U-Mart Toolkit: A General-Purpose Artificial Market Development Toolkit Library

Isao Ono[1], Naoki Mori[2], and Yoshihiro Nakajima[3]

[1] Tokyo Institute of Technology, 4259 Nagatsuda-cho, Midori-ku, Yokohama, Kanagawa, Japan.
isao@c.titech.ac.jp
[2] Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku, Sakai, Osaka, Japan.
mori@cs.osakafu-u.ac.jp
[3] Osaka City University, 3-3-138 Sugimoto Sumiyoshi-ku, Osaka, Osaka, Japan.
yoshi@econ.osaka-cu.ac.jp

Abstract. The U-Mart project has been working on artificial market research based on agent-based modeling since 1999 in order to design institutions that can be applied directly to the real world. As part of its activities, we developed artificial futures market simulators called the Itayose-based U-Mart System (the U-Mart system Ver.2) and the Zaraba-based U-Mart System (the U-Mart system Ver.4). In this paper, we introduce the U-Mart toolkit which is currently being developed. The U-Mart toolkit is a general-purpose artificial market development toolkit library to facilitate developing various artificial market simulators. The U-Mart toolkit not only makes it possible to independently use useful modules in the U-Mart systems developed so far but also makes it easy to execute parameter survey using only machine agents.

Keywords: Artificial markets · U-Mart · Development toolkit library.

1 Introduction

In the financial market, transaction volumes are increasing year by year. Therefore, the degree to which financial markets affect the economy is increasing. Therefore, analysis on market stability and influence of institutional design has become important. Since the empirical research in real markets is impossible in many cases, the artificial market approach attracts much attention [1–4].

Since 1999, the U-Mart project [5] has been conducting artificial market research based on agent-based modeling in order to design institutions that can be applied directly to the real world. In the U-Mart project, we have designed and implemented artificial futures markets called the U-Mart system [8, 7, 6]. Since the beginning, we have aimed to be able to conduct experiments in which humans (human agents) and transaction algorithms (machine agents) coexist.

The U-Mart system has been highly valued not only as a research tool for market systems and trading algorithms but also as an educational tool in the fields of economics and engineering. However, the existing U-Mart system has

two major problems. The first problem is that it is not supposed to recycle some modules in the U-Mart system, such as the Iteyose-based pricing module or the Zaraba-based pricing module. The second problem is that it is cumbersome to execute parameter survey using only machine agents where random number series, combination of agents, other institution parameters, etc. are changed.

In this paper, we introduce the U-Mart toolkit which is currently being developed to remedy the two problems of the current U-Mart system. The U-Mart toolkit is a general-purpose artificial market development toolkit library to facilitate developing various artificial market simulators. The U-Mart toolkit not only makes it possible to independently use useful modules in the U-Mart systems developed so far but also makes it easy to execute parameter survey using only machine agents.

Section 2 introduces the U-Mart system developed by the U-Mart project so far and discusses the problems of the current U-Mart system. In Section 3, in order to remedy the problems of the current U-Mart system, we introduce the U-Mart toolkit which is currently being developed. In Section 4, we summarize this paper and discuss future work.

2 U-Mart systems

2.1 History

Since the first version was released in 2000, the U-Mart system has been improved through open experiments and used as teaching materials for university education. In 2002, we compiled the knowledge learned so far and redeveloped a new system with full scratch. This system is called the U-Mart system Ver. 2 (the Itayose-based U-Mart system) [8, 7, 6]. The Itayose-based U-Mart system is a highly complete system that can seamlessly perform network experiments, experiments in a stand-alone environment, real-time experiments including human agents, and acceleration experiments with machine agents. In 2011, the U-Mart system Ver. 4 (the Zaraba-based U-Mart system) [6] adopting the Zaraba-based pricing system used in many current markets was proposed. Using the Zaraba-based U-Mart system, it was also possible to study various stock markets that could not be handled with the Iteyose-based U-Mart system. Currently, the latest U-Mart system is the Zaraba-based U-Mart system.

