

エージェントベースの取引所理論  
～ 解析編 ～

平成16年1月30日

指導教官： 有賀裕二教授

学籍番号： 00C1114014L

氏名： 原田圭

# 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 エクセルによる解析

- 2.1 はじめに
- 2.2 U-Mart のログデータの構造
- 2.3 エクセルによる解析
- 2.4 おわりに

## 第 3 章 Java による解析ツールの実装

- 3.1 はじめに
- 3.2 オブジェクト指向プログラミングの応用
- 3.3 解析ツールの実装と利用
- 3.4 おわりに

## 第 4 章 エージェントの意思決定と価格変動の考察

- 4.1 はじめに
- 4.2 エージェントの意思決定の考察
- 4.3 価格変動の考察
- 4.4 おわりに

## 第 5 章 結論

# 第 1 章 序論

## 1.1 背景と目的

私がこのゼミに入ゼミしたのは、他のゼミでは類をみないコンピュータを使うということが魅力的で当ゼミへの入ゼミを決意した。私が入ゼミした際には 3 年生(当時)は 4 人しかいなく、4 年生も 2 人ということで全体合わせて 6 人しかいなかった。だが、6 人という少人数のおかげでチームワークもよく、研究内容のほうも他のゼミにはないほど密度があり、それらの研究活動もスムーズに充実して進めていけたと思う。

この論文は、私が当ゼミでの 2 年間の研究のまとめという形となっている。コンピュータを利用する、ましてやプログラムを作成するといった未知の分野に取り組むことによって自分自身の勉強になり成長にも繋がっていくということをこのゼミで学んだ。

実際、コンピュータを利用するということは現代の生活においては必要不可欠な存在であり、使えて当たり前という時代になっている。現に、最近まで私が直面していた就職活動においてもインターネット、電子メール等を利用するのは勿論のこと、学校の課題を作成にするにしてもワード、エクセルなどの基本的な作業は使えるに越したことはない。

大学 3 年生、大学院 1 年生になってから就職活動を初めようとする際のこれらを学ぶのでは少し遅いと思われる。機械嫌いの人でも、少しでも使えていたほうがこれから、何を初めるにしても有利に進められると思う。

当ゼミでは、これらの作業の一步先を進んだことが学べたので、本当に自分のためになった。私の来年からお世話になる会社からも私が他の人に比べて少しでもこういったパソコンの知識に詳しく、貪欲な人であるということが見込まれたおかげで内定をいただくことが出来た。

実際の当ゼミで学ぶ Java のプログラムなども、勿論使えるに越したことはないだろうが、私はプログラマーではないのでそんな技術はない。少しでも新しいことを学ぼうとする姿勢、なにかに一生懸命打ち込むということ、これらのことが大事なのではないかと私は思う。そういったことがこれから社会人になるにあたっては重要になってくると思われるので、そういった強い意識を持って何事もやるということが大事なのだということがこれらの活動を通じて学ぶことが出来た。

では、実際どういうことをやってきたかということについてまとめることによって、私がどんな研究をしてきたか、それからどのようなことを得たのかを発表していきたいと思う。

## 1.2 方法と意義

この論文では、エクセルによる解析、ワードによる作成ということで、日常誰でも使うパソコンのアプリケーションを利用することによって、誰でもこのような論文を書けるのだということを実証していきたいと思う。その中に私達独自のプログラム作成の結果などを乗せることによって、それがちょっとしたスパイスになっており、基礎から応用への変換という点で注目していただければいいと思う。

大学 3 年生、4 年生の 2 年間という短期間の中でいかに学び、それを身に付けて言ったかということを実証するのがこの論文の意義である。

## 1.3 論文の構成

この卒業論文は、普通の論文と違う 3 部構成になっている。理論、実験、解析という分野を当ゼミ生各々が担当してその 3 つの論文がまとまって 1 つの論文にもなっており、なかなか面白い出来になっていると思う。それぞれの担当は、理論分野を韓国からの留学生である金さん、実験分野を中田君、この解析分野の担当が原田となっている。金さんは金融学科、中田君は会計学科、私原田は経営学科にそれぞれ在籍しており、各人特有のものの方で、書かれているのでそれも面白いと思う。(ゼミ長の石山君は商業・貿易学科に在籍しているので、4 年生全員違う学科に在籍しているという面白い事態になっている)

これらの論文を作成するにあたってのアドバイス、サポートをしてくださっているゼミ長の石山君には本当に頭が上らないほどお世話になっており、彼がいなければこの論文は完成しなかったであろう。本当に心から感謝したい。

### ・この論文(解析編)構成について

この論文は実験編と対になる形をとっており、実験編と同じ 5 章形式になっている。勿論実験編と丸々対になっているわけではないが、オブジェクト指向の基礎、応用など内容が類似している点もあるのでそこら辺にも注目してもらいたいと思う。

解析編の 1 章は、序論となっており、まず私達がどのように学び感じ取ってきたかということを述べている。次に 2 章、この章ではエクセルを利用することによって、先物市場でどのように売り買いが、値段が変化しているかなどを視覚的に捉えていけるようにしている。3 章では Java による解析ツールの実装ということで、オブジェクト指向プログラミングについての応用を述べている。これは、実験編のオブジェクト指向プログラミングの基礎とセットになっているので、その点も注意して読んでいただきたいと思う。次に第 4 章、ここが私の論文のメインとなっているところで、実際に私達が作成したプログラムについて載せている。どのようなプログラムの内容なのか、どの

ような結果が得られるのかなどを詳しく解析している。このプログラムは昨年度大阪市立大学で行われた U-Mart2002 の大会にも実際提出しており、深い思い出になっている。

大学からお金を貰いゼミ合宿を行い、U-Mart 研究会のほうからもお金を頂き、本当に国がお金を出して研究している分野に携われたことに対して強い感銘を受けた。

大阪でもいい思い出を作ることができて非常に満足したゼミ合宿であったことを記憶している。これらのプログラムを作成するのは非常に大変であったがゼミ生一丸となって取り組みなんとかこの困難な作業を乗り越えることが出来た。人間として一回り成長することが出来たなど実感できたワークであった。それで、第 5 章結論ということで、この論文のまとめ、そして 3 部作全体のまとめとして総評をしている。

