだ人工化学系となる。巣には蜘蛛の他にも、様々な情報を表す分体が存在できる。

ランタイムシステム 蜘蛛圏のネットワークを構成する各計算機では、人工化学シミュ

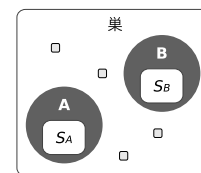


図6 巣の概念図
Fig.6 Basic concept of nest.

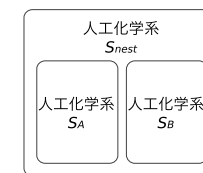


図7 巣を表現する系
Fig.7 System of nest.

レータを内蔵するランタイムシステムが動作する。ランタイムシステムはひとつの巣を管理しそれを動作させるソフトウェアである。これはまた、膜人工化学だけでは扱えない処理も担当する。具体的には、他の計算機上のランタイムシステムとの間でP2Pネットワークを構築したり、それを利用した蜘蛛の送受信を行う。

3.2 蜘蛛圏の動作

蜘蛛圏上の蜘蛛は、膜人工化学系に定義された組み換え規則に従ってそれぞれが独立して行動する。定義された組み換え規則によっては、複数の蜘蛛が相互作用する場合がある。複数の蜘蛛の行動によって、蜘蛛圏全体としての動作が形成される。

蜘蛛は図8に示すような、以下の行動ができる。

食事 蜘蛛は巣に存在する分体を自身に取り込む。蜘蛛が取り込む分体の種類は蜘蛛の種類により異なる。

分解と合成 蜘蛛は体内で自身が持つ分体を分解したり合成したりする。分体の分解や合成は、蜘蛛に与えられた膜人工化学系の組み換え規則が分体群に適用されることで行われる。

排出 蜘蛛は自身に必要なない分体を巣に排出する。蜘蛛が排出する分体の種類は蜘蛛の種類により異なる。

移動 蜘蛛は他の巣に移動することができる。蜘蛛の移動は、蜘蛛の体内に有害な物質が出現したときに起こる。有害と判断される物質は、蜘蛛の種類により異なる。蜘蛛は、有害な物質を体内に取り込んだり、体内で合成したりすると、その物質を巣に排出して他の巣に移動する。

3.3 ランタイムシステムの設計と実装

この節では蜘蛛圏のランタイムシステムの設計と実装について説明する。ランタイムシステムは巣の反応計算を行うソフトウェアである。ランタイムシステムには、巣の初期状態や生息する蜘蛛らの情報を与える。ランタイムシステムは、内蔵する人工化学シミュレータを用いて巣の動作を計算し、蜘蛛の行動を決定する。

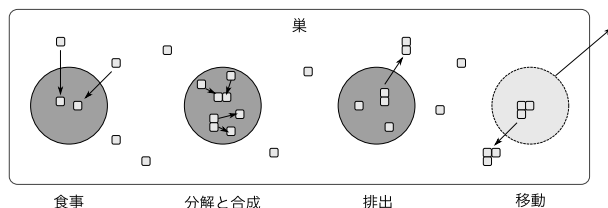


図 8 蜘蛛の行動
Fig. 8 Actions of Kumo.

ランタイムシステムの構成図を図 9 に示す。ランタイムシステムは、情報処理・変換部、人工化学シミュレータ制御部、P2P ネットワーク管理部のみつつの部分に大きく分けられる。それぞれの部分について説明する。

情報処理・変換部は、ランタイムシステムの他の部分を制御する部分である。情報処理・変換部はランタイムシステムが動作する計算機の情報を取得し、膜人工化学系で扱える形式に変換して人工化学シミュレータで計算する系に追加する。また人工化学シミュレータの状態を取得し、処理結果として出力する。

情報処理・変換部は、膜人工化学系といくつかの情報で構成される蜘蛛を管理する。蜘蛛が持つ膜人工化学系以外の情報は、必要に応じて膜人工化学系で扱える形式に変換する。具体的には、ランタイムシステムの起動時や他のランタイムシステムから蜘蛛を受信した時に、蜘蛛の食べ物や排出物、蜘蛛にとって有害な物質の情報から、それらを取り込んだり排出したりする組み換え規則を生成する。蜘蛛は基本的に、膜人工化学における小部屋と膜の一組に変換される。人工化学系の動作により、蜘蛛の体内にあたる小部屋の状況が変化することがある。その場合、情報処理・変換部は人工化学シミュレータの状態を取得して、蜘蛛の状態を確認する。その時に蜘蛛の体内に有害な物質があることを見つけると、情報処理・変換部は蜘蛛を他のランタイムシステムへ送信するように P2P ネットワーク管理部に要求する。

