

エージェントベースの取引所理論  
~教育編~

平成 17 年 1 月 3 1 日

指導教官 有賀 裕二 教授

氏 名 柳生 祐佳

学籍番号 01C1107030G

# 目次

## 第1章 序論

- 1.1 背景と目的
- 1.2 方法と意義
- 1.3 論文の構成

## 第2章 U-Mart プロジェクトの教育プログラムと利用目的

- 2.1 はじめに
- 2.2 U-Mart の概要
- 2.3 大学での教育事例
  - 2.3.1 大阪産業大学経済学部での教育例
  - 2.3.2 中央大学商学部での教育例
- 2.4 おわりに

## 第3章 U-Mart プロジェクトの位置づけと周辺領域との関係

- 3.1 はじめに
- 3.2 U-Mart における教育の位置づけ
  - 3.2.1 工学・情報系分野での利用
  - 3.2.2 経済学系教育への利用
  - 3.2.4 中央大学商学部での利用
  - 3.2.4 U-Mart サマースクール
- 3.3 人工市場
  - 3.3.1 研究分野
  - 3.3.2 人工市場とは
  - 3.3.3 目的
  - 3.3.4 何ができるのか
  - 3.3.5 人工市場でできないこと
  - 3.3.6 研究の理論的背景
  - 3.3.7 関連する研究分野
- 3.4 おわりに

## 第4章 研究プログラムの実施

4.1 はじめに

4.2 エージェントのプログラム

4.3 エージェントの解説

## 第5章 結論

謝辞

参考文献

# 第 1 章

## 序論

### 1.1 背景と目的

経済学と工学、情報科学の学際的な研究プロジェクトとしてユニークなものに U-Mart プロジェクトがある。U-Mart は先物市場のシミュレータであり、人間と人工知能がコンピュータ上に表現された仮想市場で相互に取引を行うことを可能にしている。現在、U-Mart は教材として多くの大学で利用されている。

工学・情報系分野では、予測、適応、学習、最適化などのアルゴリズムの実習を教育目的としている。また、経済・経営分野では、金融派生商品の基本概念をシミュレーションにより学習し、従来の経済学、金融工学などのアプローチと比較研究することが主な研究目的である。

本論文では、U-Mart を教材として活用した教育事例を紹介し、利用例を分類する。また、周辺領域との関係性について調査し、今後の U-Mart の利用方法の可能性を述べる。

### 1.2 方法と意義

本論文では、過去の U-Mart プロジェクトの教育イベントや、ゼミなどで教材として扱われた教育事例を取り上げ、各事例を、1) 情報・工学系分野での利用、2) 経済・経営分野での利用に分類し考察を行う。さらに、中央大学商学部での事例を 1)、2) の視点から位置づける。さらに人工市場についても言及する。

### 1.3 論文の構成

第1章は「序論」である。

第2章は「U-Mart を用いた教育事例と利用目的」であり、U-Mart の教育事例と各事例における利用目的を紹介する。

第3章は「U-Mart プロジェクトの位置づけと周辺領域との関係」であり、U-Mart プロジェクトと人工市場、進化経済学、複雑適応系などの周辺領域との関係を調査し、U-Mart プロジェクトの位置づけを明確にする。

第4章は「研究プログラムの実施」であり、学習プログラムを実施した結果について考察する。

第5章は「結論」である。

## 第2章

# U-Mart プロジェクトの教育プログラムと利用目的

### 2.1 はじめに

第2章では、U-Mart プロジェクトによる教育活動の報告を参考にし、U-Mart プロジェクトの教育プログラムと利用目的を、1) 工学・情報系分野、2) 経済・経営学系での利用に分類し、大阪産業大学経済学部での利用事例と、中央大学商学部での利用事例を紹介する[1]。

近年では、コンピュータで社会的なシステムをシミュレーションすることにより、分析、考察を重ね、制度の設計に生かそうとする研究が人工市場、人工社会と呼ばれ脚光を集めている。このような研究では、良質なモデルやシミュレーション環境を研究の材料として蓄積、構築していき、研究者の間で共有していくことが大変重要となってくる。

この章では、研究用のテストベッドとして仮想の先物市場システムを開発するU-Mart プロジェクトやそれを利用した教育事例について述べたいと思う。

### 2.2 U-Mart の概要

「U-Mart ( Unreal Market as an Artificial Research Testbed )」とは、毎日新聞が発表する J30 を原資産とし、LAN やインターネットを通してヒューマン・エージェントとマシン・エージェントとが同時に売買に参加することができる仮想的な先物市場のシステムとその開発、研究といった活動をふくめたプロジェクトの総称である。

U-Mart は、金融市場を始めとした経済・社会システムの動向や、そこで活動する経済自体の行動に興味がある研究者が一緒に利用できる共通のテストベッドを提供することを目的に開発されてきた。

では一体、U-Mart は、どのような特徴を持っているのだろうか。先物市場とは、売り手と買い手が、将来の一定の時期において、あるモノを現在の時点で約定した価格で、受け渡すことを約束する取引である。ゆえに取引者は売買の決断を現物市場での価格や

将来を予測することにより判断することになる。以上のように現実にはない仮想の市場を構成することで、仮想市場での固有の取引や価格を形成することができ、現実の市場における複雑極まりない判断を少しでも再現することができるよう期待されている。

また、U-Mart は、コンピュータ・プログラムが自らの売買を判断するソフトウェアを組み込んだマシンエージェントと人間との両方の参加を可能にした。つまり同等に成果を競わせているので、エージェント・シミュレーションとゲーミング・シミュレーションの両方のアプローチが可能となり、まさにヒトとアルゴリズムが会う人工先物市場といえるだろう。その上、より実際の市場に近い制度をシミュレートした上で、多様な取引戦略をベースに市場を仮想的に実験することで、結果を解剖学的に細かく分析することができるようになっている。