2.2 Itayose-based U-Mart System (U-Mart System Ver.2)

The Itayose-based U-Mart system is an order-driven artificial market that adopts the batch auction method, which is called “Itayose” in Japanese. This system is a model of a single exchange managing a single futures market of a single brand. This system deals with a single brand. Agents trade a virtual futures index of an existing spot index which is traded outside of the system such as Nikkei 225 and S & P 500. The futures prices in the virtual market emerge as the results of interactions among trader agents. Human agents as well as machine agents can participate in the market via the network at the same time.

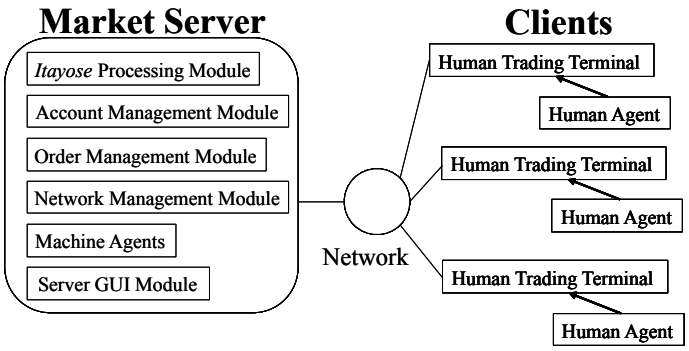

Fig. 1. The configuration diagram of the Itayose-based U-Mart system

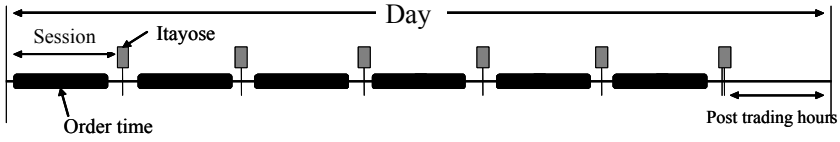

Fig. 2. An example of the schedule of the Itayose-based U-Mart system

Fig. 1 shows the configuration diagram of the system. The system is designed as a client-server model. It consists of the market server and the human agent trading terminal. The market server is modeled on a stock exchange in the real world. It is responsible for order management, account management, contract process, and so on. The market server comprises multiple modules as shown in Fig. 1.

The time is represented by day and session in the Itayose-based U-Mart system. A day consists of several sessions and post-trading hours as shown in Fig. 2. In Fig. 2, the schedule has six sessions per day. During trading hours, the exchange of the Itayose U-Mart system accepts orders from traders. At the end of every session, a contract price is determined by the Itayose method. Market-to-market is done daily in post-trading hours. Settlement is done using the spot price at the due date.

The left figure in Fig. 3 shows the GUI (Graphical User Interface) of the market server. The human agent trading terminal is a GUI program that is used by a human in order to participate in trade over the network. The GUI program is designed to provide an intuitive and easy-to-use environment. The right figure in Fig. 3 shows the GUI of the human agent trading terminal.

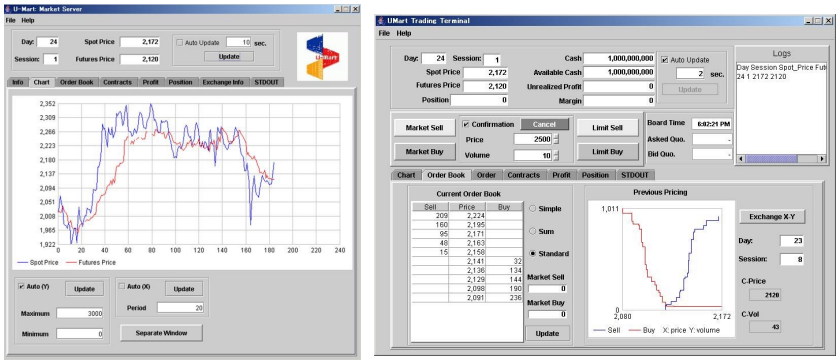

Fig. 3. Screen shots of the Itayose-based U-Mart system

2.3 Zaraba-based U-Mart System (U-Mart System Ver.4)

This section introduces the Zaraba-based U-Mart system (U-Mart Ver.4). The Zaraba-based U-Mart system has the following features comapred to the Itayose-based U-Mart system.