では、第 2 章のはじまりです。

## 第 2 章 エクセルによる解析

### 2.1 はじめに

この第 2 章では、実際の U-Mart のログデータがどのようになっているか、エクセルを利用し、図で表すことによって可視化している。

視覚的に捉えることによって、U-Mart の中身を見ていこうという狙いである。

U-Mart プロジェクトがどのようなものであるかについては、実験編の方で中田君が書いているのでどちらを参照していただきたいと思う。

### 2.2 U-Mart のログデータの構造

U-Mart のログデータの中身は次のようになっている。

まずは、SpotPrice(現物価格)、FuturePrice(先物価格)を下図で表しており、次に Date(取引日)、BoardNo(板寄せ回数)をその下の図で表している。

先物価格、板寄せなどの説明は金さんが書いている理論編の方で書いているので、そちらを参照していただきたいと思う。

Microsoft Excel - price [読み取り専用]

ファイル(F) 編集(E) 表示(O) 挿入(I) 書式(M) ツール(T) データ(D) ヘルプ(H) 8/27/08

MS P3170

|    | A    | B       | C         | D           | E | F | G | H | I |
|----|------|---------|-----------|-------------|---|---|---|---|---|
| 1  | Date | BoardNo | SpotPrice | FuturePrice |   |   |   |   |   |
| 2  | 0    | 0       | 2849      | 2849        |   |   |   |   |   |
| 3  | 0    | 0       | 2855      | 2855        |   |   |   |   |   |
| 4  | 0    | 0       | 2831      | 2831        |   |   |   |   |   |
| 5  | 0    | 0       | 2826      | 2826        |   |   |   |   |   |
| 6  | 0    | 0       | 2838      | 2838        |   |   |   |   |   |
| 7  | 0    | 0       | 2842      | 2842        |   |   |   |   |   |
| 8  | 0    | 0       | 2849      | 2849        |   |   |   |   |   |
| 9  | 0    | 0       | 2818      | 2818        |   |   |   |   |   |
| 10 | 0    | 0       | 2767      | 2767        |   |   |   |   |   |
| 11 | 0    | 0       | 2784      | 2784        |   |   |   |   |   |
| 12 | 0    | 0       | 2789      | 2789        |   |   |   |   |   |
| 13 | 0    | 0       | 2789      | 2789        |   |   |   |   |   |
| 14 | 0    | 0       | 2813      | 2813        |   |   |   |   |   |
| 15 | 0    | 0       | 2816      | 2816        |   |   |   |   |   |
| 16 | 0    | 0       | 2779      | 2779        |   |   |   |   |   |
| 17 | 0    | 0       | 2789      | 2789        |   |   |   |   |   |
| 18 | 0    | 0       | 2781      | 2781        |   |   |   |   |   |
| 19 | 0    | 0       | 2806      | 2806        |   |   |   |   |   |
| 20 | 0    | 0       | 2788      | 2788        |   |   |   |   |   |
| 21 | 0    | 0       | 2787      | 2787        |   |   |   |   |   |
| 22 | 0    | 0       | 2795      | 2795        |   |   |   |   |   |
| 23 | 0    | 0       | 2791      | 2791        |   |   |   |   |   |

MS P3170/price/

コマンド

現物価格

Microsoft Excel - price [読み取り専用]

ファイル(F) 編集(E) 表示(O) 挿入(I) 書式(M) ツール(T) データ(D) ヘルプ(H) 8/27/08

MS P3170

|    | A    | B       | C         | D           | E | F | G | H | I |
|----|------|---------|-----------|-------------|---|---|---|---|---|
| 1  | Date | BoardNo | SpotPrice | FuturePrice |   |   |   |   |   |
| 2  | 0    | 0       | 2849      | 2849        |   |   |   |   |   |
| 3  | 0    | 0       | 2855      | 2855        |   |   |   |   |   |
| 4  | 0    | 0       | 2831      | 2831        |   |   |   |   |   |
| 5  | 0    | 0       | 2826      | 2826        |   |   |   |   |   |
| 6  | 0    | 0       | 2838      | 2838        |   |   |   |   |   |
| 7  | 0    | 0       | 2842      | 2842        |   |   |   |   |   |
| 8  | 0    | 0       | 2849      | 2849        |   |   |   |   |   |
| 9  | 0    | 0       | 2818      | 2818        |   |   |   |   |   |
| 10 | 0    | 0       | 2767      | 2767        |   |   |   |   |   |
| 11 | 0    | 0       | 2784      | 2784        |   |   |   |   |   |
| 12 | 0    | 0       | 2789      | 2789        |   |   |   |   |   |
| 13 | 0    | 0       | 2789      | 2789        |   |   |   |   |   |
| 14 | 0    | 0       | 2813      | 2813        |   |   |   |   |   |
| 15 | 0    | 0       | 2816      | 2816        |   |   |   |   |   |
| 16 | 0    | 0       | 2779      | 2779        |   |   |   |   |   |
| 17 | 0    | 0       | 2789      | 2789        |   |   |   |   |   |
| 18 | 0    | 0       | 2781      | 2781        |   |   |   |   |   |
| 19 | 0    | 0       | 2806      | 2806        |   |   |   |   |   |
| 20 | 0    | 0       | 2788      | 2788        |   |   |   |   |   |
| 21 | 0    | 0       | 2787      | 2787        |   |   |   |   |   |
| 22 | 0    | 0       | 2795      | 2795        |   |   |   |   |   |
| 23 | 0    | 0       | 2791      | 2791        |   |   |   |   |   |

MS P3170/price/

コマンド

先物価格

|     | Date | BoardNo |      |      |
|-----|------|---------|------|------|
| 157 |      |         | 2379 | 2379 |
| 158 | 0    | 0       | 2338 | 2338 |
| 159 | 0    | 0       | 2325 | 2325 |
| 160 | 0    | 0       | 2291 | 2291 |
| 161 | 1    | 1       | 2220 | 2286 |
| 162 | 1    | 2       | 2122 | 2268 |
| 163 | 1    | 3       | 2172 | 2263 |
| 164 | 1    | 4       | 2247 | 2263 |
| 165 | 1    | 5       | 2306 | 2267 |
| 166 | 1    | 6       | 2245 | 2265 |
| 167 | 1    | 7       | 2310 | 2290 |
| 168 | 1    | 8       | 2312 | 2294 |
| 169 | 2    | 1       | 2272 | 2309 |
| 170 | 2    | 2       | 2225 | 2309 |
| 171 | 2    | 3       | 2161 | 2316 |
| 172 | 2    | 4       | 2127 | 2300 |
| 173 | 2    | 5       | 2166 | 2292 |
| 174 | 2    | 6       | 2228 | 2290 |
| 175 | 2    | 7       | 2192 | 2285 |
| 176 | 2    | 8       | 2217 | 2271 |
| 177 | 3    | 1       | 2201 | 2240 |
| 178 | 3    | 2       | 2181 | 2225 |

