

社会・経済シミュレーションのモデル・パターン： 複雑系における動的な変化を記述する

Model Patterns for Economic and Social Simulations:
Describing the Dynamic Changes in Complex Systems



岡部 明子

a130457@cuc.ac.jp

千葉商科大学商経学部

井庭 崇

iba@cuc.ac.jp

千葉商科大学政策情報学部 専任教員(助手)

Agenda

1. 背景と目的
2. モデル・パターンの提案
3. モデル・パターンの適用例
4. まとめ

「社会・経済シミュレーションのモデル・パターン：
複雑系における動的な変化を記述する」

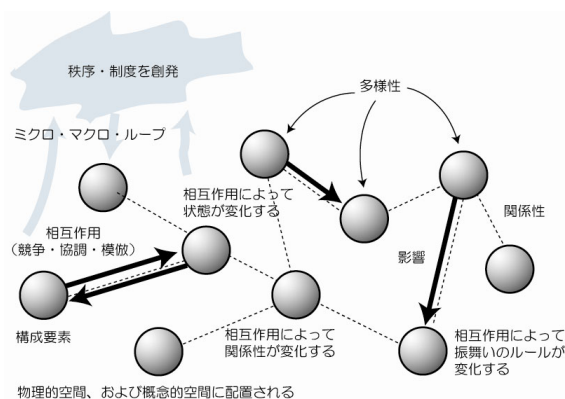
Agenda

1. **背景と目的**
2. モデル・パターンの提案
3. モデル・パターンの適用例
4. まとめ

「社会・経済シミュレーションのモデル・パターン：
複雑系における動的な変化を記述する」

複雑系としての社会・経済のモデル化

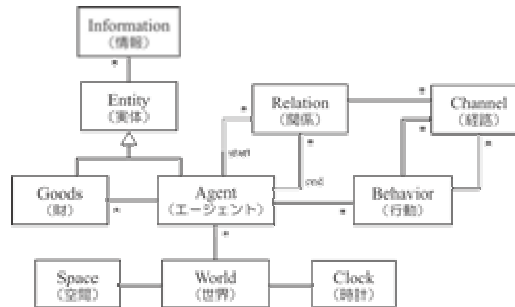
- 社会・経済を理解するためには、複雑系の捉え方が重要だといわれている [出口 00, 塩沢 97, 井庭・福原 98 等]。



- しかし、このようなモデルの記述は困難である。

これまでの私たちの提案① モデル・フレームワーク Boxed Economy Foundation Model

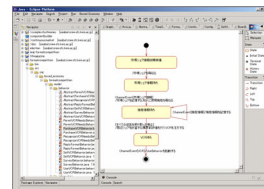
- 複雑系としての社会・経済モデルの枠組みを定義
 - 現実の社会・経済をオブジェクト指向分析によって抽象化したもの。



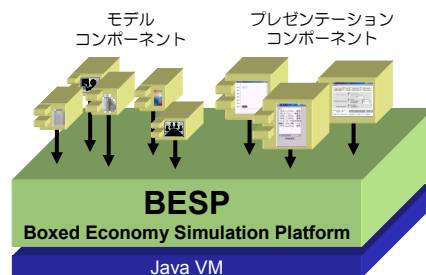
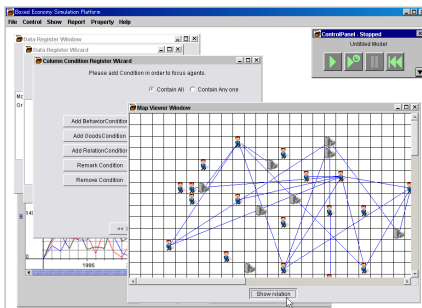
- エージェントの行動をエージェントとは別のモデル要素として定義している。
- エージェント自身が新しい行動の追加や削除、組み換えを行うことができる。

これまでの私たちの提案② シミュレーション・プラットフォーム Boxed Economy Simulation Platform (BESP)

- 複雑系としての社会・経済モデルのシミュレーションを、作成・実行・分析するためのプラットフォーム。



モデルコンポーネントを作成するためのComponent Builder



BESPは、<http://www.boxed-economy.org/> より、無料でダウンロードできます。

本発表の目的

- モデル・フレームワークとシミュレーション・プラットフォームによって、モデルの記述と操作が可能になった。
- しかし、これらを用いて、どのようにモデルを作成していけばよいのかということは、明らかではない。
- 本発表では、モデルの記述のノウハウを「モデル・パターン」としてまとめることを提案する。

Agenda

1. 背景と目的
2. **モデル・パターンの提案**
3. モデル・パターンの適用例
4. まとめ

「社会・経済シミュレーションのモデル・パターン：
複雑系における動的な変化を記述する」

パターンの考え方

- 分析や設計で繰り返し現れる問題を明らかにし、その問題の解法をまとめたもののこと。
- 初心者であっても、効率的な方法でその問題を解決することができる。
- その設計原理についての共通の語彙を提供する。
- パターンは、もともと建築デザインのために考案され [Alexander, 1979]、その後ソフトウェア・デザインに取り入れられている [Gamma等, 1995]。