## 2.3 大学での教育事例

U-Mart は下記の表にあるとおり、工学、情報、経済、経営学系のいくつかの大学・大学院での教育に用いられている。

### 工学・情報系

大学名	学部
徳島大学	工学部
防衛大学校	情報工学科
東京工業大学	大学院 総合理工学研究科
東京大学	工学部

### 経済・経営学系

大学名	学部
京都大学	経済学部 大学院 経済学研究科
大阪市立大学	経済学部
大阪産業大学	経済学部
筑波大学	大学院 経営政策科学研究科
中央大学	商学部
近畿大学	経済学部
千葉工業大学	社会システム学部

表 1 : U-Mart を用いた教育の実施例

ここでは、まず実際に他校、特に経済・経営学系の大学ではどのような教育がおこなわれているのかみていきたい。

### 2.3.1 大阪産業大学経済学部教育例

U-Mart は、大阪産業大学経済学部における演習「金融投資とコンピュータ・ネットワーク」で利用された実績がある[2]。演習に参加した人数は 20 名であり、1 年間を通して、下記の(1)から(4)の流れで演習を行った。

- (1) 証券市場の基礎学習
- (2) 体験学習
- (3) 研究発表
- (4) U-Mart 実験の実施

#### (1) 証券市場の基礎学習

まずはじめに、指導教授が演習用に作成した資料(2001年進化経済学会・福岡大会で配布)を使用し、証券市場に関して全般的に学習する。経済学部の学生とはいえ、全員が十分な知識があるとは限らない。ゆえに丁寧な講義を行い、証券市場についての理解を深める。しかし、あまりに丁寧にやりすぎると、半分しか消化できない場合があるので注意しなくてはならない。

#### (2) 体験学習

自分が学んだことがどれほどのものか、また自分には何が足りないのかを知るためにも、今まで得た知識を生かし、インターネット上で仮想投資を行なってみる。このとき利用するのが野村証券の「バーチャル株式投資クラブ」(<http://www2.nomura.co.jp/vstock/VirtualServlet>)というものである。これは、参加エントリーをすると仮想の所持金100万円が支給され、その100万円を元手に自由に株式の売買シミュレーションを行うことができるというものだ。4月と10月に新規スタートする。約14万人が参加している(2005年1月22日現在)。

参加者全員の中での順位も分かると同時に、参加ゼミ学生内の順位を知ることができる。よって株価を通して、企業や産業に関する具体的な知識が身につき、かつ金融取引全般への導入として利用することができるといえるだろう。

#### (3) 研究発表

夏季合宿を利用し、学生が各々の興味に基づき、金融、株式、インターネットなど、演習に関連したテーマで研究発表を行う。

#### (4) U-Mart 実験の実施

学部の PC ラボ教室を実験環境として使い、後期の 2 月と 4 月の計 4 回、時間は各回授業 2 コマ連続で U-Mart の実験を行なった。

初回は、スタンド・アローンで GUI クライアントの操作方法を学習し慣れさせる。マシン・エージェントはあらかじめ作成されたものが参加した。

実験を行なった結果として、毎回実験のたびに事前に準備をするのは時間と手間がかかるため、実験には、U-Mart 実験専用のサーバーを用意したほうがよいことが判明した。

なお、[2]によると、U-Mart サーバー専用のマシンを準備するには、10 万円以下の費用でできるため、経済系の研究室でも準備可能である。

実験実施後にアンケート調査を行っている。学生の反応は下記の通りであった。

- ・ 野村証券の「バーチャル投資クラブ」と比べて、テンポが速く、値動きの形がよく分かりやすい。
- ・ 取引を実感することができる・取引の結果がすぐにわかる為、次の相場がどうなるのか楽しみである。
- ・ テレビゲームのようであり取り組みやすい・一日がすぐに経つので、状況判断が難しい。
- ・ 取引主体の数が少ないと、指値売買が難しい。
- ・ 破産覚悟でなら価格操作ができる。しかし、簡単に破産してしまうので自分では株に手を出さないようにしようと思った。
- ・ 株の売買はギャンブル性が多く含まれており、また取引には色々な戦法があることを実感することができた。

上記の結果から、「証券市場に関する基礎的な学習」、「インターネット上のバーチャル投資」、「U-Mart 実験」という手順を踏んだ演習は、従来のテキストの輪読、発表という形式に比べて立体的になったことがわかる。

#### 2.3.1 中央大学商学部の教育例

次に、中央大学商学部における教育事例を述べていきたいと思う。中央大学商学部における演習名は、「経済学とコンピュータ：市場実験」であり、演習に参加した人数は 10 名程度であり、1 年間を通し、下記の (1) から (6) の流れで演習を行った。

- ( 1 ) スタンドアローン版による U-Mart 体験
- ( 2 ) テキストを用いた取引所理論の学習
- ( 3 ) マシンエージェントの作成
- ( 4 ) U-Mart 国際大会への参加
- ( 5 ) 複雑適応系の外国書購読
- ( 6 ) U-Mart 国内大会への参加