取引日  
(わかりやすいように Date と  
入力している)

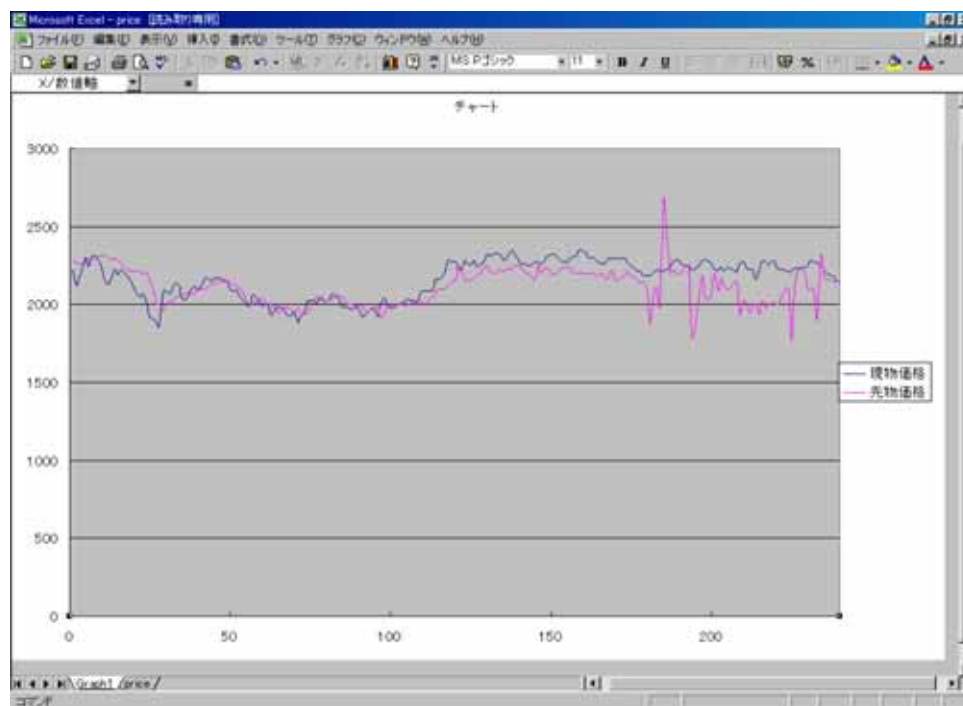
|     | Date | BoardNo |      |      |
|-----|------|---------|------|------|
| 157 |      |         | 2379 | 2379 |
| 158 | 0    | 0       | 2338 | 2338 |
| 159 | 0    | 0       | 2325 | 2325 |
| 160 | 0    | 0       | 2291 | 2291 |
| 161 | 1    | 1       | 2220 | 2286 |
| 162 | 1    | 2       | 2122 | 2268 |
| 163 | 1    | 3       | 2172 | 2263 |
| 164 | 1    | 4       | 2247 | 2263 |
| 165 | 1    | 5       | 2306 | 2267 |
| 166 | 1    | 6       | 2245 | 2265 |
| 167 | 1    | 7       | 2310 | 2290 |
| 168 | 1    | 8       | 2312 | 2294 |
| 169 | 2    | 1       | 2272 | 2309 |
| 170 | 2    | 2       | 2225 | 2309 |
| 171 | 2    | 3       | 2161 | 2316 |
| 172 | 2    | 4       | 2127 | 2300 |
| 173 | 2    | 5       | 2166 | 2292 |
| 174 | 2    | 6       | 2228 | 2290 |
| 175 | 2    | 7       | 2192 | 2285 |
| 176 | 2    | 8       | 2217 | 2271 |
| 177 | 3    | 1       | 2201 | 2240 |
| 178 | 3    | 2       | 2181 | 2225 |

板寄せ回数  
(わかりやすいように BoardNo と  
入力している)

## 2.3 エクセルによる解析

ここでは、ログデータの中身を実際に可視化するように図で表した。

エクセルを利用することによって、価格変動の差を簡単に理解しようとする狙いである。今回はそれに加えて価格変動差のリターンマップも作成した。



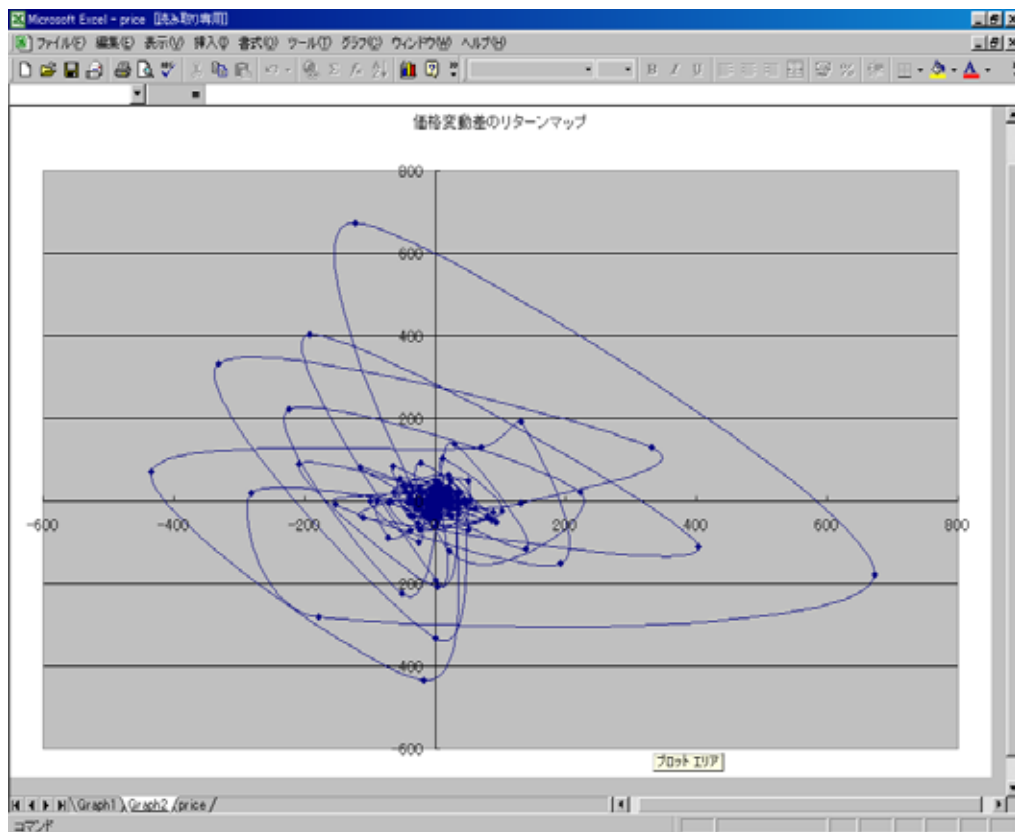
上記のチャートは、1日板寄せ回数8回、30日分合計240回の現物価格、先物価格の価格変動差を図で表したものである。一見ただの数字の羅列に見えるログデータもこのように図で表すことによって、誰でも簡単に理解できるようになっている。

