

平成 23 年度 卒業論文

論文題目

多様性を実現する群知能の  
振舞いのモデル

神奈川大学 工学部 電子情報フロンティア学科

学籍番号 200802957

内山 竜佑

指導担当者 木下宏揚 教授

# 目次

第1章	序論	5
第2章	基礎知識	6
2.1	Boid	6
2.1.1	Separation (引き離し)	7
2.1.2	Alingment (整列)	7
2.1.3	Cohesion (結合)	8
2.2	covert channel の定義	9
2.3	言語ゲーム	11
2.3.1	ウィトゲンシュタインの言語ゲーム	11
2.3.2	言語の限界	11
2.3.3	家族的類似	11
2.3.4	群れる振舞い	12
2.3.5	進化する言語ゲーム	12
2.4	マルチエージェント・シミュレータの種類	14
2.4.1	artisoc	14
2.4.2	Swarm	14
2.4.3	MASON	15
第3章	提案手法	16
3.1	アクセス行列で生じる covert channel の問題	16
3.2	群知能のパラメータ	18
3.3	社会システムにおける公私の価値循環を色で表現する	19
3.4	方針	20
3.4.1	Boid 的規則 (連携)	21
3.4.2	フェロモン (関係性)	21

---

3.5 提案する群知能 . . . . .	23
3.6 方法 . . . . .	24
第4章 結論	28
謝辞	29
参考文献	30
質疑応答	31

## 目次

2.1	衝突の回避	7
2.2	群れの中心に向かう	8
2.3	向きをあわせる	8
2.4	covert channel とフローレベル	10
2.5	進化する(公私の)言語ゲーム	13
3.1	covert channel (間接情報フロー)	17
3.2	図 3.1 に object, subject を増やす方法	17
3.3	言語ゲームとみなすクラウド	18
3.4	色の混色規則	19
3.5	Particle の集まり	20
3.6	RGB それぞれの Particle が 3 つに分離するイメージ	21
3.7	色の濃い Particle に集まるイメージ	22
3.8	群知能のモデル	23
3.9	friendship モデルの実行結果	25
3.10	classmates モデルの実行結果	26
3.11	national モデルの実行結果	27

## 表 目 次

# 第1章

## 序論

インターネット上で様々な種類の情報を交換しながら築かれる人間関係によって、多様なコミュニティ（群れ）が形成される。例えば、コミュニティの例として企業を挙げると、企業は企業間や顧客との多様でダイナミックな関係の中でビジネスするようになってきた。そのような現代において、自分のコミュニティの価値を守りつつも、他のコミュニティの価値を取り入れて、常に活動を続ける社会が望まれる。そのような進化的社会を実現させるための必要条件として、情報漏洩をいかに防止するかという問題がある。そこで、コミュニティからの情報漏えいを制御するシステムを研究する目的で、情報漏えいは covert channel から生じるという点に着目した。

従来のセキュリティは、客体自体を護るという概念である。完全に護ろうとすることが強い制約になり、情報を創発する振舞いを阻害する方向性を持つ。提案するセキュリティは、客体を使って主体の群れが行為することを護る、という概念である。行為し続けることを護るので、主体の群れの振舞い自体が進化し、情報を創発し続けるシステムである。

この研究はクラウドシステムの情報セキュリティを目的とする。常に活動し続ける社会の中で引き起こされる情報漏洩の調和を、個々の情報リソースを護るのではなく、群れの進化的な作用自体を動きの中で保つシステムを探求する。つまり、動的な社会の中に公私の価値の循環（調和）を実現するシステムを研究したい。そこで、「群知能」に着目し、進化と制約の矛盾を解決するような抽象的なシステムをシミュレーションし、多様な社会をアナロジーとして記述するための構造を研究する。

## 第2章

# 基礎知識

### 2.1 Boid

Boidとは1987年にアメリカのアニメーション・プログラマであるクレイグ・レイノルズによって考案・作製された人工生命シミュレーションプログラムである。Boidというモデル名は、鳥もどきという意味の言葉である“bird android (バード・アンドロイド)”が短くなったことに由来している。Boidの群れを実現させる振る舞いは、3つの要素からなり、「衝突の回避」、「群れの中心に向かう」、「向きをあわせる」といった3つのルールを規定するだけで鳥の群れをシミュレーションすることができる[1][?][8]。このように単純な行動規範をそれぞれの個体が持ち、全体として複雑な群れ行動が創発する。

これらをまとめると時刻  $t$  での  $i$  番目の Boid の速度ベクトル  $[\vec{v}_i(t)]$  の更新式は次のようになる。

$$\vec{v}_i(t) = \vec{v}_i(t-1) + \overrightarrow{Next}_i(t-1) + \vec{G}_i(t-1)$$

ここで  $\overrightarrow{Next}_i(t-1)$  は個体  $i$  の最も近くの Boid の速度ベクトル、 $\vec{G}_i(t-1)$  は個体  $i$  から重心へ向かうベクトルである。なお、慣性で動くことを実現するため、1タイムステップ前の速度  $\vec{v}_i(t-1)$  を加えている。

それぞれの Boid は自分の視界を有している。最も近くにいる Boid を探す場合、自分の視界内の Boid だけを考える。ただし、群れの重心を計算するには他の Boid も含めた全体の座標位置を使う[?].

### 2.1.1 Separation (引き離し)

Separation は、近くの鳥や物体に近づきすぎたらぶつからないように離れるルールである。衝突の回避のために、Boid はそれぞれ自分にとっての最適距離を持っている。そして自分の最も近くにいる仲間との間で、この距離を保ちたいと考えて振る舞う。もし、Boid 同士が近づきすぎてしまったら、前を飛んでいる Boid はスピードを速くし、後ろを飛んでいる Boid はスピードを遅くする。障害物、例えば柱や壁などに対しては、それにぶつからないように方向転換して衝突を避けるようにする [1]。