- The Zaraba-based U-Mart system supports the continuous double auction method, which is called "Zaraba" in Japanese in addition to the Itayose method as pricing methods. The Zaraba method is adopted in almost all the money markets in the world including the Tokyo Stock Exchange (TSE).
- The Zaraba-based U-Mart system supports spot and/or futures maket(s) of multiple brands. Agents can trade not only futures but also spot of multiple brands in the Zaraba-based U-Mart system. This enables users to investigate correlation of spot and futures prices of multiple brands.
- Various institutions can be easily investigated in the Zaraba-baesd U-Mart system. Users not only can use predefined institutions such as tick size and quote parameters which are adopted in TSE but also can define original institutions to investigate their effect. Tick size is an increment by which prices move. Quote parameters define the range of price fluctuation. The next execution price must be within some range around the most recently executed price.

Fig. 4 shows the configuration diagram of the Zaraba-based U-Mart system. The Zaraba-based U-Mart system consists of the market server and the human agent trading terminal as the Itayose-based U-Mart system does. The market server comprises multiple modules as shown in Fig. 4. Each module in the Zaraba-based U-Mart system has been modified significantly from the Itayose-based U-Mart system because the time management system is different from that of the Itayose-based U-Mart system.

The time is represented by day, session and unit time (ut) in the Zaraba-based U-Mart system. One ut is the minimum unit of time. Fig. 5 is an example

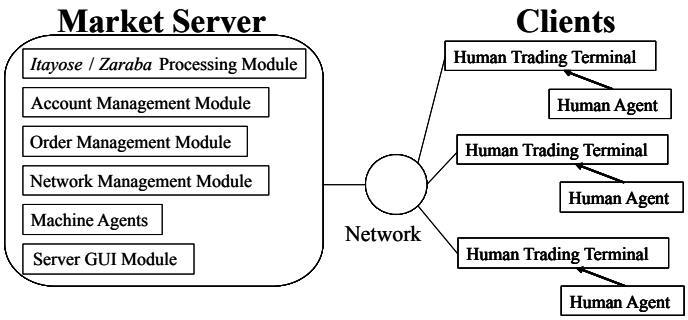

Fig. 4. The configuration diagram of the Zaraba-based U-Mart system

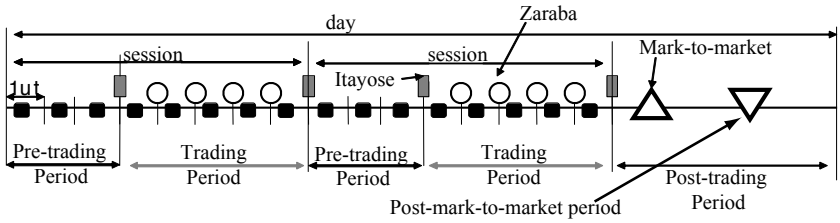

Fig. 5. An example of the schedule of the Zaraba-based U-Mart system

of the schedule of the Zaraba-based U-Mart system. Opening and closing prices are determined by the Itayose method and other prices by the Zaraba method as done in TSE.

The left figure in Fig. 6 shows the GUI of the market server of the Zaraba-based U-Mart system. The Zaraba-based U-Mart system provides an easy-to-use trading terminal for human agents as shown in the right figure in Fig. 6. The trading terminal provides users rich information such as technical indicators. This enables users to make decisions easily and quickly.

3 U-Mart Toolkit

The Zaraba-based U-Mart system has two major problems. The first problem is that it is not supposed to recycle some modules in the U-Mart system, such as the Iteyose-based pricing module or the Zaraba-based pricing module. This is because the Zaraba-based U-Mart system is designed as a framework, which means that each module in the U-Mart depends on each other and it is difficult to use it separately as a library. In particular, since almost all the modules rely on a complex time model where day, session and ut are used, it is difficult to use the modules of the Zaraba-based U-Mart system when markets adopting other time models are implemented. The second problem is that it is cumbersome