今回の結果は、前半、現物、先物価格の変動差はあまりなかったが、後半板寄せ180回目くらいから変動差が激しくなっているのがわかるであろう。

このように、可視化することによって、一回一回の取引で価格がどのように変動しているかがログを解析することによってすぐ理解できるということが実証された。

次は、価格変動差のリターンマップの図である。





上記の図は、価格変動差のリターンマップを表しているということであるが、普段目にしない図であるかもしれない。この図では実際の現物と先物の価格差を一回ごとに取り、それを図で表したものである。価格変動差があまりない板寄せのほうが圧倒的に多いので、真ん中に点が集約している。一つ上の図で後半に価格が激動しているのがわかるが、その影響が上記の図にも表れている。外回りの線がその影響を受けた証である。

このように、現物、先物の価格変動差の図を取るだけでなく、それをリターンマップとして表すことによって、また新しい物事が見えてくるのでなかなか面白いと思う。

## 2.4 おわりに

この章では、U-Mart の中身であるログデータ、さらにそれを図で表すことによって可視化した。ただのプログラムだけではわからないことも視覚的に見ることによって、新しいことが発見出来るということを理解していただきたいと思い、このような構成にした。

これを見て、少しでも興味が湧いていただいたのならこの狙いは成功したと言えよう。

## 第 3 章 Java による解析ツールの実装

### 3.1 はじめに

ここでは、Java による解析ツールの実装ということで、オブジェクト指向プログラミングの応用とゼミ長の石山君が作成した解析ツールの実装とその利用の仕方について述べている。

オブジェクト指向については、中田君作成の実験編の方にも書いてあるのでそちらも参照していただきたいと思う。

### 3.2 オブジェクト指向プログラミングの応用

こちらでオブジェクト指向という言葉が初めて出てくる。この言葉は普段耳にしない言葉なので、初めて目にする方も少なくないと思う。オブジェクト指向については、実験編でも述べてはいるが復習の意味も込めてこちらでも書いておきたいと思う。石山君の知人である黒田洋介氏のホームページに Object Oriented (オブジェクト指向) について詳しく書かれているので、そちらを参照していただきたいと思う。

#### ・オブジェクト指向とは何か？

システムとはなんであるかという命題に対し、オブジェクトによって類推(アナロジー)したものである、ということになります。これだけでは何の事やら判らないと思いますが。

我々は、世界を原子から成り立っていると言うアナロジーをします。今は、素粒子やヒモの方が人気があるようですが、原子と言うアナロジーによって、世界がどのようなものであるのかを掴んでいくわけです。これと同じようなことを、つまり、そのシステムはどういうものであるかを掴むためにオブジェクト(もの)を単位として類推(アナロジー)していくのです。この作業は、Object Oriented (以下 OO) の中では、分析に割り当てられているかもしれませんが。しかしシステムがなんであるのかを最終的に記述するのはソースコードです。世界をどのように記述していくかで、ソースコードと UML などの抽象度に応じた段階分けがされます。判りやすく言えば、同じ OO でも、曖昧に書いたものと詳細に書いたものと有ります。当然、曖昧から詳細がシステムの開発ですが、現実世界を読み取るならば、その逆を辿ることになるでしょう。この行為そのものが OO によるアナロジーと言えるのです。

最後に、OO は絶対に正しいものでは有り得ません。今は、それが適している事例が多いというだけです。じゃあ、流行り廃りが有るのか、と言われれば「あるでしょう」と答えざるを得ません。しかしながら、学問の一つとして習得すれば、学問はインクリメンタルなものであると言う言葉を

信じれば、次の段階の概念への重要な基礎になるでしょう。

・クラスとは何か？

オブジェクトを抽象化したものです。プログラミングでは型 (Type) ですが、現実でもそのようにとらえてしまっても構わないかもしれません。

オブジェクト(例:人間) → 抽象化 → クラス(例:生き物)

部品が型から大量生産できるように、オブジェクトはクラスから大量生産できます。もちろん一つしか生産しない場合も有りますが、クラスから生まれてきたオブジェクトをインスタンスと呼びます。これは、ソフトウェア開発では、開発の前の段階でこういうオブジェクトが存在すると予想(用語では分析と言います)していたものを、実際にソフトウェアを構成するのはインスタンスと言うことを指しています。

ソフトウェア開発では、オブジェクト(インスタンス)がどういう状態にあるかも重要ですが、クラスがどういう型であるかと言う情報(最終的にはソースコード)は最も重要です。オブジェクト指向とは、クラスをどうやって定義するかが鍵となっています。(それゆえ、クラス指向と呼んでも差し支えない程です。)

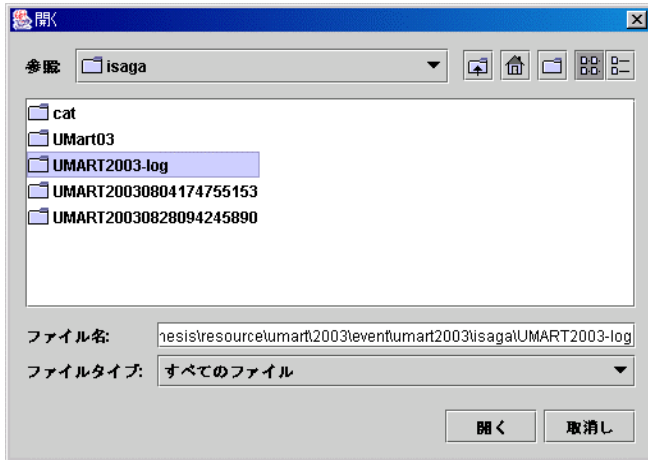
上記の内容は黒田氏のホームページから引用させていただいた文である。これらを読んで少しでも興味を持っていただけたのなら、参考文献に載せた黒田氏のホームページに行って内容を読んでいただきたいと思う。