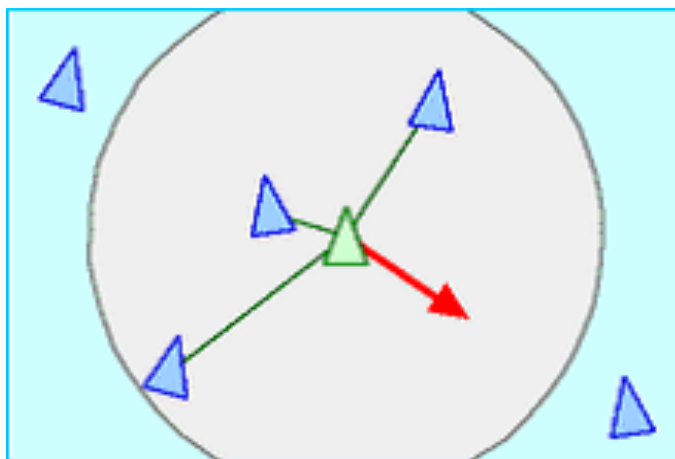


図 2.1 衝突の回避

### 2.1.2 Alingment (整列)

Alingment は、近くの鳥たちと飛ぶスピードや方向を合わせようとするルールである。すなわち、同じ方向にあまり距離を空けないように飛ぶようにする。このルールは、ある一定の距離より遠ざかりすぎてしまったら前を飛んでいる Boid はスピードを遅くし、後ろを飛んでいる Boid はスピードを速くするようにすることで実現することができる [1]。



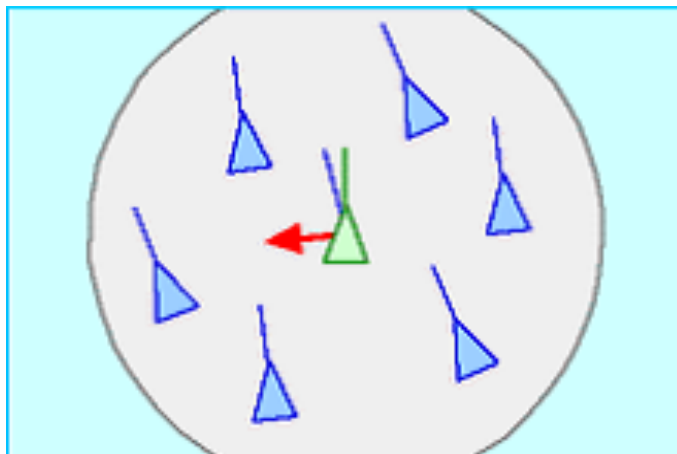


図 2.2 群れの中心に向かう

### 2.1.3 Cohesion (結合)

Cohesion は、鳥たちが多くいる方へ向かって飛ぶルールである。鳥が多くいる方向というのは、大ざっぱにいうと群れの中心（重心）方向ということになる。つまりこのルールは、Boid に群れの中心の方向へ飛んでいくことを指示している。この群れの中心は、全 Boid の位置（座標）の平均として求める [1]。

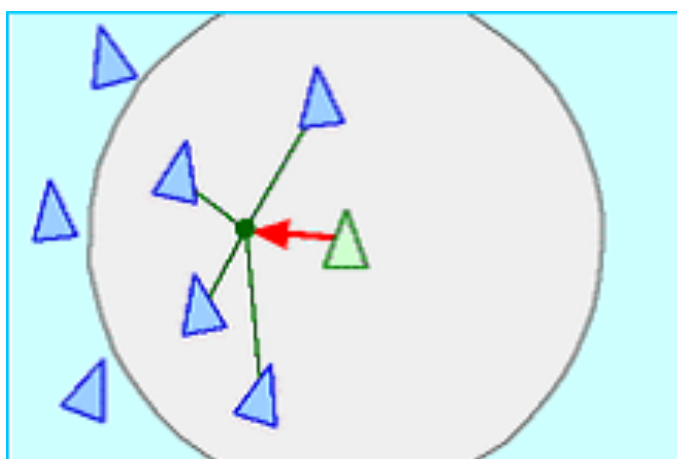


図 2.3 向きをあわせる

## 2.2 covert channel の定義

アクセス行列に於いて, subject, object, パミッションをアクセストリプルと定義し, そのアクセストリプル間で起きる不正な情報の経路 covert channel を定義する.

【定義】covert channel アクセス行列において, アクセス禁止のパミッションに矛盾する情報フローを covert channel と呼ぶ.

即ち,

subject  $S_i (i = 1, 2, \dots), S_j (j = 1, 2, \dots),$  但し  $i \neq j$

object  $O_n (n = 1, 2, \dots), O_m (m = 1, 2, \dots),$  但し  $n \neq m$

パミッション  $P\{RW, \neg RW, \neg WR, \neg(RW)\},$  但し R(READ), W(WRITE)

とする時, アクセストリプル  $\langle S_i, O_n, P \rangle$  について,

If  $\langle S_i, O_n, \neg R \rangle,$

And if  $\langle S_j, O_n, R \rangle \wedge \langle S_j, O_m, W \rangle \wedge \langle S_i, O_m, R \rangle,$

Then covert channel  $(S_i, S_j, O_m(O_n)).$

と定義する. この定義では, subject  $S_j$  は object  $O_n$  という名前とその内容を READ 禁止  $\langle S_i, O_n, \neg R \rangle$  であるにも拘わらず, subject  $S_j$  が object  $O_n$  の内容を READ し, object  $O_m$  にそれを WRITE し, subject  $S_i$  が  $O_m$  の内容から object  $O_n$  の内容を READ する事 ( $\langle S_i, O_n, R \rangle$  と表現される) によって, アクセストリプル  $\langle S_i, O_n, \neg R \rangle$  と矛盾する結果が生じる事が示される. また, covert channel が引き起こされる時, それに関わる subject の数によって covert channel の程度 (フローレベル) を定義する.

【定義】フローレベル covert channel に於いて, ある subject からある subject へと情報が流出する回数をフローレベルと定義する.

また, フローレベル  $k$  はフローレベル 2 が基になっている [2]. 図 2.10 に covert channel とフローレベルの例を示す.