人工化学シミュレータ制御部は人工化学シミュレータを動作させ、膜人工化学系の動作を計算する部分である。人工化学シミュレータ制御部は、情報処理・変換部から系の操作要求や状態の取得要求を、人工化学シミュレータ操作インターフェイスを通して受け取る。要求を受け取ると人工化学シミュレータ制御部は人工化学シミュレータを動作させる。

P2P ネットワーク管理部は、他のランタイムシステムとの間で P2P ネットワークを構築する部分である。ランタイムシステムが起動すると、P2P ネットワーク管理部は、ラン

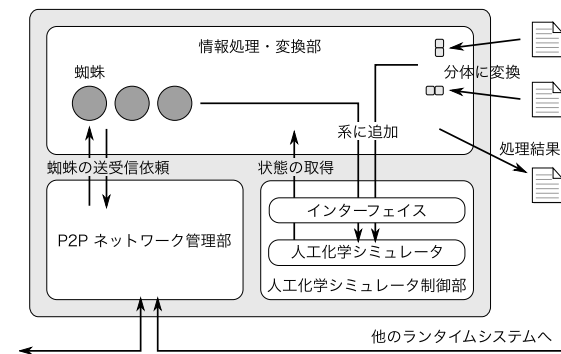


図 9 ランタイムシステムの構成
Fig. 9 Overview of Kumonoi runtime system.

タイムシステムが動作すると思われる他の計算機に対して接続を試みる。ランタイムシステムにはあらかじめ、起動時に接続を試みる対象となる計算機の一覧を与える。接続が確立されたら、蜘蛛困のネットワークに参加している他の計算機群を教えてもらい、その中からさらにいくつかの計算機に接続を試みる。以降その接続を利用して、蜘蛛困に参加している計算機群の情報をランタイムシステム間で共有する。P2P ネットワーク管理部は情報処理・変換部から蜘蛛の送信要求を受け取ると、蜘蛛困に参加している計算機群のなかから無作為にひとつの計算機を選び、その計算機で動作するランタイムシステムへ蜘蛛を送信する。また、他のランタイムシステムから蜘蛛を受信すると、情報処理・変換部に蜘蛛を渡す。

3.4 ランタイムシステムの実行例

この節では蜘蛛困のランタイムシステムの実行例を示す。ランタイムシステムをふたつ起動し、両方に以下の情報を与えて動作させた。

- 巣の分体群
 - O#A/ 100 個
 - O#B/ 100 個
 - O#X/ 50 個
- 巣の組み換え規則群
 - O#ABX/ → O#AB/ + O#X/ (3)
 - O#BAX/ → O#BA/ + O#X/ (4)
 - O#AB/ + O#BA/ → O#A/ + O#B/ + O#B/ + O#A/ (5)

表 1 ランタイムシステムに与える蜘蛛の定義
Table 1 Kumo specification.

	蜘蛛 A	蜘蛛 B
食べ物	O#A/ O#B/ O#X/ O#BA/	O#A/ O#B/ O#X/ O#AB/
排出物	O#ABX/	O#BAX/
有害な物	O#BA/	O#AB/
分体群	なし	なし
組み換え規則群	O#A/ + O#X/ -> O#AX/ O#B/ + O#X/ -> O#BX/ O#A/ + O#BX/ -> O#ABX/ O#AX/ + O#B/ -> O#ABX/	O#A/ + O#X/ -> O#AX/ O#B/ + O#X/ -> O#BX/ O#B/ + O#AX/ -> O#BAX/ O#BX/ + O#A/ -> O#BAX/

● 巣に生息する蜘蛛

表 1 に示すような蜘蛛 A 5 匹, 蜘蛛 B 5 匹

ランタイムシステムが動作すると, これらの情報をもとにして, 巣を表す膜人工化学系が生成され, 人工化学シミュレータに与えられる. 人工化学シミュレータがこの系の動作を計算し, 蜘蛛団が動作する.