#### ( 1 ) スタンドアローン版による U-Mart 体験

U-Mart 概要を把握するために、スタンドアローン版を用いて U-Mart の仕組みを体験した。この際、ゼミの上級生から体験前に 20 分程度の指導を受けた。

#### ( 2 ) テキストを用いた取引所理論の学習

経済学のテキストを用いて取引所理論について学習・発表を行い、U-Mart における価格決定の仕組みを理解した[3]。

#### ( 3 ) マシンエージェントの作成

上級生の指導の元、プログラミングの基礎知識とマシンエージェントの作成方法について 2 ヶ月程度の指導を受けた。その際、過去に上級生が作成したマシンエージェントのソースコードを読んだ。また、上級生が作成したエージェント作成マニュアルを用いてプログラミング作法を学習した[4]。マシンエージェントの完成が早かった学生は、U-Mart のサマースクールに参加し、取引所サーバの設計・実装を行った。

#### ( 4 ) U-Mart 国際大会への参加

2003 年夏に「かずさアカデミアパーク」で行なわれた国際学会 ISAGA2003 における U-Mart 国際大会にマシンエージェントを出展し、さらにヒューマンエージェントとしても参加した。また、同学会を契機として千葉県木更津市で合宿を行った。

#### ( 5 ) 複雑適応系の外国書購読

後期の授業が始まり、気分を変えるために複雑適応系の外国書購読を行い、市場の複雑さを解明する基礎となる理論を学習した[5]。

## ( 6 ) U-Mart 国内大会の参加

2004 年 10 月、東京工業大学にて U-Mart の国内大会に参加するとともに、近畿大学の学生と合同ゼミを行った。

## 2.4 おわりに

第 2 章では、大学における具体的な教育事例を見てきた。自身の経験から、U-Mart に取り組み始めたばかりの頃は、何をしたらよいのか分からず、モチベーションが低かった。しかし、教育の蓄積を重ねるにつれ、取引実験の成績や結果がよくなっていき、徐々に「面白さ」を感じはじめ、取り組む意欲が上がっていく。だが、そのうち単に取引実験で前回より儲けたり、または勝つだけでは物足りなくなってくる。そして、現実とのギャップやプロジェクトそのものの目的が分からなくなる等の問題が出てくる。そのような数々の壁を乗り越えるためにも、文系学生にはきめ細やかな、きちんとステップを踏んだ教育をしていかななくてはならない、ということが分かった。

## 第3章

# U-Mart プロジェクトの位置づけと周辺領域との関係

### 3.1 はじめに

第3章では、U-Mart プロジェクトにおける教育活動の報告を参考に、U-Mart プロジェクトの位置づけと周辺領域との関係について、まず、工学・情報系と経済学系分野、有賀ゼミにおける各々の位置づけについて言及し、人工市場、進化経済学、複雑適応系などの周辺領域との関係を多角的に調査した上で U-Mart プロジェクトの位置づけを明確にする[1]。

### 3.2 U-Mart における教育の位置づけ

U-Mart プロジェクトでは、研究・教育・イベントの3つの活動を行っている。

研究：U-Mart のシステム開発とそれを利用した人工市場の研究

教育：U-Mart のシステムを用いた計算機科学や経済学などの教育

イベント：U-Mart のシステムを使い、参加者を募り、コンテスト形式で行われるエージェント、ゲーミング・シミュレーションの公開実験

プロジェクトの活動を通じて、上記の3種類の活動は相互に関連性を持っている。つまり、研究に研究を重ねて構築されたシステムが教材として利用されると同時に、教育に利用するためのさらなる高品質化を目指し、マニュアルやテキストなどの文書が出来上がっていく。そして、教育現場で利用され、実験経験が蓄積されるにつれ、研究への参加者も増えていくため、教育のニーズにも応えるような研究に有用な開発・分析ツールも蓄積されるのである。また、研究と公開実験の開催により、多種多様な取引戦略

をもつエージェントやその開発手法などの知見が蓄積される。さらに教育で利用する際、公開実験に参加することで学生レベルでの大学間交流や学際的交流の機会に恵まれることにもなる。

### 3.2.1 工学・情報系分野での利用

U-Mart は工学、情報系の分野の教材としては、学部レベルでの初級プログラミングの課題や大学院レベルでのエージェント・シミュレーションの実習、大学院レベルでの予測・学習・適応・最適化・アルゴリズムの実習、また工学系の学生への経済学や金融の紹介などに有効であるとされている。

特にプログラミングの課題としては、知識や技術のレベルに応じた課題、目標を設定することや、自分のオリジナル案をプログラムのコードとして実現すること、事前の予想をもとに作成したプログラムを実際の複雑な環境で動作の確認をする、といったように主体的に学習できることが通常のプログラミングの演習とは異なる点である。

また機械系の学科で行われているロボットコンテストのように、学習のモチベーションを高めるためにコンテスト形式のイベントを開催している。

### 3.2.2 経済学系教育への利用

経済学系(経済学、経営学、商学などの学部、大学院を指す)の分野において、U-Mart は学生自身が公開実験等で提供されたソフトウェアのエージェントを利用した、市場取引に参加するゲーミング・シミュレーションに用いられている。

U-Mart は経済学系の教材としては、各自のエージェント戦略や行動の反応により価格が形成されるという基本的な概念を、実験に参加することで気軽に体験することができる。また、個々のエージェントの戦略や行動が市場に与える影響を短い時間で体験することができ、取引実験はテレビゲームのようで競技性があるため、学生の学習の動機付けがしやすい。つまり、金融市場、とりわけ金融先物市場の制度や考え方は記述だけでは分かりにくいのが、これを体験することにより、雰囲気がかみやすくなる、という効用がある。さらに、実験のログデータを保存したり、実験後にアンケートをとることにより、客観的な情報と学習参加者の主観的な情報を比較し、考察を重ねることにより、今後につなげることも可能である。