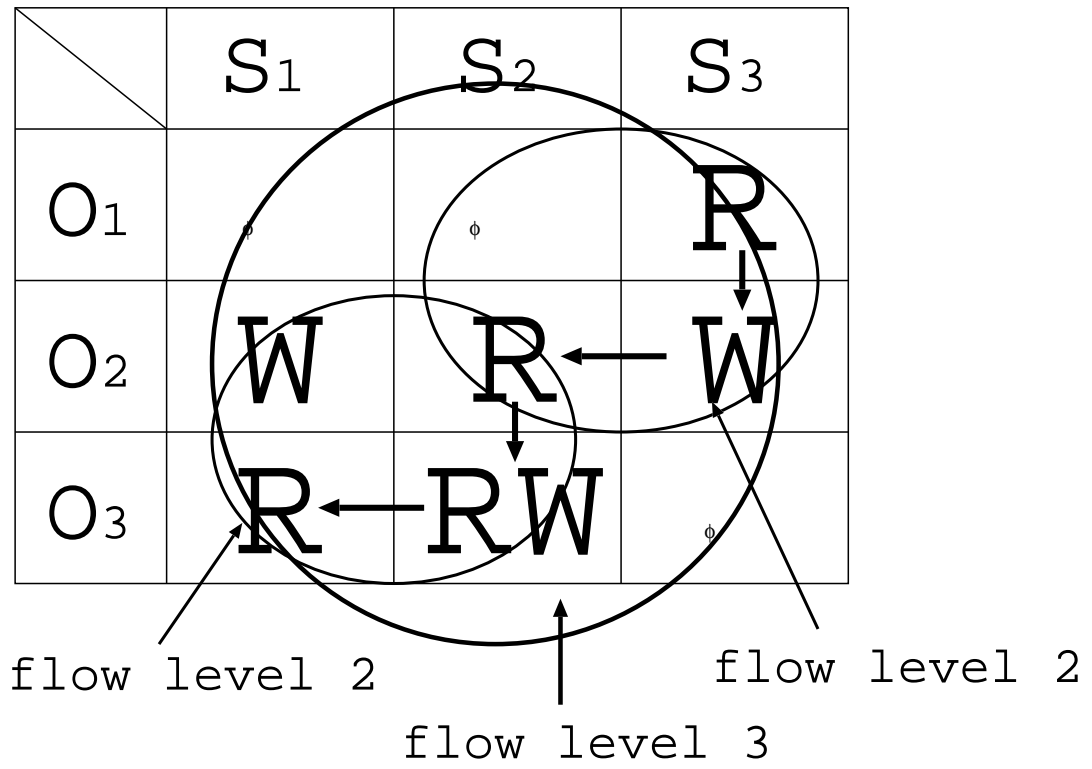


図 2.4 covert channel とフローレベル

## 2.3 言語ゲーム

### 2.3.1 ウィトゲンシュタインの言語ゲーム

- “ <規則に従う>ということは一つの実践である ”。(Wittgenstein“ 哲学探究-202 ”)
- “ 規則に従っているとき , ... わたくしは規則に盲目的に従っているのだ ”。(Wittgenstein“ 哲学探究-219 ”)
- “ ... 言葉の意味はその使用法である ”(Wittgenstein“ 確実性の問題-61 ”)
- “ 意味と規則のあいだにはひとつの対応関係が成立する ”(Wittgenstein“ 確実性の問題-62 ”)
- “ 知識の究極の根拠は承認にある ”(Wittgenstein“ 確実性の問題-378 ”)
- “ 実在とは , 予期されたものにはまだ欠けており , その予期が実現するとそれに付け加わる性質ではない . 実在はまた , 暗闇において , いわば無色で存在していたものに色を与える日の光のようなものでもない ”(Wittgenstein“ 断片-60 ”)

### 2.3.2 言語の限界

規則に基づく行為は言語の限界であり , 言語世界に住む <私> の観察視点からは , 言語的に量子化された <私> は言語の規則に盲目に従うように見える . その意味で規則や規則に基づく行為は見直され , 社会システムのために転回される必要がある . すなわち , 言語から規則や規則に基づく行為 “ 言語ゲーム ” へと思索の視点を変えねばならない . しかし , そのような “ 見直し ” には限界がある . ひとたび行為世界に入ったとしても “ 見直し ” は限界のある言語によって記述されなければシステムを構成する事が出来ない .

### 2.3.3 家族的類似

言語ゲームで振る舞う particle は , 家族的類似性によって群れを作る , と定義した . 家族的類似性は言語と行為の類似性を表現するものである . 家族的類似性は同値関係でもないし , 等価関係でもない . 常に変動しつつ , 少しずつ似ているエンティティ

の集まりである。しかし、それは自己から見れば、同値関係であってもよいし等価関係であってもよい。そのような個人個人の意味論を統合して世界を記述する意味論が無い、ということである。家族的類似性は不確実な世界の中の同類の定義である。家族的類似性は公的言語世界の観察視点の同類の定義である。

#### 2.3.4 群れる振舞い

集まる力の源は、“家族的類似”であり、群知能のパラメータとして表現される。  
群れる正の力

- 群れの中心に向かう力： Cohesion
- 隣人と家族的類似行為をする力： Alignment
- 行為濃度： Pheromone

Pheromone は行為の軌跡の重要性を表現する。Pheromone は揮発性である。  
濃度が濃い Pheromone は重要な行為を表す。

群れる負の力

- 群れから排除する力、群れから離れる力： Separation

#### 2.3.5 進化する言語ゲーム

言葉の定義

- 行動： 人のはたらきかけ。
- 行為： 言語・規則に則る行動。
- 相互作用： 行為と行為がはたらき合うこと。
- 振舞い： 生活世界に裏打ちされた行為。言語ゲームの部分を成す。