U-Mart は Java で書かれている。解析ツールを Java で書けば、実験終了直後に自動的にログデータを加工し解析しやすいように予めプログラムすることが出来る。次の 3.3 では、石山君が作成した解析ツールが載せてあるので、そちらを見ていただきたい。ただ、プログラムを組むだけではなく、当たり前のようにこのようなツールを作ってしまう石山君は本当に遠い世界に行ってしまったのだなとしみじみ思ってしまった。

自分はまだまだ未熟者なので少しでも精進していきたいと日々思っている。

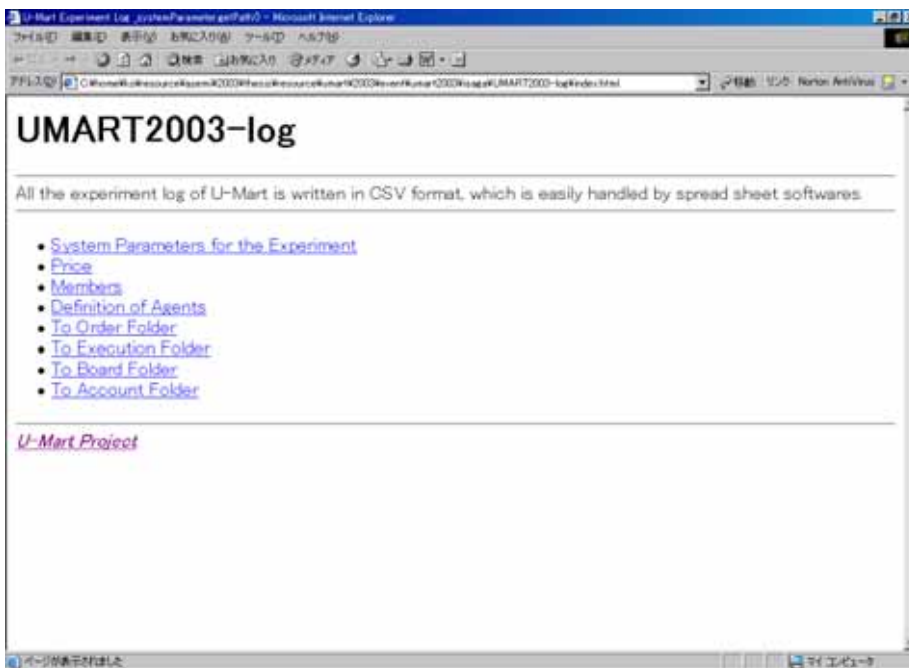
### 3.3 解析ツールの実装と利用

ここでは例として、ログ解析用の HTML ファイルを自動的に作成する Java オブジェクトを実装した。下がそのツールのスクリーンショットとそれを実行した後のログデータのスクリーンショットである。

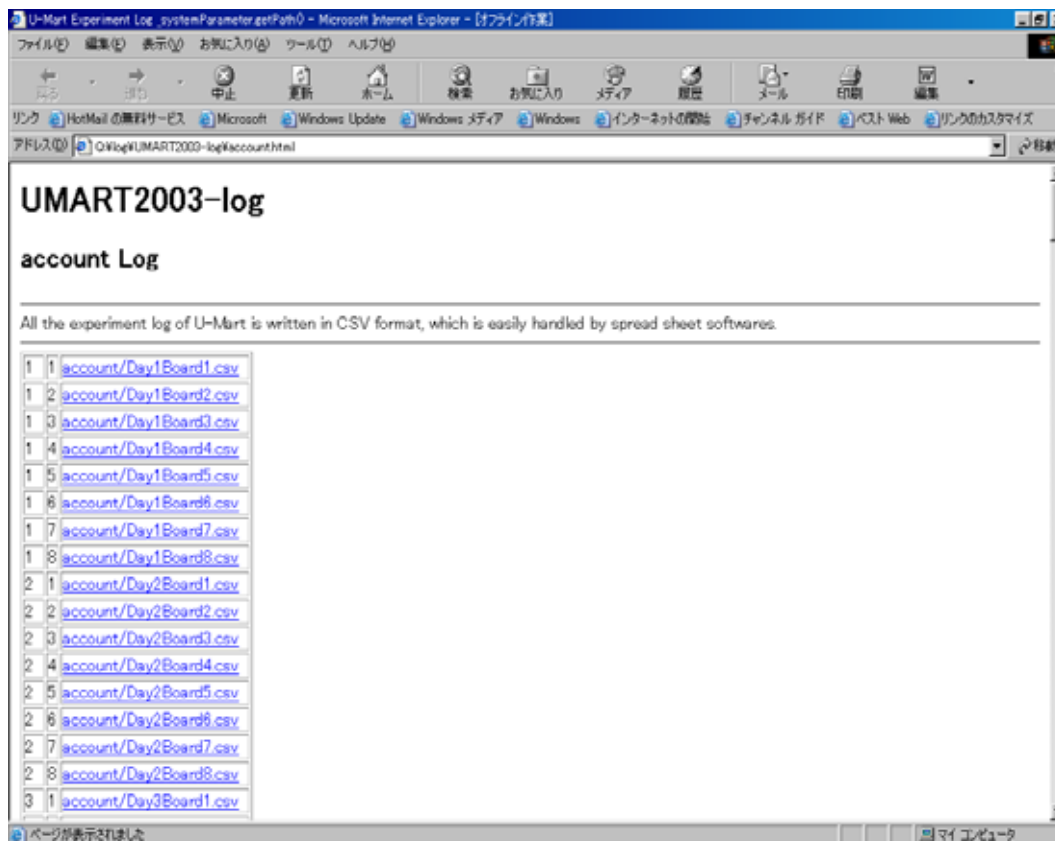


(石山洸著・ULogNavigator)

- ・ Linux のシェルコマンドの Cat などの操作を U-Mart 用に GUI で実現。HTML も自動構成する解析ソフト)



これを実行すると下図のようになる。



これは、account のログであるが、このように 1 日 8 回 30 日間計 240 回分のログデータが抽出された。Board など同じように抽出することが出来る。

### 3.4 おわりに

この章では、Java による解析ツールの実装ということで、オブジェクト志向の応用と、解析ツールの実装とその利用について述べた。

急にオブジェクト指向という聞きなれない単語が出てきたりして、難しい内容になってしまったが、そのあたりは、実験編の方とセットになっているというこの論文独特の構成のためということでご容赦願いたい。黒田氏のホームページには、こういったオブジェクト指向などについて、詳しく説明がされており、本当に参考になった。