### 3.2.3 中央大学商学部での利用

中央大学商学部では、経済学を学び、マシンのアルゴリズムをも学ぶことにより、自分の思い通りの戦略をマシンに組み込み動かすことができる。かつ、的確に自分でマ

シンエージェントの意思決定履歴（未実現損益の時系列データ、ポジション管理）を U-Mart を用いているため、ログデータから入手、分析できる（詳しくは、エージェント取引所理論である「トレーニング編」と「デイトレード編」を参照すること[6][7]）。さらに、マシンの実装が早期に完成し、プログラミングに関して興味をもてた学生は U-Mart サマースクールに参加し、さらに複雑なアルゴリズムであり、取引所サーバー本体の設計、実装に関して学習することができる。経済学にコミットし、さらにプログラムを書くという労働集約的なクリエイティブな領域にも、文系の立場からコミットし双方向的な立場をとることで、学際的な創造性を発揮することが有賀ゼミの魅力といえる。

### 3.2.4 U-Mart サマースクール

我がゼミでは、さらなるスキルアップを目指し、より自在に自分の戦略をマシンに組み込めるよう、またより深く U-Mart を理解できるよう、毎年、金銭的、時間的に余裕のあるゼミ員は「U-Mart サマースクール」というものに参加させていただいている。

U-Mart などのエージェント・シミュレーションにおいては、複雑なソフトウェアが必要である。しかし、そのためのソフトウェアや工学的な開発手法は大学教育ではあまり紹介されていない。ゆえに、U-Mart プロジェクトでは、人材育成も含めて、この種の研究に取り組む学生の技術向上を目指した、U-Mart サーバーの設計をテーマにオブジェクト指向のソフトウェア設計を学ぶサマースクールを 2002 年から毎年、開催している。ここでは、U-Mart プロジェクト教育の活動報告や実際の参加者の話を参考に、どのようなことを行っているのか、以下の流れで述べていきたい。

- (1) サマースクールの学習目標
- (2) カリキュラム(1～5日目)

U-Mart サマースクール		
年	開催地	
2002	東京工業大学	大学院総合理工学研究科
2003	京都大学	学術情報メディアセンター
2004	公立はこだて未来大学	

表 2 : U - Mar t サマースクールの実施例

- (1) サマースクールの学習目標

1. エージェント・シミュレーションの基本的なプログラミングの知識として、「ク

クライアントサーバーモデル」、「TCP/IP による通信プログラミング」、「スレッドを用いた並行プログラミング」をマスターする。

2. オブジェクト指向ソフトウェア分析、設計、実装、コード化の慣習、デザインパターン、メソッドのモジュール化、プログラムのユニットテスト、UML によるプログラマ間のコミュニケーション、アジャイル開発の考え方、といった大規模プログラムの開発手法を学ぶ。

3. 具体的な人工市場システムの実装を通じて、上記の要素技術をオブジェクト設計についての利用経験を得る。

## (2) カリキュラム

### 【1日目】(午後)

解説：U-Mart 入門

実習：U-Mart による先物取引実験、U-Mart システムの取引所サーバーの仕様の理解と分析

### 【2日目】(午前・午後)

解説：オブジェクト試行設計と関連するソフトウェア開発技法の紹介

(コード化の慣習、UML、テストファーストアプローチ、デザインパターン、リファクタリングなど)

実習：事前課題の提出とコードの相互レビュー、コードの改良と完成

### 【3日目】(午前・午後)

解説：スレッドを用いた平行プログラミングとソケットを用いたネットワークプログラミング

実習：スレッド、ソケットを用いたプログラミング課題の演習、取引所サーバーの設計

### 【4日目】(午前・午後)

実習：取引所サーバーの設計

### 【5日目】(午前)

実習：取引所サーバーの報告とディスカッション

実習は5人程度のグループに分かれて行い、まず、U-Mart サーバーの主要な3つの構成要素（クライアントからのコマンドの処理、値付け、資金移動と口座の管理）の設計を行う。

次に、各要素を学んだ受講生を含むようにグループを再編成し、サーバー全体のシナリオの確認から、オブジェクトの候補を抽出し、ボトムアップに組み上げてゆく形で行なわれる。各グループは適宜、講師によるレビューを受けるとともに、全員の前でプレゼンテーションを行う。

### 3.3 人工市場

これまでは市場参加者の個人的な特性の違いや心理的な側面は無視されていた面があった。が、近年、激しく変動する株式市場や外国為替市場において、市場に参加している人間の心理的な効果が非常に注目されるようになってきている。このような現状から、より現実に近い市場モデルを構築することを目指した人工市場(artificial market)研究と呼ばれる手法が登場した。では、この人工市場とは一体何なのだろうか[8]。順を追って考察していきたい。

#### 3.3.1 研究分野

人工市場は、複雑系アプローチと金融市場理論の両方の分野に属する。主に使用されている手法や分析対象は、計算機科学、数理科学、社会科学、そして行動科学の研究領域と関連する。

#### 3.3.2 人工市場とは

人工市場とは、その言葉を見てもわかるように、計算機上に人間の手で人工的に作り出された架空の市場のことである。人工市場に参加しているのはエージェントと呼ばれ、計算機のプログラムによって表現された、まさに仮想的なディーラーといえる。もちろん、生身の人間がエージェントに混じり人工市場での取引に参加する場合もある。人工市場には現実の金融市場を反映した価格決定メカニズムがあり、各エージェントの投資行動が集まり、蓄積されることによって人工市場の上で仮想的な価格が決定される。つまり、人工市場を簡単に言うと、「人工市場 = エージェント + 価格決定メカニズム」ということができるだろう。ゆえに、金融市場で取引するようなエージェントがどんなに沢山集まった計算機プログラムでも、そのプログラムに価格決定メカニズムが含まれていなければ、それは人工市場とはいえないのである。