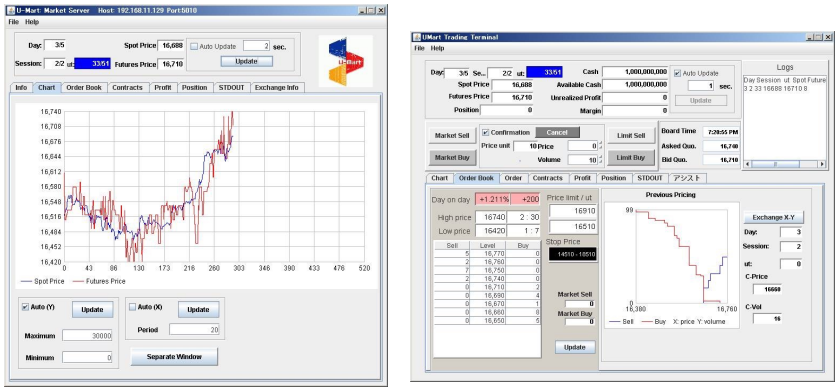

Fig. 6. Screen shots of the Zaraba-based U-Mart system

to execute parameter survey using only machine agents where random number series, combination of agents, other institution parameters, etc. are changed.

The U-Mart toolkit is a general-purpose artificial market development toolkit that we are currently developing to remedy the problems of the above-mentioned existing U-Mart system. The U-Mart toolkit is a general-purpose artificial market development toolkit library to facilitate developing various artificial market simulators. The U-Mart toolkit allows users to easily use the useful modules of the existing U-Mart system independently as a library. In addition, the U-Mart toolkit employs a simple time model that uses only time steps. Furthermore, the U-Mart toolkit provides a module that makes it easy to execute parameter survey using only machine agents. The U-Mart toolkit is an object-oriented Java-based library. The following describes the modules of the U-Mart toolkit.

3.1 Order management module

The order management module consists of an order class named “UOrder” and an order manager class named “UOrderMager”.

The order class maintains information such as its action type indicating ‘new’, ‘repay’, ‘cancel’ or ‘change’, its trading type indicating ‘buy’, ‘sell’ or ‘none”, its order type indicating ‘limit’ or ‘market’, its price, its volume, a user name making this order, a time step when this order was submitted, its ID and the target ID of the order agreed with this order.

The order manager class provides methods for agents to make orders, to change orders and to cancel orders. The order manager class manages all the orders made by agents.

3.2 Itayose-based pricing module

The Itayose-based pricing module matches orders form agents and determines a price by the Itayose method. The Itayose method is a pricing method in which all

buy and sell orders are compared and a price is determined so that the number of executed orders is the maximum as shown in Fig. 7. The priority of the orders is as follows:

1. Order-type priority
   Market orders have priority over limit orders. A limit order is an order to buy/sell at no more/less than a specific price. A market order is an order to be executed at the current market price.
2. Price priority
   A sell/buy limit order at a lower/higher price has a higher priority. If there are many orders indicating the same price, the rule of time priority is applied.
3. Time priority
   If two orders indicate the same price, the older one has a higher priority than the newer one. If these two orders are placed in the same session, the priority is determined randomly.

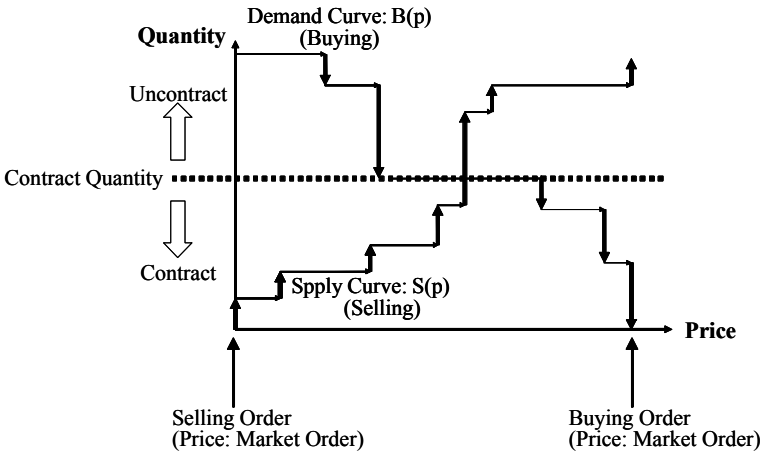

Fig. 7. The Itayose method

The Itayose-based pricing module allows markets to check market instructions such as circuit breakers and price limitations twice per transaction as shown in Fig. 8. As shown in Fig. 8, the step of the temporary contract enables markets to check market institutions such as price limitations before contract. This allows users to simulate various markets in the real world.

The Itayose-based pricing module mainly consists of a processed order information class named “UProcessedOrderInformation” and an Itayose-based pricing class named “UCallAuction”.