次に、解析ツールの方であるが、U-Mart のプログラム自体が Java 言語で書かれているので、Java を使ったツールを利用することですぐに解析できるといった仕様になっている。こちらを作成した石山君も本当に素晴らしい才能を持った人間だと改めて認識した。

次が、私(達)の論文のメインである、エージェントの意思決定と価格変動の考察ということで、実際に U-Mart の大会に提出したプログラムについていろいろ述べてあるので注目していただきたいと思う。

## 第 4 章 エージェントの意思決定と価格変動の考察

### 4.1 はじめに

ここが、私達の論文のメインであるエージェントの意思決定と価格変動の考察である。私(達)の集大成といっても過言ではない。

私達のゼミは、「コンピュータと経済学：市場実験」というテーマで、森嶋通夫先生の「無資源の経済学」を読んで勉強したり、John. H. Holland の「Hidden Order」を読んだりして勉強してきた。

だが、私達のメインはコンピュータである。U-Mart 研究会が開発した仮想先物実験を利用した人工市場研究を行ってきた。まず、私達はスタンドアロン版という個人で出来る実験を行ってきた。だが、私達のゼミは少人数ということもあり、何か新しいことをやったほうがいいのではないかという意見が出た。そこで、私達は自分たちでその戦略を考えそれをプログラム化するという作業をすることになった。しかし、私たちはみな文系の学生ということもあって、プログラムについての知識は皆無であった。そこで先生が、防衛大の佐藤先生や、東工大の小山先生を講師として呼んでくださり、U-Mart プロジェクトについて、プログラムの解説などをしてくださったおかげで導入はしやすかった。

そこからは、個人で勉強することになったが、このような困難な作業も石山君が猛勉強をして先頭に立って指導してくれたり、または中田君のサポートを受けたりすることによって、やっとプログラムを完成することが出来た。元々、2002 年の 11 月に大阪市立大学で行われた U-Mart の大会に出場することが狙いでもあったので、なんとかそれに間に合ったのでよかったと思う。残念ながら私達のチーム(小林さん、中田君、原田)は入賞できなかったが、もうひとつのチーム(石山君、川崎さん、金さん)がチームで 4 位に入賞できたのが、とてもうれしかった。周りの理系の学生や院生の方に引けを取らなかったというのがこれからやっていくにあたっての自信に繋がったからである。

4 年生になってからは就職活動が忙しく、新しいプログラムを作成するには至らなかったが、3 年生の時点でこのようなものが作れたのは非常によい機会であったと思う。このようにプログラムという作業をすることによって、自分自身で作ったものが形となって残るということは素晴らしいことだと思うし、大変貴重な経験になった。

ただ、一般のゼミのように本を読んでレジュメを作り、それについて議論するといった内容では、このような充実感は得られなかったであろう。就職活動をする際にでも、他の大学や学部の学生ではこのような経験をしてきた人はいないし、これらのことについて話すと面接官の方の反応も上々であったので、よかったと思う。

このプログラムには私のいろいろな思い出が込められているのである。

## 4.2 エージェントの意思決定の考察

ここでは、実際に私達が作成したプログラムとその解説について述べてある。

2002年のU-Martの大会では、1人1体計6体のエージェントを提出したが、今回はその中でも3人(石山君、中田君、原田・3年次)で作成したものを載せたいと思う。

```
/* COPYRIGHT 2002 (C) Sho Nakada, Kei Harada, Ko Ishiyama All Right Reserved.
 * 2002 U-Mart Project
 */
package strategy;

import java.util.*;

/**
 * A agent decides its buying or selling subject to stochastic strategy.
 *
 * @author Sho Nakada, Kei Harada, Ko Ishiyama
 */
public class TestStrategy extends Strategy{
    private Random random;
    private final int WIDTH_OF_PRICE = 100;
    private final int MAX_QUANT = 50;
    private final int MIN_QUANT = 30;
    private final int MAX_POSITION = 300;
    private final double EDGE = 0.3;
    private final int referenceTerm = 10;

    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 latestSpotPrice = spotPrices[119];
int latestFuturePrice = futurePrices[59];
if ( (latestSpotPrice <= 0) || (latestFuturePrice < 0) ) {
    order.buysell = Order.NONE;
return order;
}

int K1HighestPrice = 0;
int K1LowestPrice = 99999;
for (int i = 0; i < 24; i++) {
    if (futurePrices[i+35] > 0) {
        K1HighestPrice = Math.max(K1HighestPrice,
futurePrices[i+35]);
        K1LowestPrice = Math.min(K1LowestPrice,
futurePrices[i+35]);
    }
    //System.out.println("K1" + K1HighestPrice + " K1" +
K1LowestPrice);
}

int K2HighestPrice = 0;
int K2LowestPrice = 99999;
for (int i = 0; i < 24; i++) {
    if (futurePrices[i+36] > 0) {
        K2HighestPrice = Math.max(K2HighestPrice,
futurePrices[i+36]);
        K2LowestPrice = Math.min(K2LowestPrice,
futurePrices[i+36]);
    }
    //System.out.println("K2" + K2HighestPrice + " K2" +
K2LowestPrice);
}

int D1HighestPrice = 0;
int D1LowestPrice = 99999;

```



```

        for (int i = 0; i < 40; i++) {
            if (futurePrices[i+19] > 0) {
                D1HighestPrice = Math.max(D1HighestPrice,
futurePrices[i+19]);
                D1LowestPrice = Math.min(D1LowestPrice,
futurePrices[i+19]);
            }
            //System.out.println("D1 " + D1HighestPrice + " " D1 " +
D1LowestPrice);
        }

        int D2HighestPrice = 0;
        int D2LowestPrice = 99999;
        for (int i = 0; i < 40; i++) {
            if (futurePrices[i+20] > 0) {
                D2HighestPrice = Math.max(D2HighestPrice,
futurePrices[i+20]);
                D2LowestPrice = Math.min(D2LowestPrice,
futurePrices[i+20]);
            }
            //System.out.println("D2 " + D2HighestPrice + " " D2L " +
D2LowestPrice);
        }

        double K1 = (double)(latestFuturePrice - K2LowestPrice) /
(double)(K2HighestPrice - K2LowestPrice);
        double K2 = (double)(latestFuturePrice - K2LowestPrice) /
(double)(K2HighestPrice - K2LowestPrice);
        double D1 = (double)(latestFuturePrice - D1LowestPrice) /
(double)(D1HighestPrice - D1LowestPrice);
        double D2 = (double)(latestFuturePrice - D2LowestPrice) /
(double)(D2HighestPrice - D2LowestPrice);
        //System.out.println("K1 " + K1);
        //System.out.println("K2 " + K2);
        //System.out.println("D1 " + D1);
        //System.out.println("D2 " + D2);