モデル・パターン

- ここでは、このパターンの考え方を社会・経済のモデル・デザインに導入する。
- モデル・パターンとは、モデル要素をどのように組み合わせ、動的な振る舞いを構成すればよいのかをまとめたもののことである。

モデル・パターンのカタログ

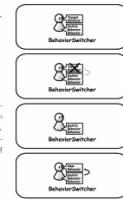
- いくつかのパターンを集めて体系化し記述化したもの。

行動変化のモデル・パターン

Behavior Switching

目的
エージェントが持っている行動を削除し、新しい行動を追加する。

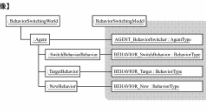
動機



基本動作
BehaviorSwitcher コンプオネントは TargetBehavior と SwitchBehaviorBehavior を持っている。SwitchBehaviorBehavior によって、TargetBehavior を削除し、SwitchBehaviorBehavior によって、NewBehavior という新しい行動を追加する。


設計

【全体像】



193

【SwitchBehavior Behavior】



サンプルコード

【BehaviorSwitchingWorld クラス】

```
...
public void switchBehavior() {
    // ...
    Agent agent = new Agent();
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
    BehaviorSwitchingWorld agentWorld = new BehaviorSwitchingWorld(agent);
}
...

```

【SwitchBehavior Behavior クラス】

```
...
protected void setBehavior(Behavior b) {
    // ...
    this.agentWorld.addBehavior(b);
    // ...
}
...

```

194

本発表で提案するモデル・パターン

- **エレメンタリーなモデル・パターン**
 - Agent Creation
 - Relation Creation
 - Related Agent Creation
 - Agent Destruction
 - Goods Creation
 - Information Creation
- **コミュニケーションのモデル・パターン**
 - Information Sending
 - Blank Information Sending
 - Internal Information Sending
 - Immediate Reply
 - Collect Immediate Replies
 - Appointed Destination Reply
 - Super BehaviorType Calling
- **行動変化のモデル・パターン**
 - Behavior Creation
 - Behavior Destruction
 - Behavior Switching
 - Temporary Behavior Creation
 - Requested Behavior Attachment
 - Forced Behavior Attachment
- **アクティベーションのモデル・パターン**
 - TimeEvent Distributer Agent
 - TimeEvent Filtering
 - TimeEvent Distributer Behavior
 - Time-Consuming Behavior

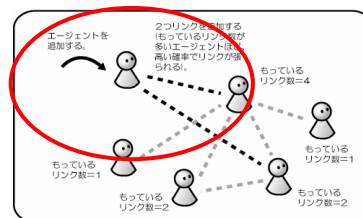
Agenda

1. 背景と目的
2. モデル・パターンの提案
3. モデル・パターンの適用例
4. まとめ

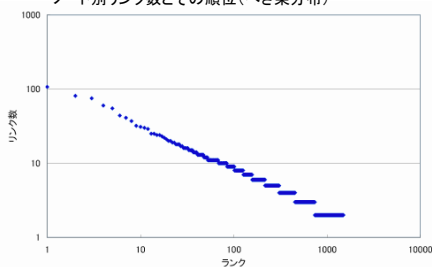
「社会・経済シミュレーションのモデル・パターン：
複雑系における動的な変化を記述する」

適用例① 成長するネットワーク

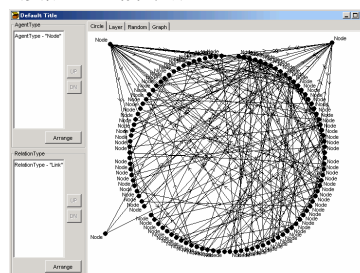
- 「スケールフリーネットワーク」の生成モデル [Barabasi, 1999][Barabasi, 2002]



生成されたネットワークにおける
ノード別リンク数とその順位（べき乗分布）



生成されたネットワークにおける構造
（少数のハブが存在する）



適用例① 成長するネットワーク

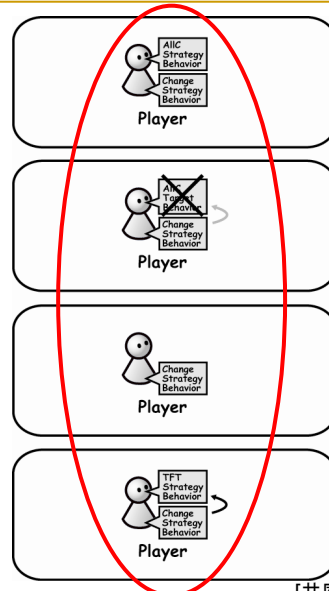
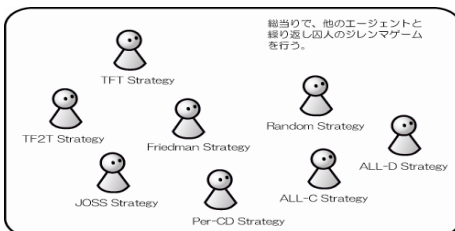
このモデルで使用しているモデル・パターン