### 3.3.3 目的

人工市場の目的は、市場参加者による投資や意思決定に関する行動が集積して、金融バブルなどの市場全体でみられるある現象が出現する仕組みの解明を行なうことだ。その仕組みを知るためにも、価格決定メカニズムはモデルの中心部を支えるとても重要な構成要素といえよう。工学的な予測のプログラムとは異なり、人工市場が価格決定のメカニズムをもつことは必須条件なのである。

### 3.3.4 何ができるのか

人工市場を使うことによって、主に「現実の市場分析」、「現場の支援ツールの構築」、「既存の経済理論の検証」の3つのことができる。

### 3.3.5 人工市場でできないこと

人工市場の目的は、市場のメカニズムを解明することであり、市場の内部構造を考慮せずに結果がすべてであるような予測を目的とはしていない。

### 3.3.6 研究の理論的背景

人工市場モデルがもつ大きな特徴として挙げられるのが、市場参加者どうしの影響の及ぼしあいに重点をおいてモデルをつくっているところである。そのため、人工市場モデルの基本的要素「エージェント」、「マルチエージェント」、「戦略」、「学習」、「相互作用」、「ミクローマクロ関係」という6つの基本的要素は、すべての研究において共通しているといえるだろう。

#### 「エージェント」

これはもともとコンピュータ科学の分野では、仮想世界の中でルールを持って自律的に行動する基本単位のことを指している。しかし、人工市場モデルでは、ルールをもって投資や予測・学習といった行動をとる、仮想的なディーラーをエージェントと呼んでいる。

#### 「マルチエージェント」

この用語もコンピュータ科学の分野の用語だ。しかしここでは、仮想的なエージェントが複数集まったところに市場のメカニズムが導入された人工市場全体を示している。

#### 「戦略」

意思決定論の分野で使われている意味に近く、ある目的のためにエージェントがとる行動やその行動のもととなる方針のことを指している。人工市場モデルでは、各仮想ディーラーが利得追求という目的のために行なう、投資行動や将来の金融価格の予想行動を示す。

#### 「学習」

人工市場モデルをより現実世界に近いモデルにするためにも、時間が経過するにつれて各仮想ディーラーが自分の目的にかなった戦略に修正していく能力が必要である。そのような適応行動をエージェントの学習行動とよぶ。

#### 「相互作用」

人工市場モデルにおけるエージェントどうしの影響の及ぼし合いを相互作用とよぶ。例えば、ある仮想ディーラーの売買行動が、市場全体の需要と供給の関係や金融価格の動きを通して、他の仮想ディーラーの売買に影響を与えることがある。このほかにも、ある仮想ディーラーが他の仮想ディーラーの予想方式や投資戦略に影響を受けて、自身自身のルールを学習していくことも相互作用の例といえよう。

#### 「ミクローマクロ関係」

人工市場モデルでは、個々のエージェント、つまりミクローなレベルにおいて直接は意図されていなかったが、人工市場全体のマクローなレベルにおいて予想もしなかったような特徴が現れてくることがある。金融価格におけるバブルという現象は、意図して特定の仮想ディーラーが現れるのではなく、複数の仮想ディーラーの相互作用によって現れる現象である。このようにミクローなエージェントのレベルの行動に簡単に還元できないようなマクローな現象が現れてくる関係をミクローマクロ関係と呼ぶ。

### 3.3.7 関連する研究分野

人工市場について理解を深めるために、この節では関連する研究分野について簡単に説明していこうと思う。

まず、人工市場は「社会科学」、「計算機科学・数理科学」、「行動科学」の3つの研究領域の融合である。その研究分野の中にも、大きく分けて2つの流れがある。

1つは近年のコンピュータの処理能力の向上に伴って現れた「複雑系アプローチ」と呼ばれ、人工知能や人工生命、人工社会が含まれるという流れがある。人間の知能や自然の生命現象、現実の社会現象といった多数の要素が相互に絡み合った現象についての

理論モデルを構築する場合、従来の手法では分析できないほど複雑なモデルとなる。そこで、計算機シミュレーションを使い理論モデルの振る舞いを分析し、研究対象のメカニズムを解明するというアプローチをとる。

2つ目は、新たな金融市場の理論モデルを求めていく流れである。これには経済物理や実験経済学、行動ファイナンス、マーケットマイクロストラクチャー理論などが含まれる。既存の計量経済学における金融市場モデルでは、市場参加者の合理性と呼ばれる非現実的な仮定がされていたため、いくつか説明するのが困難な現象があった。この合理性の仮定の代わりに、行動科学などから得られた市場参加者の新たな仮定を導入して、新しい市場理論を構築しようとする流れである。

人工市場は、以上の2つの大きな流れの接点にある研究分野であるといえよう。以下、順にそれぞれの研究分野を説明していく。

### *[複雑系アプローチ]*

#### (1) 人工知能

人間の知能の動きを解明するために、計算機科学の手法と行動科学の知見を用いて、人間の思考過程の中で行なわれていると思われる操作を計算機のプログラムによって再現していこうとする研究分野のこと。人工市場が金融市場のモデルをつくるように、人工知能は人間の知能モデルをつくっているのだ。

人工市場における仮想エージェントの学習行動や行動決定を記述するために、人工知能分野にあるモデルを用いていることが多い。たとえば仮想ディーラーの学習行動や行動決定を記述するために、人工知能分野にあるモデルを用いることが多々ある。例えば、仮想ディーラーの学習行動を記述するために、遺伝的アルゴリズムや遺伝的プログラミング、ニューラルネットワークなどの、元々は人工知能分野にあった学習手法を用いている人工市場モデルがある。