```

```

if ( (K1 > D1) || (K2 < D2) || (1-EDGE <= K2) || (1-EDGE <= D2) ) {
    order.buysell = Order.SELL;
}else if ( (K1 < D1) || (K2 > D2) || (EDGE >= K2) || (EDGE >= D2) ) {
    order.buysell = Order.BUY;
}else {
    order.buysell = Order.NONE;
}

if ( futurePrices[59] > 0 && futurePrices[58] > 0 && futurePrices[57] > 0 ) {
if ( futurePrices[59] > futurePrices[58] && futurePrices[58] >
futurePrices[57] ) {
    order.buysell = Order.BUY;
} else if ( futurePrices[59] < futurePrices[58] && futurePrices[58] <
futurePrices[57] ) {
    order.buysell = Order.SELL;
} else if ( futurePrices[59] > futurePrices[58] && futurePrices[58] <
futurePrices[57] ) {
    order.buysell = Order.BUY;
} else if ( futurePrices[59] < futurePrices[58] && futurePrices[58] >
futurePrices[57] ) {
    order.buysell = Order.SELL;
} else {
    order.buysell = Order.NONE;
}
} else {
order.buysell = random.nextInt(2) + 1;
}

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

```

```

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

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

    order.quant = MIN_QUANT;
    return order;
}
}

```

以上が提出したエージェントである。

以下解説文に続く。

#### U-MART2002 エージェント解説

原田 圭

中央大学 商学部 有賀裕二研究室

#### ・ ストカスティクス理論(改)

オシレーター系の指標のひとつにストカスティクスというものがある。このプログラムはストカスティクス理論を元にし、それをこの理論の弱点であるトレンドへの対策を組み合わせたプログラムとなっている。

ストカスティクスでは%K、%D、SD と呼ばれる3種類のラインが用いられる。

$$\text{ストキャスティクス}\%K = (C - L_n) \div (H_n - L_n) \times 100$$

C = 当日終値

L<sub>n</sub> = 過去n日間の最安値

H<sub>n</sub> = 過去n日間の最高値

ストキャスティクス%K は、一定期間(n日)の最高値から最安値の間で、当日終値からその最安値を引いた幅が、何%を占めるのかを百分率(0% ~ 100%)の指数で表したものである。0%に近いほど売られ過ぎ、100%に近いほど買われ過ぎということになる。

一般的に期間は5日と設定、ストキャスティクスの 70%以上を買われ過ぎ、30%以下は売られ過ぎとし、30%以下で買い・70%以上を売るのが一般的。

$$\text{ストキャスティクス}\%D = (C - L_n \text{の} m \text{日間の合計}) \div (H_n - L_n \text{の} m \text{日間の合計}) \times 100$$

C = 当日終値

L<sub>n</sub> = 過去n日間の最安値

H<sub>n</sub> = 過去n日間の最高値

ストキャスティクス%D は、ストキャスティクス%K をm日ゆるやかにしたものである。

一般的な期間の設定は、「n日 = 5、m日 = 3」で、ストキャスティクスの 70%以上を買われ過ぎ、30%以下は売られ過ぎとし、30%以下で買い・70%以上を売るのが一般的。

$$\text{ストキャスティクス} SD = \text{キャスティクス}\%D \text{の} j \text{日移動平均}$$

一般的な期間の設定は、「n日 = 5、m日 = 3、j日 = 3」で、ストキャスティクスの 70%以上を買われ過ぎ、30%以下は売られ過ぎとし、30%以下で買い・70%以上を売るのが一般的。

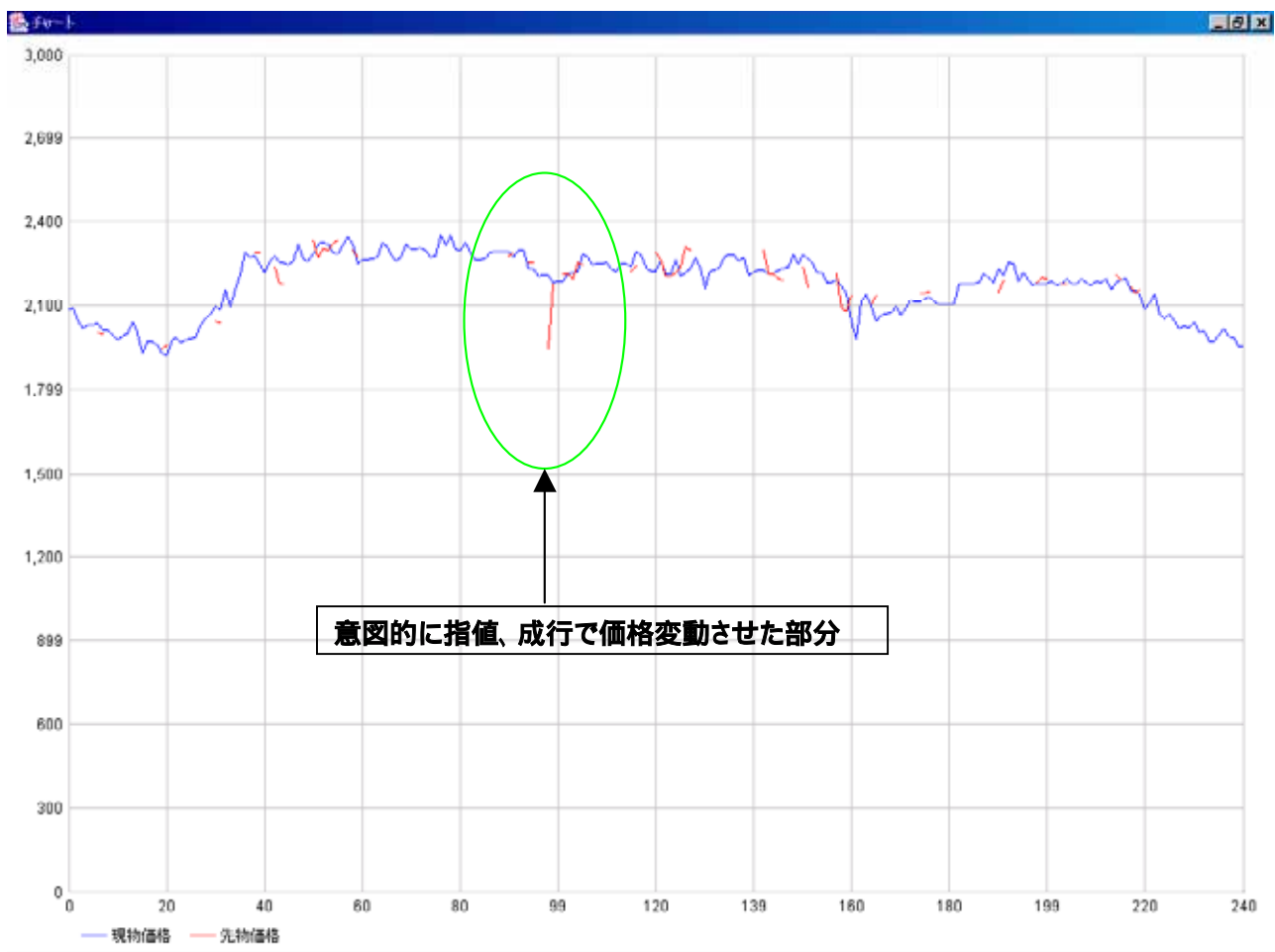
今回は売買の判断に%K、%D の二本のラインを用いた「ファースト・ストキャスティクス」を採用した。「ファースト・ストキャスティクス」も基本的には他のストキャスティクス同様、0%に近いほど売られ過ぎ、100%に近いほど買われ過ぎを示している。