The processed order information class maintains its ID, the IDs of two matched orders, a time step when it was processed, its price, its volume and its brand name.

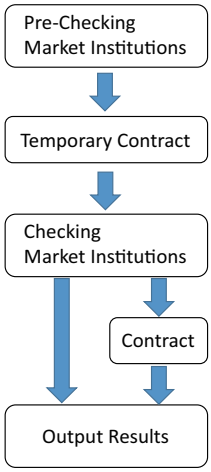

Fig. 8. The transaction process with the Itayose-based pricing module

The Itatyose-based pricing class provides methods to execute temporary contract and contract. The temporary contract method takes an order manager instance and returns a processed order information instance including a price and a transaction volume. The contract method takes the order manager instance and the processed order information instance and returns a list of processed order information instances.

3.3 Zaraba-based pricing module

The Zaraba-based pricing module determines a price and a volume by the Zaraba method. The Zaraba method is a pricing method in which a new order is matched with existing orders, and a price is determined by auction-like process as shown in Fig. 9. The priority of the orders is the same as that in the Itayose-based pricing module.

The Zaraba-based pricing module allows markets to check market instructions such as circuit breakers and price limitations twice per transaction as the Itayose-based pricing module does. The step of the temporary contract enables markets to check market institutions such as price limitations before contract. This allows users to simulate various markets in the real world.

The Zaraba-based pricing module mainly consists of the processed order information class shared with the Itayose-based pricing module and a Zaraba-based pricing class named “UContinuousSession”.

The Zaraba-based pricing class provides methods to execute temporary contract and contract. The temporary contract method takes a new order instance and an order manager instance and returns a processed order information instance including a price and a transaction volume. The contract method takes

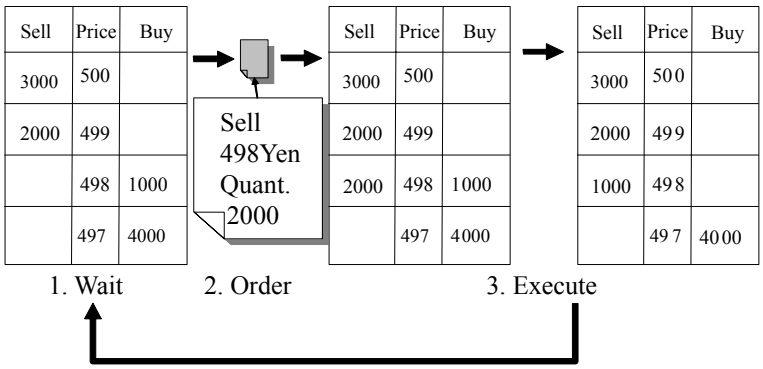

Fig. 9. The Zaraba method

the order manager instance and the processed order information instance and returns a list of processed order information instances.

3.4 Account management module

The account management module manages a position and profit and loss for a brand of an agent. Currently, it supports only futures markets. Other types of markets will be supported in near future.

The account management module has an account management class named "UProfitAndPositionManagerForFutures". This class provides methods to register a contracted order, to do mark-to-market at the end of the day and to do settlement.

3.5 Logging and report module

The logging and report module provides functions to facilitate data logging and making reports. This module mainly consists of the following four classes.

- TCCsvData class provides methods to write/read data to a file in the CSV format and to apply various operations to the data.
- TCTeXTable class takes a TCCsvData instance and converts it to a table in TeX format.
- TCTeXGnuplot class takes a TCCsvData instance and converts it to multiple graphs in TeX formant using the gnuplot.
- TCScreenGnuplot class takes a TCCsvData instance and displays it as multiple graphs on screen by using the gnuplot.

3.6 Parameter survey module

The parameter survey module allows users to execute parameter survey in parallel and seamlessly not only on PCs but also on PC clusters or supercomputers

where the SSH service and a job management service such as TORQUE [9], SLURM [10] and Univa Grid Engine [11] work and Java is installed. The parameter survey module works without database service. In parameter survey, users can set the values of primitive-type variables and the instances of class-type variables with external configuration files, which is realized by the dependency injection (DI) mechanism [12]. The parameter survey module also summarizes the results of parameter survey generated by the logging module in a single CSV file.

A user needs to execute the following steps for parameter survey on a PC.