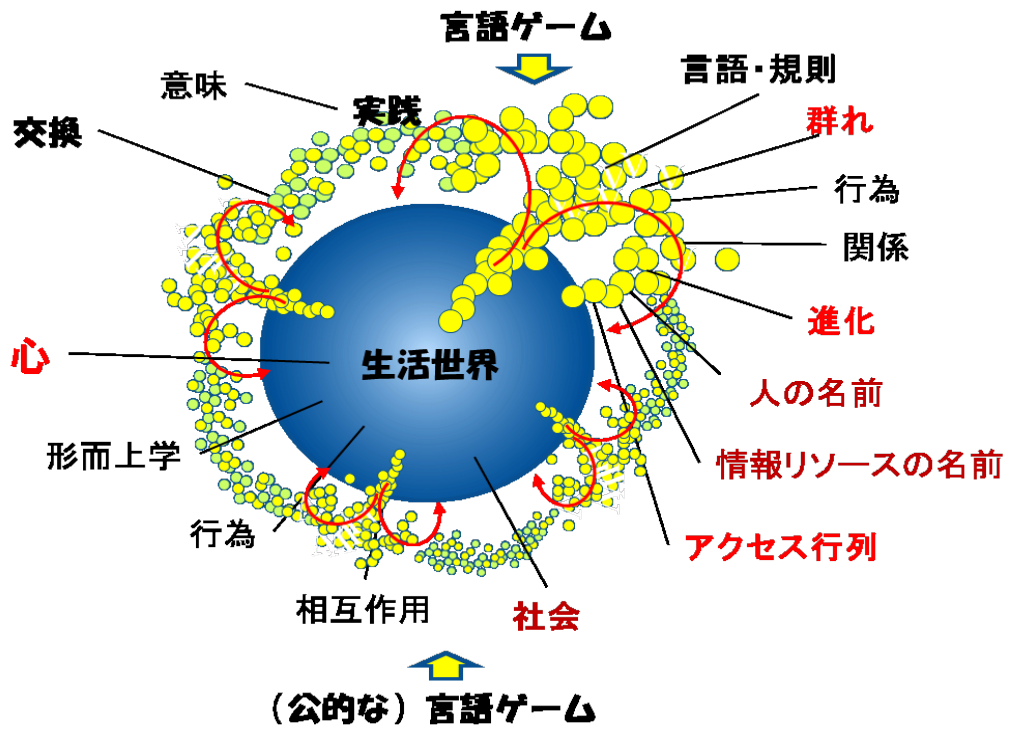


図 2.5 進化する(公私の)言語ゲーム

## 2.4 マルチエージェント・シミュレータの種類

### 2.4.1 artisoc

もっぱら理科系のツールとして開発されていたマルチエージェント・シミュレーションが、社会科学の方法のひとつとしても有望ではないかと言われはじめてから10年以上たつ。しかしこの手法が今でもあまり浸透していないのは、社会科学の研究者や学生にとって、まずプログラミング言語から学ぶ必要があり、要するに敷居が高い手法だったことに起因すると思われる。この事情は、アプリケーションソフトが多数市販されている統計分析の手法と比べると一目瞭然である。

たしかにアメリカを中心に、人工社会を念頭においた汎用シミュレータはいくつか存在しているが、使いやすいシミュレータでは単純で定型的なモデルしかつけない。複雑で様々なモデルを作れる強力なシミュレータを操作するには高度な専門知識が必要である。

そのような状況で、数年前に登場したものが Windows パソコン用のシミュレータ KK-MAS である。これは構造計画研究所が開発した、日本語環境のパソコンの上で、プログラミング言語やプログラミング技法を学ばなくても利用できる、しかも汎用性のあるマルチエージェント・シミュレータである。この KK-MAS をプロトタイプにしつつ、社会現象のモデルを作り、シミュレーションを実行することを念頭に置いて開発された第2世代のシミュレータが artisoc<sup>[6]</sup> である。

### 2.4.2 Swarm

Swarm は、生物エージェント（虫など）と無生物エージェント（壁や障害物など）からなる人工世界のボトムアップモデルに基づくシミュレータである。エージェントの動きを簡単な規則で記述し、エージェント間の相互作用による創発現象を観察する。

Swarm の基本的な特徴は次のようになる。

1. シミュレーション・プログラミングを補助するオブジェクト指向型のソフトウェアライブラリーの集まりである。
2. ユーザは自分のプログラムから Swarm ライブラリーのオブジェクトを取り込むことでシミュレーションを構築する。
3. Swarm のライブラリーは Java と Objective-C に対して提供されている。

Swarm のシミュレーションは

- Model (モデル) :人工世界を抽象的にモデル化しシミュレートする.
- Observer (オブザーバ) :モデルをシミュレーションを観察し表示などを行う.

の2つの部分からなる [?].

Swarm のソースコードと Swarm を動作させるために必要なソフト類は,  
<http://www.santafe.edu/projects/swarm/> から入手することができる .

### 2.4.3 MASON

2次元・3次元の空間を使用した物理・社会モデル用のシミュレーションライブラリ.適切なクラスを継承してモデルを実装することでそれに合わせた GUIが提供される.モデル開発というより, MASON ライブラリを使用したアプリケーション開発という色合いがどうしても強くなってしまふ.分析機能はなく,動きを見て楽しむというのが主眼のようである [4].

MASON を動作させるために必要なソフト類は,  
<http://www.cs.gmu.edu/eclab/projects/mason/> から入手することができる.



## 第3章

# 提案手法

### 3.1 アクセス行列で生じる covert channel の問題

図 3.1 の場合, 矢印の流れで subject2 が本来読めないはずの object1 を読めてしまう. covert channel 流出の流れは以下のようにになっている. これは間接情報フローとも呼ばれる.

- subject1 が object1 を読み込む.
- subject1 が object2 に object1 で読んだ内容を書き込む.
- subject2 が object2 を読む.
- covert channel により間接的に object1 の内容を読めてしまう.

つまり図 3.1 の場合, 情報漏えい起きたことになる.

	S1	S2
O1	r	Φ
O2	w	r

図 3.1 covert channel ( 間接情報フロー )

( 各 S:subject, 各 O:object, r:読みこみ可能, w:書き込み可能, Φ:読み込み, 書き込み不可 )

このままであれば, このパターンを否定するのが妥当だが, それでは使いづらいため, 工夫を施す必要がある. そもそも covert channel は, S1 と S2 が「連携しているか」, 「競争しているか」によって状況が変わるはずである. また, アクセス権を動的に許可, 不許可することによっても covert channel を制御できるだろう.

あるいは下図のようにすれば, covert channel はなくなるだろう.

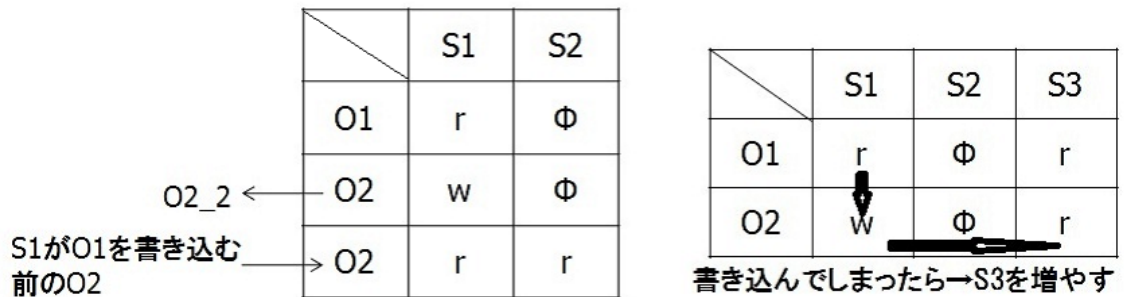


図 3.2 図 3.1 に object, subject を増やす方法

図 3.1 の covert channel は感染 ( 遺伝 ) に相当する ( 創発ともいえる ). その影響を受けても生き残るために, 図 3.2 では object 或いは subject を増やした. それは即ち生命システムのアナロジーである. つまり, 進化のシステムともいえる.