#### (2) 人工生命

人工生命は、自然界の生命現象を計算機上に再現・構成することにより、生物の特徴的な振る舞いを示す人工的なモデルを用いて、生命現象の分析や解析を行う研究分野である。

研究対象は遺伝や進化などの生命科学領域に含まれ、対象自体は人工市場とは関連がないが、計算機科学の手法を用いている点は共通している。

#### (3) 人工社会

人工社会とは、その名前からわかるように、計算機上に現実にある社会現象にみられる特徴的な振る舞いを示すような計算機モデルを構築することである。この計算機モデルのシミュレーションにより、特に社会集団の移住や意見形成などの社会現象の分析を行っている。経済現象も社会現象の一種なので、人工社会研究に含まれている。

### *[新しい市場理論]*

#### (1) 経済物理学

現実世界の経済現象に対して、物理学にあるモデルや手法を当てはめて解析しようという研究分野のことである。例えば、金融市場での市場参加者の同調現象の解析に、物理学における固体レーザーの挙動を示すモデルを使ったり、企業所得の規模の分布を解析するために、岩石やガラスを破壊した時の破片の大きさの分布のモデルを用いたりしている。

経済物理学の研究対象は経済学の分野に含まれる現象である。経済物理学のモデルでは、各要素の振る舞いを詳細な計算機プログラムとして記述することよりも、モデル全体の振る舞いを数式などで記述して数理的に解析することに重点がおかれていることが多い。

経済物理学の目的は、経済現象の各参加者をより現実的に記述するというよりも、経済現象全体の振る舞いの法則性を見つけることにある。ゆえに、個人の行動をモデル化するために、行動科学の知見を用いるということあまり行われない。

#### (2) 実験経済学・行動ファイナンス

心理実験のように、条件が規制された実験室の中で経済的状況を再現し、人間が参加する実験を行うものを実験経済学という。例えば、実験室の中で参加者に架空の資産を与え、いくつかの条件で様々な投資選択を行ってもらう実験市場などがある。

従来の計量経済学で用いてきた経済の実証データには、さまざまな環境要因や攪乱要因が含まれているので、分析が困難な場合があった。条件が純粋に操作されて雑音が除かれている実験データを分析し、人間の行動特性と経済現象の間の関連を探るのが実験経済学の目的である。

人工市場とは、研究対象および市場参加者の行動を行動科学の知見と手法によって分析を行っている点が共通している。ただ、人工市場ではさらに市場参加者のモデルを集積して金融市場全体のモデルを計算機上に構築し、シミュレーションによって市場全体の分析も行うが、実験経済学や行動ファイナンスあくまでも生身の人間の実験データの

分析を中心としている。

### ( 3 ) マーケットマイクロストラクチャー理論

人間の行動決定について効用関数などの数式を用いてモデル化を行い、金融市場における取引制度や情報の流通がどのように金融価格形成に影響を与えているのかを、意思決定論やゲーム理論の知見を用いて数理的に分析するのがマーケットマイクロストラクチャー理論である。

### ( 4 ) 金融工学

金融価格の動きを数学の確立仮定で表し、様々な金融取引のリスクを計算するための研究分野が金融工学である。いわゆるデリバティブとよばれる金融派生商品の開発も、金融工学の手法をもとに行われている。経済物理学と同様に、モデルの振る舞いを数式で記述し、その計算のみを計算機に行わせているので、計算機科学よりも数理的な手法を用いている。ゆえに人工市場とは異なって、行動科学領域との関連はほとんど無いといえるだろう。

## 3.4 おわりに

この第 3 章では、それぞれの分野における U-Mart の位置づけ、さらには筆者が属する中央大学商学部における位置づけはどうか、ということについて順を追ってみてきた。また、U-Mart の根底にある人工市場とはどのようなものなのか、周辺領域との関係についても深く言及してきた。このことにより、U-Mart が実に様々な理論をバックグラウンドにもち、かつ相互作用による学際的なプロジェクトであることが分かっていただけたら幸いである。

## 第 4 章

### 研究プログラムの実施

#### 4.1 はじめに

この章では、実際に学んだ知識をいかし、かつ諸先輩方やゼミ員の助けを借り、試行錯誤しながら実際に作りあげ、筆者が東京工業大学にて行われた U-Mart の国内大会にて、発表したエージェントについて言及していきたい。

#### 4.2 エージェントのプログラム

```
package strategy;

import java.util.*;
/**
 * Relative strength Index(Stochastics Strategy)
 *
 * author Yuka Yagyū(Chuo University)
 */

public class TestStrategy extends Strategy {

    private Random random;
    private final int widthOfPrice=60;
    private final int maxQuant=50;
    private final int minQuant=10;
    private final int maxPosition=300;
    private final int nominalPrice=3000;
```

```

public TestStrategy(int seed) {
    random = new Random(seed);
}

public Order getOrder(int[] spotPrices, int[] futurePrices, int pos,
                    long money, int restDay){

    Order order = new Order();
    int prevPrice = getLatestPrice( futurePrices );
    if ( prevPrice == -1 )
        prevPrice = getLatestPrice( spotPrices );
    if ( prevPrice == -1 )
        prevPrice = nominalPrice;
    int latestSpotPrice = spotPrices[119];
    int latestFuturePrice = futurePrices[59];

    if ((latestSpotPrice <= 0) || (latestFuturePrice < 0)) {
        order.buysell = Order.NONE;
        return order;
    }

    double k = getK(spotPrices,0);
    double d = getD(spotPrices);
    double prevK = getK(spotPrices,1);
    double prevD = getPrevD(spotPrices);

    if( (prevK < prevD) && (k > d) ) {
        order.buysell = Order.BUY;
    }

    if( (prevK > prevD) && (k < d) ) {
        order.buysell = Order.SELL;
    }

    if(order.buysell == Order.BUY) {