今回の実行例で動作させたふたつのランタイムシステムをそれぞれランタイムシステム α , ランタイムシステム β とし, またランタイムシステム α , ランタイムシステム β が扱っている巣をそれぞれ N_α, N_β とする. この蜘蛛団の中には蜘蛛 A と蜘蛛 B がそれぞれ 10 匹ずつ存在し, 初期状態として N_α, N_β に均等に配置されている. それぞれの巣で蜘蛛らは分体を自身に取り込み, 体内で組み換える.

蜘蛛 A は体内で分体 O#ABX/ を合成すると, 巣に排出する. 反対に蜘蛛 B は体内で分体 O#BAX/ を合成すると, 巣に排出する. これらの分体に, 巣の組み換え規則 (3) や組み換え規則 (4) が適用されることによって原体 X がはずれ, 蜘蛛 A, B にとって有害な分体 O#AB/, O#BA/ が巣にできる. この分体を蜘蛛らが取り込むことで, 蜘蛛の移動が起こる. つまり蜘蛛 A は, 蜘蛛 B にとって有害な物質を, 蜘蛛 B は, 蜘蛛 A にとって有害な物質を作るのに関係していることになる. よって蜘蛛 A が多く生息する巣には蜘蛛 B が留まりにくくなり, 蜘蛛 B が多く生息する巣には蜘蛛 A が留まりにくくなるので, 同じ種類の蜘蛛が一箇所に集まるようになる. 巣の組み換え規則 (5) は分体 O#AB/, O#BA/があるとき, それぞれを分解して中和する役割を持っている. そのため, 同じ巣に蜘蛛 A と蜘蛛 B が存

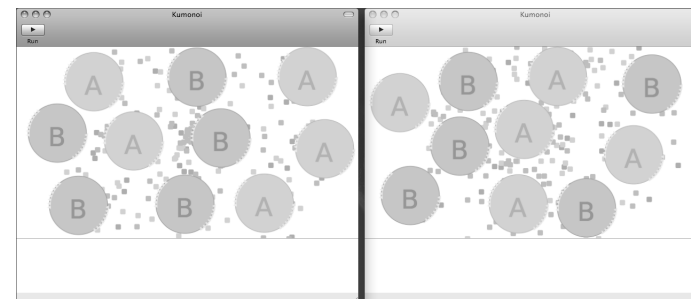


図 10 ランタイムシステムの実行例 (起動直後)

Fig. 10 Execution example of Kumonoi runtime system (initial state).

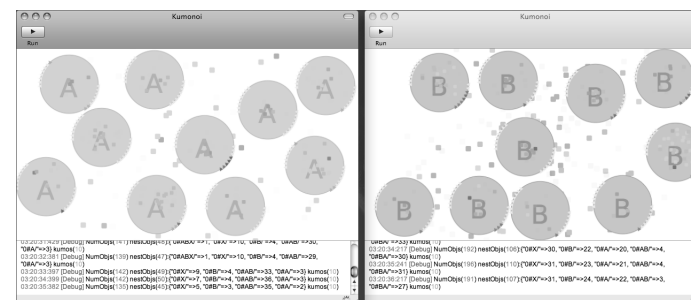


図 11 ランタイムシステムの実行例 (しばらく動作後)

Fig. 11 Execution example of Kumonoi runtime system (after a little).

在し, それぞれが互いにとって有害な物質を生成して排出したとしても, 数が多い方の蜘蛛が有利となる.

ランタイムシステムの実行例を図 10 と図 11 に示す. 図 10 はランタイムシステムの起動直後の画面であり, 図 11 はしばらく動作させた後の画面である. 起動直後は 2 種類の蜘蛛がそれぞれのランタイムシステムに同数生息していたが, しばらく動作させた後は, 同じ種類の蜘蛛がひとつのランタイムシステムに集まっている. 期待通りの結果が得られたことが分かる.