## 3.2 群知能のパラメータ

クラウドを関係の地図と見做していたが、ここではクラウドを言語ゲームと見做す

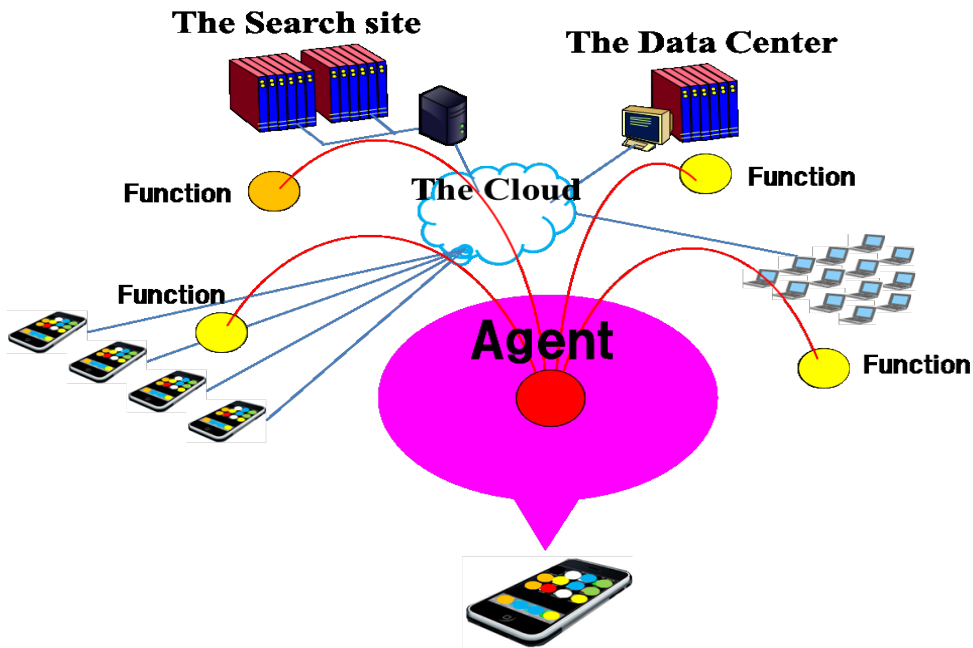


図 3.3 言語ゲームとみなすクラウド

提案するセキュリティの概念は、“群れの振舞い自体を護る”，という概念である。群れの振舞い自体を何から護るのか？攻撃者は変動する秩序自体である。“秩序とは言語ゲームという振舞いが群れにとって善くはたらいている状態をいう”。言語ゲームという振舞いの中には群れや情報リソースや行為自体が含まれる。変動的な攻撃に対処するため群れは進化し続けなければならない。進化する言語ゲームのための必要条件が群知能である。その時、情報漏洩は完全に無くなるのでない。情報漏洩はシステムが定める範囲で少なくなる。そのことと引き換えに、“群れの振舞いの帰結としての情報リソースの内容自体の創発を活性化させる効果”を期待する。提案するセキュリティは、群れの振舞い自体が進化し、情報リソースを創発することを継続するためのシステムである。

### 3.3 社会システムにおける公私の価値循環を色で表現する

「色の混色」と「鳥の群れ」を社会システムのアナロジーとして考える。着目するアナロジーは、「色の混色」という操作と、「鳥の群れ」という相互作用である。情報の意味を創発することのアナロジー、群れをなして意味を創発するアナロジーは、色光の三原色の加法混色と色料の三原色の減法混色を使って表現される。

#### 色光3原則（加法混色）

R: red  
G: green  
B: blue

#### 色料3原則（減法混色）

C: cyan  
M: magenta  
Y: yellow

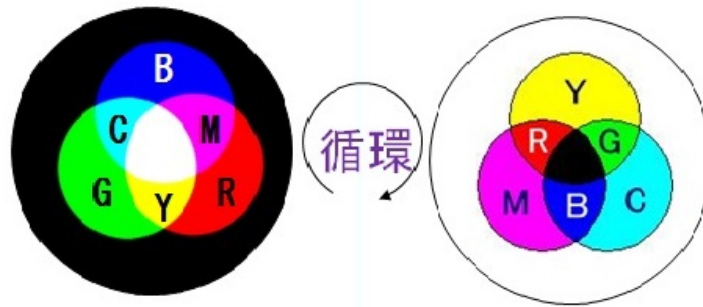


図 3.4 色の混色規則

### 3.4 方針

図にあるように、1つ1つの丸は Particle であり、それぞれの Particle は人を表し、Particle の集まりは人が集まる群れを表している。群れが（濃度の高い）中心に集まる振舞いをマルチエージェントシミュレータで表現する。関係性・連携作用に着目した、群知能の振舞いをシミュレーションし、群れが調和するパラメータを求めたい。求めたパラメータから、各 Particle 間のシンプルな規則によって群れの多様性を維持するメカニズムを究明する。