売買の基準は2本のラインのクロスで、%K ラインが%D ラインを下から上に抜けてきた時に買い、上から下に抜けてきた時に売りとする。しかし、何度もクロスが発生するので、ダマシが多くなる。このため、「30%以下の位置で上に抜けたら買い」、「70%以上の位置で下に抜けたら売り」というフィルターをかけている。

上記にプラスして、直前の値段が上がったら買い、下がったら売るというプログラムを入れることによってトレンドへの対策としている。

#### 4.3 価格変動の考察

ここでは、価格変動の考察をして行こうと思う。今回利用したエージェントは、ランダムエージェントと実際私が作成したストカスティクスエージェントである。初日からは現物、先物と大きな変動はなくゆったりと動いてきた。そこで、90 回目あたりの取引で、ヒューマンエージェントとして指値、成行注文を出し、意図的に価格操作をした。その結果が下図に表れている。一気に価格変動が起こったのである。このようにただエージェントを実装するのではなく、ヒューマンとして参加することで、影響を与えることが出来ることがわかった。



(ランダムエージェント、ストカスティクスエージェントを利用した実験結果)

#### 4.4 おわりに

この章では、実際に私達が U-Mart2002 の大会に参加した時のエージェントとその解説、そしてそのエージェントを利用した実験結果(価格変動)の考察を行ってきた。

自分自身で作成したエージェントに加え、自分もヒューマンエージェントとして参加できるので、そういう楽しみが出来るのも面白いと思う。

このエージェントを作成する作業というのは、とても大変なことであったが、みんなが協力し合い乗り越えることが出来た。この作業を通じてさらにゼミ員の結束が強くなったと思う。

実際、エージェントを作成したり、それを実験したりといったことは一見難しいかもしれないが、何事も前向きに取り組むという姿勢があれば乗り越えていけると思う。

私は、これらの作業をしている時にそれを痛感した。これらを通じて得たものは大きかったと思う。

## 第 5 章 結論

この論文は一見、一般的に言える論文ではないかもしれない。しかし、私達は 2 年間の演習の総決算として、この論文を書いている。ただ、本を読み、資料、ホームページなどを参考にして作成する論文とは一線を画している。序論でも書いたようにこの論文は 3 部構成になっている。その一つ一つが勿論論文になっており、総合的に見ても論文という新しい構成になっている。これが、私達が積み上げてきた歴史なのであり、そして当ゼミでの最後の挑戦なのである。

この論文はまず、ログデータの解析から始まり、オブジェクト指向の応用、そしてメインであるエージェントの考察と繋がっている。どれもワードや、エクセルなど身近に出来るソフトだけで解説してあるのでわかりやすくなっていると思う。エージェントに関しても、実際にプログラムを組むという作業は多少困難ではあるが、内容を理解するという点では、あまりプログラムの知識は重要ではない。何が書かれているかと知ろうとする姿勢が大事なのだと思う。

実際、論文を書いているときに感じたが、何事も前向きにしようという姿勢が重要だということを改めて実感した。私が体験した就職活動もそうだし、普段日常にしている勉強についてもそれは言えることだと思った。すぐに結果を求めるのではなく、常にプラス思考で、一生懸命頑張ることが大切なのだという事を伝えたいと思う。

この論文も最初は困難を極めたが、皆で頑張ろうという気持ちがあったからこそ乗り越えることが出来たのだと思う。

その結果、理論から始まりそして実験、解析へと繋がるほかに類を見ない素晴らしい論文が出来たのだと思う。理論、実験、解析としっかり見ていただければこれほど幸せなことはない。

## 謝辞

この論文を作成するに当たっては私一人では、到底出来なかったであろう。このような変わった論文を作成しようとした際に一緒に協力してくれた中田君、金さんは勿論、この論文の方向性を見出させてくれ、労を惜しまないほど手伝ってくれた石山君には頭が上がらない思いである。

そして、このゼミに入ることによってたくさんの新しい発見をすることができた。その2年間の指導教授でもある有賀教授にも感謝したいと思う。先生のおかげで、自分のフィールドが多少なりとも広がったと思う。

2年間という短期間で、自分の力不足を痛感した。しかし、それとともに得たものも大きかったと思う。まだまだ、未熟者であるがこれから社会人になるにあたって気合を入れて何事も取り組んで行きたいと思うし、それを通じて成長していきたいと思っている。

次期ゼミ生にも、目標を持って頑張ってもらえれば幸いである。

## 参考文献

### 関連ホームページ

1. 黒田洋介氏のホームページURL <http://www13.big.or.jp/~akasata/>

### プログラミングコード

1. 石山洸、ULogNavigator (Linux のシェルコマンドの Cat などの操作を U-Mart 用に GUI で実現。HTML も自動構成する解析ソフト), November 2002.
2. 中田翔、石山洸、原田圭、Stochastic Strategy<sup>1・2</sup>, October 2002.

### 引用文献

1. 中田翔、U-Mart2002 エージェント解説,2002
2. 原田圭、U-Mart2002 エージェント解説,2002