```

```

        if( pos > maxPosition ) {
            order.buysell = Order.NONE;
            return order;
        }
    } else if ( order.buysell == Order.SELL ) {
        if(pos < - maxPosition) {
            order.buysell = Order.NONE;
            return order;
        }
    }
}

while (true) {
    order.price = prevPrice + (int)(widthOfPrice *
random.nextGaussian() );
    if ( order.price > 0 ) break;
}

order.quant = minQuant + random.nextInt (maxQuant - minQuant + 1);
return order;
}

//過去の最高価格の設定
public int getHighestPrice(int[] theArray, int theStartIndex, int
theEndIndex) {
    int highestPrice = 0;
    for(int i = theStartIndex; i <= theEndIndex; i++) {
        if (highestPrice < theArray[i]) {
            highestPrice = theArray[i] ;
        }
    }
    return highestPrice;
}

//過去の最低価格の設定
public int getLowestPrice(int[] theArray, int theStartIndex, int
theEndIndex) {

```

```

        int lowestPrice = Integer.MAX_VALUE;
    for(int i = theStartIndex; i <= theEndIndex; i++) {
        if (lowestPrice > theArray[i]) {
            lowestPrice = theArray[i] ;
        }
    }
    return lowestPrice;
}

```

//K の計算

```

public double getK(int[] spotPrices, int t) {
    int highestPrice = getHighestPrice(spotPrices, 46 - t, 59 - t);
    int lowestPrice = getLowestPrice(spotPrices, 46 - t, 59 - t);
    double k = (double)(spotPrices[59 - t] - lowestPrice) / (double)(highestPrice -
lowestPrice);
    return k;
}

```

//D の計算

```

public double getD(int[] spotPrices) {
    double k0 = getK(spotPrices,0);
    double k1 = getK(spotPrices,1);
    double k2 = getK(spotPrices,2);
    double d = (k0 + k1 + k2) / 3;
    return d;
}

```

//prevD の計算

```

public double getPrevD(int[] spotPrices) {
    double k1 = getK(spotPrices,1);
    double k2 = getK(spotPrices,2);
    double k3 = getK(spotPrices,3);
    double prevD = (k1 + k2 + k3) / 3;
    return prevD;
}

```

```
}
```

### 4.3 エージェントの解説

```
public class TestStrategy extends Strategy {  
    private Random random;  
    private final int widthOfPrice=60;  
    private final int maxQuant=50;  
    private final int minQuant=10;  
    private final int maxPosition=300;  
    private final int nominalPrice=3000;
```

これは、「先物価格が一回の板寄せで、平均して60円、最大注文量 = 50、最小注文量 = 10、最大ポジション = 300、ノミナルプライス = 3000」ということを表している。

```
public Order getOrder(int[] spotPrices, int[] futurePrices, int pos,  
    long money, int restDay){
```

これで、配列である、現物価格、先物価格、ポジション、保有現金、残日データ郡を閲覧することができる。

```
int latestSpotPrice = spotPrices[119];  
int latestFuturePrice = futurePrices[59];
```

現物価格は120回、先物価格は60回分のデータが格納されてる。

現物価格 : Spot Price[0]~ Spot Price[119]

先物価格 : FuturePrice[0]~FuturePrice[59]

を閲覧できることになり、数字が大きいほど、現在に近いデータなので、spotPrices[119]で前回板寄せ時の現物価格、futurePrices[59]で前回板寄せ時の先物価格となる。

```
if ((latestSpotPrice <= 0) || (latestFuturePrice < 0)) {  
    order.buysell = Order.NONE;  
    return order;  
}
```

```

latestSpotPrice <= 0 または latestFuturePrice < 0 のときは注文を行わない
double k = getK(spotPrices,0);
double d = getD(spotPrices);
double prevK = getK(spotPrices,1);
double prevD = getPrevD(spotPrices);

```

のちに売買判断を行う際に必要な%K、%D、prev%K、prev%Dが現物価格を取り出せるようにしている

```

if( (prevK < prevD) && (k > d) ) {
    order.buysell = Order.BUY;
}

if( (prevK > prevD) && (k < d) ) {
    order.buysell = Order.SELL;
}

```

これは、私の売買判断です。

ファスト・ストキャスティクスという

%Kが%Dを下から上へ抜いた地点を買いサイン（図1参照）

%Kが%Dを上から下へ抜いた地点を売りサイン（図2参照）

という作戦。

図 1 :

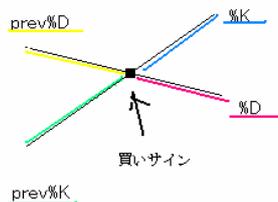
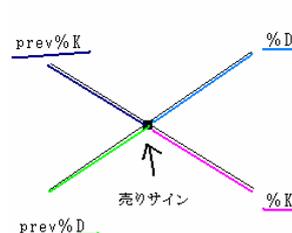


図 2 :



```

if(order.buysell == Order.BUY) {
    if( pos > maxPosition ) {
        order.buysell = Order.NONE;
    }
    return order;
}

```

```

    } else if ( order.buysell == Order.SELL ) {
        if(pos < - maxPosition) {
            order.buysell = Order.NONE;
            return order;
        }
    }
}