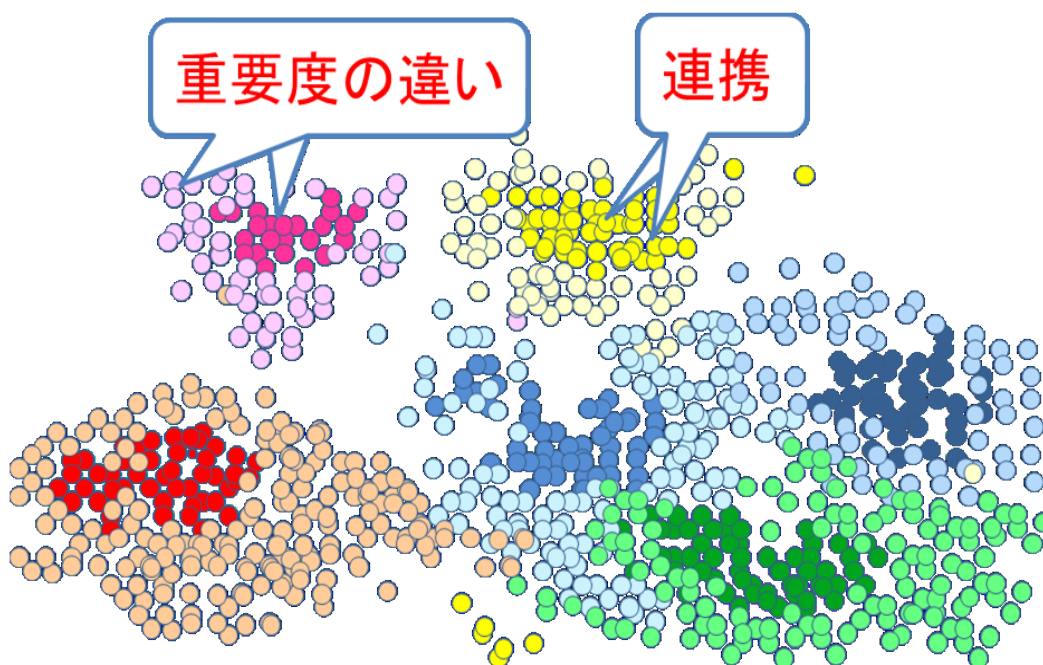


図 3.5 Particle の集まり

### 3.4.1 Boid 的規則 ( 連携 )

Boid 的規則, 即ち連携は,

- Particle は互いに衝突を避ける
- 同じ色の Particle が集まる
- 同じ色の Particle は重心に向かって移動する

といった規則を持つ.

色彩三原色 RGB の Particle の群れが RGB に関わらずばらばらにある状態から, RGB それぞれの Particle が, R の群れ, G の群れ, B の群れに分離するプログラムの構造を考える.

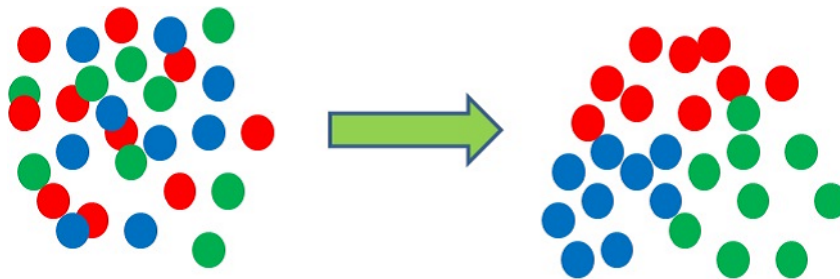


図 3.6 RGB それぞれの Particle が 3 つに分離するイメージ

### 3.4.2 フェロモン ( 関係性 )

フェロモンの概念から、

- 各 Particle に異なった重要度を持たせ、「異なった重要度 = 色の違い」とする
- Particle 達は、似ている色 ( フェロモン ) の濃い ( 重要度が高い ) Particle のもとに集まる
- 他の色の群れにも、Particle は関心を示す

という規則を持たせる.

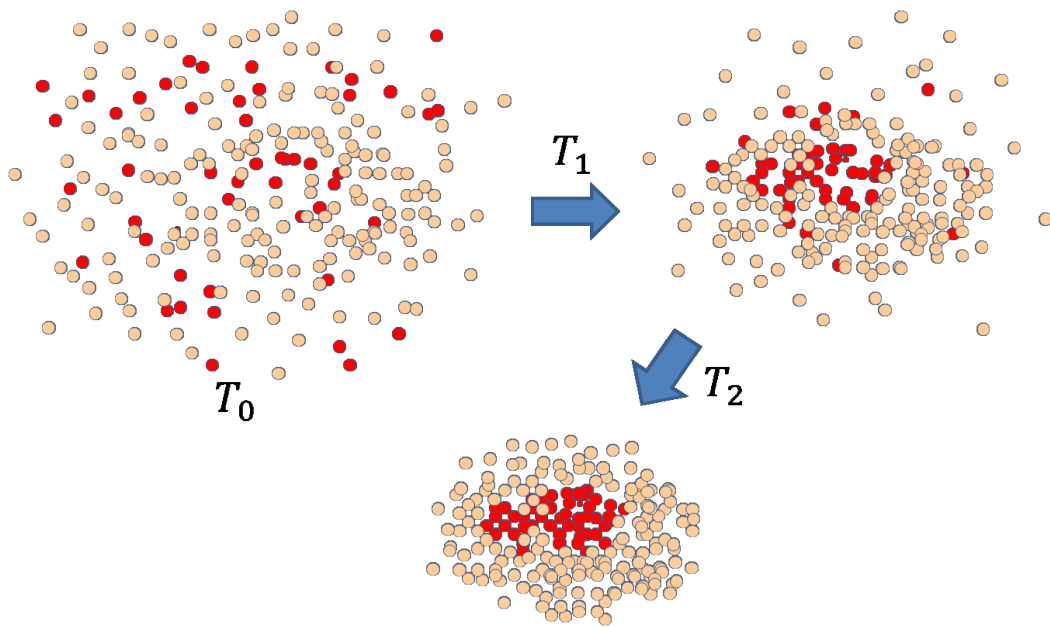


図 3.7 色の濃い Particle に集まるイメージ

### 3.5 提案する群知能

Boid 構造は連携を表現し, Particle は群れを作り動き回るための「衝突の回避」, 「群れの中心に向かう」, 「向きをあわせる」といった規則を持つ. 群れからの情報漏えいを調和するシステムを研究していくにあたって、群れ同士の相互作用、特に「関係性」と「連携作用」に着目する。関係性を「色彩」・「明度」、連携作用を「Boid」の群れ作用とACOの「フェロモン」として表現したい。