Agent Creation	ノードが追加される処理で使用
Related Agent Creation	管理エージェント (Organizer) が新しいノードを生成して、そのノードに対する関係を結ぶ処理で使用

カタログ p.8~9

適用例② 繰り返し囚人のジレンマ

- 「囚人のジレンマ」は、利己的な主体間で利害が対立する状況において、どのように協調が形成されるのか、を調べるための枠組みとして用いられている。
- **【拡張】**各トーナメント終了後、自分に勝った戦略のなかから1つを選び、その戦略に変更する。



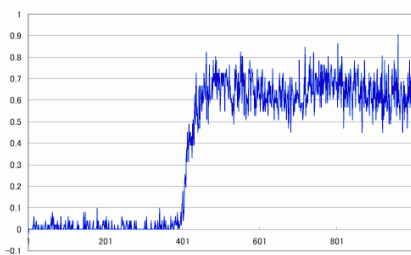
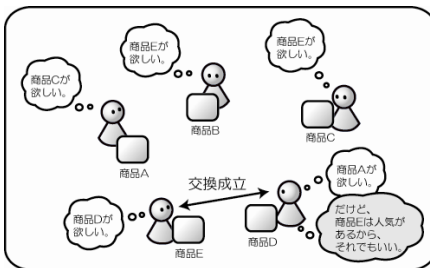
適用例② 繰り返し囚人のジレンマ

このモデルで使用しているモデル・パターン

Information Creation	自分の戦略に基づいて次回の手をInformationとして作成する処理に使用
Information Sending	Informationとして作成した次回の手を送る処理に使用
Immediate Reply	各エージェントが自分の戦略に基づいて手をだす処理に使用
Internal Information Sending	総当たり戦を管理するエージェントにおいて、総当たり戦を管理する行動から、今回対戦するペアの情報、各対戦を管理する行動に連絡する時に使用
Blank Information Sending	各ペアに対戦開始を告げる処理に使用
Immediate Reply	次回の手を聞かれて返答する処理に使用
Behavior Creation	戦略変更時に、新しい戦略を付与する時に使用
Behavior Destruction	戦略変更時に、古い戦略を削除する時に使用
Behavior Switching	戦略変更時に、古い戦略を新しい戦略に切り替える時に使用 (※ Behavior Destruction と Behavior Creation をあわせたパターンです)

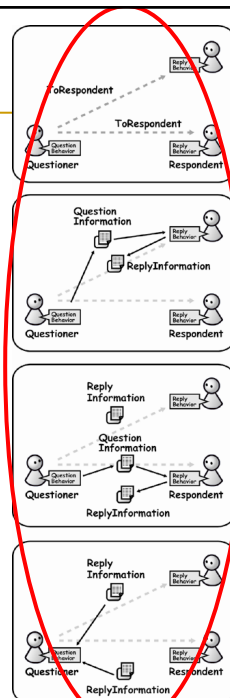
カタログ p.48~49

適用例③ 貨幣の自生と自壊



主体全員が生産者かつ消費者である社会において、物々交換している商品のひとつが、ある時から貨幣としての役割を担うというモデル。
[安富, 2000]

[井庭 03]



適用例③ 貨幣の自生と自壊

このモデルで使用しているモデル・パターン

Goods Creation	行動終了時に財を生産する処理で使用
Information Creation	主体が交換しあう「商品に対する見解」をInformationとして作成する処理に使用
Information Sending	Informationとして作成された「商品に対する見解」を送る処理に使用
Immediate Reply	所有財リストを質問された時にただちに応答する時に使用
Internal Information Sending	主体が、各ステップの処理を行うときに、前のステップで得られた計算結果を次のステップを管理する行動に渡す時に使用
Collect Immediate Replies	全ての主体に所有財リストを尋ねて、複数の返答を同時に受け取る時に使用
Super BehaviorType Calling	モデルの拡張によって、物々交換取引時に追加された処理(知識交換過程など)を呼び出す時に使用

カタログ p.32~35

Agenda

1. 背景と目的
2. モデル・パターンの提案
3. モデル・パターンの適用例
4. まとめ

「社会・経済シミュレーションのモデル・パターン：
複雑系における動的な変化を記述する」

まとめ

- 本発表では、「パターン」の考え方を社会・経済モデルに適用し、Boxed Economy Foundation Modelのためのモデルパターンを具体的に提案した。
- 今後、これらのようなモデル・パターンを追加・検討し、蓄積していくことが重要である。
- 初心者によるモデル・パターンの利用の効果についても、検証していく必要がある。
- モデル・パターンを組み込んだモデル作成支援ツールの開発も考えられる。

第2回 情報科学技術フォーラム
Forum on Information Technology (FIT2003)

社会・経済シミュレーションのモデル・パターン： 複雑系における動的な変化を記述する

Model Patterns for Economic and Social Simulations:
Describing the Dynamic Changes in Complex Systems

岡部 明子

a130457@cuc.ac.jp
千葉商科大学商経学部

井庭 崇

iba@cuc.ac.jp
千葉商科大学政策情報学部 専任教員(助手)