```

ポジション管理をしている。

もし売買判断が「買い」でも、最大ポジションが300を超えたら、注文を行わない。

```

if(order.buysell == Order.BUY) {
    if( pos > maxPosition ) {

```

という条件が false かつ「売り」で、ポジションが-最大ポジションより小さいければ、注文は行わない。

```

while (true) {
    order.price = prevPrice + (int)(widthOfPrice *
random.nextGaussian() );
    if ( order.price > 0 ) break;
}
order.quant = minQuant + random.nextInt (maxQuant - minQuant + 1);
return order;
}

```

random.nextGaussian は、平均1の「正規」分布だ。

意味は、「平均して、だいたい1くらいの値を返す」ということ。

注文価格は、前期価格 + 平均 widthOfPrice ということ。

平均して60円動くと予想している。

```

public int getHighestPrice(int[] theArray, int theStartIndex, int theEndIndex)
{
    int highestPrice = 0;
    for(int i = theStartIndex; i <= theEndIndex; i++) {
        if (highestPrice < theArray[i]) {
            highestPrice = theArray[i] ;
        }
    }
}

```

```

        return highestPrice;
    }

```

ここで、過去の最高価格の設定をしています。  
the Array とすることで、int の配列であれば、どのようなものでも、最高値を作れるようにしている。この Array は現物価格からとってきている。

```

public int getLowestPrice(int[] theArray, int theStartIndex, int theEndIndex)
{
    int lowestPrice = Integer.MAX_VALUE;
    for(int i = theStartIndex; i <= theEndIndex; i++) {
        if (lowestPrice > theArray[i]) {
            lowestPrice = theArray[i] ;
        }
    }
    return lowestPrice;
}

```

上記と同様、過去の最低価格の設定をしている。

```

public double getK(int[] spotPrices, int t) {
    int highestPrice = getHighestPrice(spotPrices, 46 - t, 59 - t);    . . .

    int lowestPrice = getLowestPrice(spotPrices, 46 - t, 59 - t);    . . .

    double k = (double)(spotPrices[59 - t] - lowestPrice) /
(double)(highestPrice - lowestPrice);
    return k;
}

```

これは K の計算式。

と は、ある一定期間内（私は、14日間としました。）の最高値と最安値が出るようにしている。

公式「(最新の終値 過去14日間の最安値) / (過去14日間の最高値 過去14日間の最安値)」を使い、Kを求める。

```
public double getD(int[] spotPrices) {  
    double k0 = getK(spotPrices,0);    . . .  
    double k1 = getK(spotPrices,1);    . . .  
    double k2 = getK(spotPrices,2);    . . .  
    double d = (k0 + k1 + k2) / 3;  
    return d;  
}
```

Kの式と同様に、Dを求めている。

で、今期、1期前、2期前のKの値を取り出している。

Dの求め方・・・Kの3日間の移動平均。

prevDも同じことをして出す。

## 第5章 結論

U-Mart、つまり人工市場研究とよばれる分野は、世界的にみても新しいアプローチであり、様々な学問、研究がバックグラウンドにあるため、文系でも理系の学生でも取り組める実に魅力的な学問だといえるだろう。また単に学ぶだけでなく、U-Martの活動を通して様々な分野で活躍している方々と交流すること、幅広く学問に触れることも、苦勞の連続で、実に大変だったが、非常に有益で普通の文系ゼミでは味わえない貴重な体験だったと思う。結果がどうであれ、とりあえず何事も挑戦してみることの大切さ、壁にぶちあたっても、努力して根気よく立ち向かうことの大切さを学んだ。

この論文は、エージェントベースの取引所理論と題し、「教育編」、「デイトレード編」、「トレーニング編」の三部構成になっている。もちろん一つ一つが独立して読むこともできるし、続けて読んでいただければ、より一層深く U-Mart について味わっていただくことができるだろう。

まだまだ多くの問題と可能性を秘めており、これからもどんどん新しい U-Mart モデル、人工市場モデル構築されていくであろう。そのような中で、少しでもこの論文がお役に立てれば幸いである。

## 謝辞

平成 17 年 1 月 31 日、本日をもって 2 年間の有賀ゼミにおける活動の集大成として無事に書き上げることができたのは、ひとえにこの方たちのおかげである。

有賀裕二教授（2 年間ご指導いただきまして本当に有難うございました。）

石山さんをはじめとする有賀ゼミの諸先輩方

有賀ゼミメンバー（阿部寛之、工藤寛之、新城拓也）

こころよく卒論を書き上げる場を提供してくださった出口研究室の皆様

深く感謝いたします。

## 参考文献

[1]喜多一、小野功、森直樹、佐藤浩、谷口和久、松井啓之、中島義裕、小山友介、石山洸、U-Mart 人工先物市場プロジェクトにおける教育活動（システム・情報部門学術講演会、講演論文集）、2004.

[2] 谷口和久、松井啓之、出口弘、五十嵐寧史、教育ツールとしての U-Mart：経済学教育での活用事例、進化経済学会大阪大会（関西大学）、2002.

[3]森嶋通夫、無資源国の経済学、岩波書店（岩波全書）、1984.

[4]石山洸、JAVA 演習（<http://c-faculty.chuo-u.ac.jp/~aruka/seminar.html>）

[5]JOHN HOLLAND、HIDDEN ORDER、PERSEUS BOOKS.

[6]工藤寛之、エージェント取引所理論～デイトレード編～、中央大学卒業論文集、2005

[7]新城拓也、エージェント取引所理論～トレーニング編～、中央大学卒業論文集、2005

[8]和泉潔、相互作用科学シリーズ・人工市場、森北出版株式会社、2003 .