“ 群知能のモデル ” は収束解を得るためのモデルではなく、変化しながらも秩序を保つ、その構造とは何かを究明するものである。

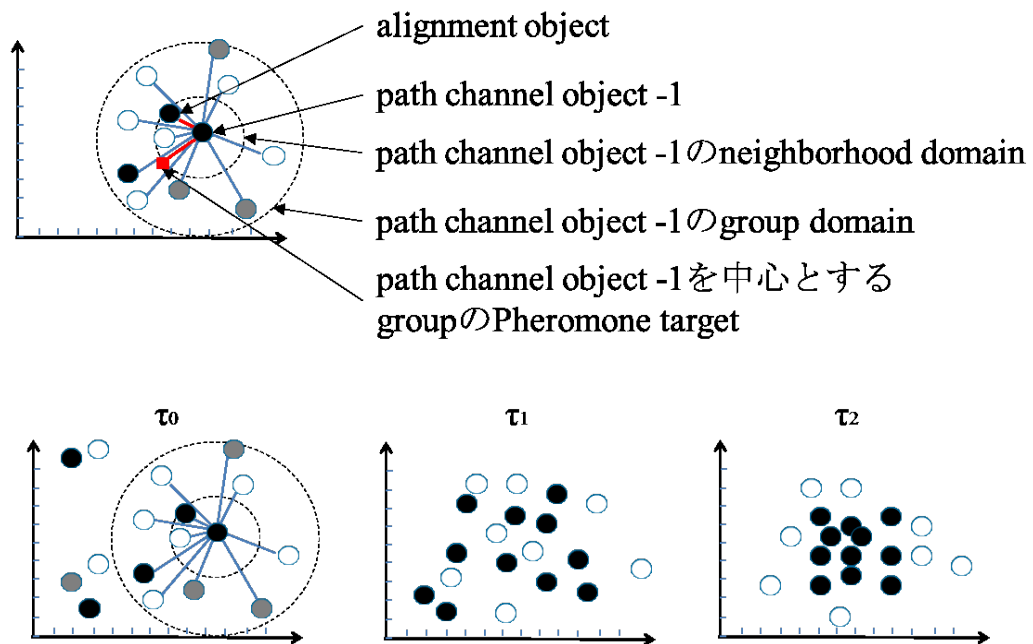


図 3.8 群知能のモデル



## 3.6 方法

コミュニティ同士の連携や競争といった相互作用を研究することで、コミュニティの価値の漏洩を防ぎつつ、常に活動し続ける社会を表現するというのが研究の目的である。

それを実現するためには、`artisoc` というマルチエージェント・シミュレータの様々なサンプルモデルの中から、次の機能が特に必要なのではないかと考えた。

- 「friendship モデル」でのエージェントを集合させる機能 …… A
- 「classmates モデル」での「好意が重複 = 濃度が高い」とし群れさせる …… B
- 「national モデル」での微妙な色を数値化 …… C

### friendship モデル

friendship モデルの入力コマンドと実行結果の図を下記に記述する。

```
Agt_Init{ //シミュレーションの最初だけに実行されるルール
My.X = Rnd()*50 //50 x 50 の空間の中で
My.Y = Rnd()*50
My.Direction = Rnd()*360 //ランダムな方角に進む
ClearAgtset(My.friends) //friends を初期化 (中を空に) する
}

Agt_Step{ //毎ステップ実行されるルール
Dim s As Integer //s という変数を整数型として (As Integer)
Dim one As Agt
Dim temp As Agtset
Dim close As Agtset
Dim neighbor As Agtset
Turn(Rnd()*360) //とりあえず、好みはランダムに変わろうとする
//自分の好みと似ている人を友人にする
MakeAllAgtSetAroundOwn(neighbor, 3, False)
If CountAgtset(neighbor) > 0 then
    one = GetAgt(neighbor, Int(Rnd()*CountAgtset(neighbor)))
    AddAgt(My.friends, one) //one を My.friends に追加
    TurnAgt(one) //one の好みにあわせようとする
    //My.friends の重複をなくす
    DuplicateAgtset(My.friends, temp)
End if
Forward(1)
}
```

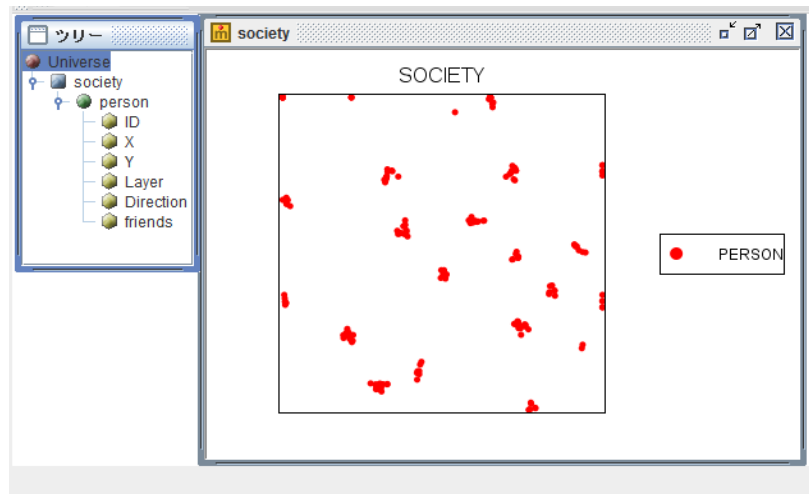


図 3.9 friendship モデルの実行結果

friendship モデルは、好みの近い者同士が友達になる過程をモデル化したものである。コマンドの大まかな意味は、//の後に示した。

## classmates モデル

classmates モデルの入力コマンドと実行結果の図を下記に記述する。

```

Agt_Init{
My.X=Rnd()*20
My.Y=Rnd()*20
My.Direction=Rnd()*360
}

Agt_Step{
Dim i As Integer
Dim one As Agt
Dim neighbor As Agtset
Turn(Rnd()*180-90)
If Forward(1)<>-1 Then //前進してぶつかったらUターン
    Turn(180)
End if
//近くにいる同級生と遊ぶかケンカする
MakeAllAgtsetAroundOwn(neighbor,2,False)
For each one in neighbor
    i=Round(Rnd()) //四捨五入して0(遊ぶ)か(ケンカ)
    My.relation(one.ID,i)=My.relation(one.ID,i)+1 //関係を蓄積
Next one
//同級生と恒久的関係の構築
ClearAgtset(My.friends) //好きな相手を初期化
ClearAgtset(My.enemies) //嫌いな相手を初期化
MakeAgtsetspace(neighbor,Universe.classroom)
For each one in neighbor
    If My.relation(one.ID,0)>=My.relation(one.ID,1) +3 Then
        AddAgt(My.friends,one) //好きな相手に加える
    ElseIf My.relation(one.ID,1)>=My.relation(one.ID,0) +3 Then
        AddAgt(My.enemies,one)
    End if
Next one
}