4. 議 論

本研究では、膜人工化学に基づくネットワークモデルを構築した。このネットワークモデルでは、膜人工化学系の動作によって振舞う複数の計算主体（蜘蛛）が計算機間を移動して処理を行う。

蜘蛛のように、計算機間を移動して処理を継続するものにモバイルエージェントがある。モバイルエージェントはそれぞれが各計算機で、自発的で非同期に動作する³⁾。蜘蛛困の蜘蛛も人工化学系という環境のなかで、自発的で非同期に動作していると捉えることができる。蜘蛛は環境にある情報を選択して自身に取り込む。その取り込んだ情報を自身の中で組み換えることが処理となる。これらの動作はそれぞれの蜘蛛で非同期に行われる。

モバイルエージェントシステムのエージェントの振舞いはプログラムコードで定義する。それに対して、蜘蛛の振舞いは膜人工化学系の化学反応で記述する。計算主体の振舞いをプログラムコードではなく化学反応で記述できると、様々な利点がある。

ひとつめは、情報を物として捉えて振舞いを定義できることである。蜘蛛は情報を膜人工化学における分体で持つ。計算主体が情報を取得する動きは、蜘蛛がその情報を表す分体を食べるという動作で表現できる。逆に情報を撒くという動きは、情報を表す分体を排出するという動作で表現できる。例えば、あるデータファイルが存在することを蜘蛛困のネットワーク全体に知らせたいとする。その場合はデータファイルを分体に変換して蜘蛛に持たせ、別の計算機に移動したら分体を排出するように振舞いを定義すればよい。このように、情報を物として捉えることで振舞いを簡潔に定義できる。

ふたつめは、計算主体の振舞いの合成が容易にできることである。現在はまだ実装していないが、蜘蛛の融合という動作を加えることを考えている。この動作は、ある振舞い A を持つ蜘蛛 α と振舞い B を持つ蜘蛛 β が融合し、振舞い A と振舞い B の両方を持つ新しい蜘蛛が生まれる動作である。蜘蛛の振舞いは組み換え規則で定義しているので、蜘蛛 α と蜘蛛 β の振舞いの合成は、単純にそれぞれが持つ組み換え規則群の和集合として実現できる。

計算主体の振舞いを化学反応で記述する際の欠点として、情報をどのような形の分体に変換するかを定めることが難しい点が挙げられる。効率よく処理を進めるために、情報の性質をよく見極めて分体の形を定義する必要がある。また、多くの振舞いに対応できるように普遍的な形を選ぶ必要がある。この問題については、ひとつの情報から数種類の分体を生成したり、分体を処理に応じて加工し他の蜘蛛を助ける蜘蛛を定義することで対策できるかもし

れない。

蜘蛛困の応用のひとつとして、情報の自動整理が考えられる。蜘蛛に情報を与え、似た情報や関連する情報を持った蜘蛛が近寄るように蜘蛛の行動を定義する。また、大きく異なる情報を持つ蜘蛛同士は離れるように蜘蛛の行動を定義する。これにより蜘蛛らの自発的な秩序形成を促すことができる。蜘蛛らに対応する情報の再配置を行うことで、乱雑に散らばった情報を整理できると考えられる。情報の整理をしているうちに、より効率よく整理を進めたり、より細かい分類に仕分けたりする蜘蛛が生まれるように、蜘蛛の進化を導入することも考えられる。

5. おわりに

本研究では膜人工化学に基づくネットワークモデル蜘蛛困を構築し、そのランタイムシステムを実装した。実装したランタイムシステムに、計算主体の振舞いとそれが振る舞う環境を与え、期待通りに動作することを確認した。そしてネットワークモデル蜘蛛困の応用の例を示した。今後の課題としては、蜘蛛の動作を追加して、より幅広い分野に応用できるように蜘蛛困を改良することなどがある。

参 考 文 献

- 1) Dittrich, P., Ziegler, J. and Banzhaf, W.: Artificial Chemistries — A Review, *Artificial Life*, Vol.7, pp.225–275 (2001).
- 2) Amari, N. and Tominaga, K.: Simulation Minus One Makes a Game, *EvoWorkshops '09: Proceedings of the EvoWorkshops 2009 on Applications of Evolutionary Computing*, Berlin, Heidelberg, Springer-Verlag, pp.273–282 (2009).
- 3) Lange, D.B. and Oshima, M.: Seven good reasons for mobile agents, *Commun. ACM*, Vol.42, No.3, pp.88–89 (1999).