1. Create directories and configuration files for parameter survey. This is done by writing a short Java program using TCJavaExperimentDesigner class.
2. Run a program named "TCExecutor" with a command line argument of the directory where the directories and configuration files were made in Step 1. TCExecutor performs parameter survey according to the configuration files made in Step 1.
3. Run a program named "TCSummarizer" to summarize the results.

A user can execute parameter survey on a remote PC cluster or a remote supercomputer by doing almost the same operations as those required on a PC. The differences are executing a program named "TCRemoteParallelExecutor" instead of TCExecutor in Step 2 and downloading results from the remote PC cluster or the remote supercomputer by executing a program named "TCResultDownloader" in Step 3.

1. Create directories and configuration files for parameter survey. This is done by writing a short Java program using TCJavaExperimentDesigner class.
2. Run TCRemoteParallelExecutor with a command line argument of the directory where the directories and configuration files were made in Step 1. TCRemoteParallelExecutor uploads all the directories and files necessary for parameter survey to the remote PC cluster or the remote supercomputer automatically and submits jobs necessary for parameter survey on the remote PC cluster or the remote supercomputer according to the configuration files made in Step 1.
3. Download all the directories and configuration files and resulting files from the remote PC cluster or the remote supercomputer by using a program named "TCResultDownloader" after the parameter survey is finished. A user can check whether all the experiments on the remote PC cluster or the remote supercomputer are finished or not by using a program named "TCStatusChecker".
4. Run a program named "TCSummarizer" to summarize the results.

4 Conclusion

In this paper, we introduced "U-Mart toolkit" which is currently being developed to facilitate not only developing various artificial market simulators and but also

doing parameter survey seamlessly on a PC, a PC cluster and a supercomputer. The U-Mart toolkit is a general-purpose artificial market development toolkit library that makes it possible to independently use useful modules in the U-Mart systems developed so far. The U-Mart toolkit provides the Itayose-based pricing module, the Zaraba-based pricing module, the account management module, the logging and report module and the parameter survey module.

In future work, we would like to extend the functions of each module to handle more various markets and institutions. Also, we have a plan to reimplement the Zaraba-based U-Mart system to enhance its generality.

References

1. Arthur, W. B., Holland, J., LeBaron, B., Palmer, R. and Taylor, P.: Asset pricing under endogenous expectations in an artificial stock market, in Arthur, W. B., Durlauf, S., and Lane, D., eds., ‘ The Economy as an Evolving Complex System II ’ , Addison-Wesley, pp. 15-44 (1997).
2. Raberto, M., Cincottia, S., Focardib, S., and Marchesic, M. : Agent-based simulation of a financial market, Physica A, Statistical Mechanics and its Applications, Vol. 299, Issues 1-2, pp. 319-327 (2001).
3. Chen, S-H., Corresponding and Yeh, C-H. : Evolving traders and the business school with genetic programming: A new architecture of the agent-based artificial stock market ” , Journal of Economic Dynamics and Control Vol. 25, Issues 3-4, pp. 363-393 (2001).
4. Izumi, K., and Ueda, K. : Phase Transition in a Foreign Exchange Market: Analysis Based on an Artificial Market Approach, IEEE Transactions on Evolutionary Computation, Vol.5, No. 5, pp 456-470 (2001).
5. http://www.umart.org/html/
6. Kita, H., Taniguchi, K. and Nakajima, Y. (Eds.): Realistic Simulation of Financial Markets, Springer (2016).
7. Ono, I., Sato, H., Mori, N., Nakajima, Y., Matsui, H., Koyama, Y. and Kita, H.: U-Mart System: A Market Simulator for Analyzing and Designing Institutions, Evolutionary and Institutional Economics Review, 5-1, 63/79 (2008).
8. Ono, I., Mori, N., Sato, H., Kita, H., Matsui, H. and Nakajima, Y. : U-Mart System Version 2: A Multi-Purpose Artificial Market Simulator”, Proc. the 4th International Workshop on Agent-based Approaches in Economic and Social Complex Systems , in CD-ROM (2004).
9. http://www.adaptivecomputing.com/products/torque/
10. https://slurm.schedmd.com/
11. http://www.univa.com/products/
12. https://en.wikipedia.org/wiki/Dependency_injection