```

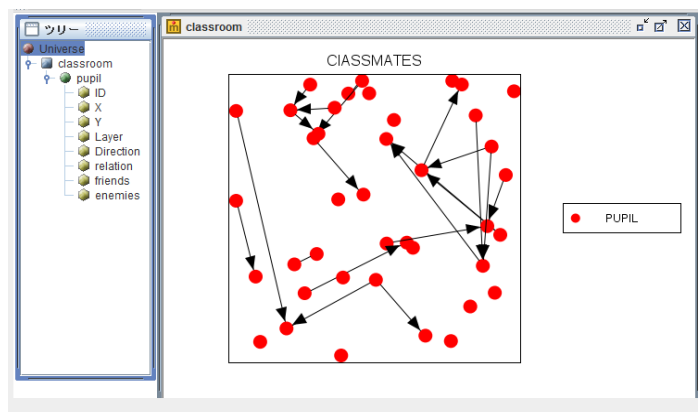


図 3.10 classmates モデルの実行結果

### national モデル

national モデルの入力コマンドと実行結果の図を下記に記述する。

```
Agt_Init{
My.X = 1
My.Y = 1
My.PGR = 0.02 + (Rnd() - 0.5) / 50
My.EGR = 0.02 + (Rnd() - 0.5) / 50
}

Agt_Step{
My.X = My.X*(My.PGR + 1)
My.Y = My.Y*(My.EGR + 1)
If My.Y >= My.X + 5 Then
  My.ecolevel = Color_Red
Elseif My.X >= My.Y + 5 Then
  My.ecolevel = Color_Blue
Else
  My.ecolevel = Color_Green
End if
If My.X >= 50 Or My.Y >= 50 Then
  ExitSimulationMsgLn("Completed after" & GetCountStep() & "steps")
End if
}
```

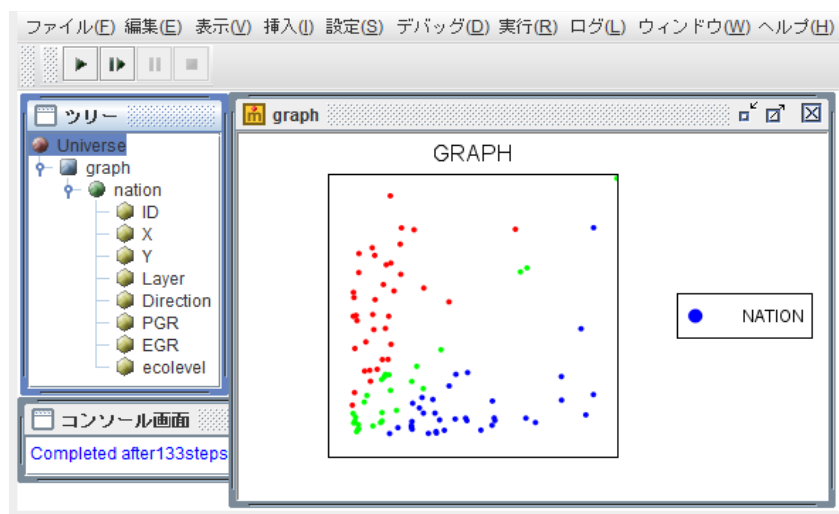


図 3.11 national モデルの実行結果

## 第4章

# 結論

コミュニティ同士の情報漏えいを防ぎ、常に活動し続ける社会を表現するため、情報漏えいは covert channel から生じるという点に着目した。

提案システムでの色の違いはセキュリティポリシーの違いが反映されている。シミュレーションで得られたエージェント間の距離はコミュニティにおける活動を表していて、距離が近いほどそのアクティビティ（情報交換で新しいものを作ること）が高いことを意味する。従来のアクセス制御では情報漏洩が起きないことを最優先として、情報フィルターを一意的に適用していたが、それではコミュニティのアクティビティを阻害してしまう可能性が高い。提案方式ではこのエージェント間の距離に応じて情報フィルターを適用することにより、アクティビティを維持しつつ深刻な情報漏洩のみを防止することが可能となる。

提案した群知能のモデルを作成することができなかった。Boid のモデルを用いて particle を集めようと考えていたが、artiso の Boid モデルは提案するモデルの集まる作用に不適合であると感じた。そこで、Boid に代替する friendship のモデルを利用しようとしたが、実装まで到達できなかった。

今後群知能モデルを完成し、パラメータを得ることができたら、現実の covert channel 問題と色彩のモデルの対応関係を更に明確にでき、現実のシステム制御パラメータに置き換える研究に繋がっていくのではないかと思う。

## 謝辞

本研究を行本研究を行うにあたり，終止熱心にご指導して頂いた木下宏揚教授と南出氏，ご多忙の折研究室に足を運び様々な面で有益なご助言をして頂いた森住哲也氏に深く感謝いたします．さらに，公私にわたり良き研究生活送らせて頂いた木下研究室の方々に感謝いたします．

2012年2月  
内山 竜佑

## 参考文献

- [1] ”Boid とは”  
<http://members.jcom.home.ne.jp/ibot/boid.html>
- [2] 小松充史, 木下宏揚: ”Covert Channel 分析制御のために推論を導入した情報フィルタに関する研究”
- [3] 山影進: ”人工社会構築指南 artisoc によるマルチエージェント・シミュレーション入門 (2010)”
- [4] ”マルチエージェントシミュレータ比較”  
<http://www.gpgsim.net/gpgsim/comp-mas.html>
- [5] 山影研究室  
<http://citrus.c.u-tokyo.ac.jp/mas/models/models.htm>
- [6] MAS コミュニティ - サンプルモデル  
<http://mas.kke.co.jp/modules/mydownloads/viewcat.php?cid=1>
- [7] ウイトゲンシュタイン (著), 藤本 隆志 (翻訳) ”ウイトゲンシュタイン全集 8 『哲学探究』 (Philosophical investigations)”
- [8] ”ウィキペディア (Wikipedia)”  
<http://ja.wikipedia.org/wiki/>

## 質疑応答