Chapter 7
Agent-Based Simulation for Service and Social Systems and Large-Scale Social Simulation Framework

Hideyuki Mizuta

Abstract Because of the dynamic and heterogeneous interactions among human beings with their bounded rationality, a service system discussed in Service Science, Management, and Engineering (SSME) is recognized as a complex adaptive system to which quantitative scientific analysis is difficult to apply. In this chapter, we introduce a computational approach for such complex adaptive systems called agent-based simulation. Since the 1990s, agent-based simulation has gained significance as a tool to reproduce complex stock market interactions by modeling human traders as software agents. Recently, agent-based social simulations are utilized to support the decision-making of city planners for various real social issues. For this purpose, we have developed a large-scale social simulation framework "X10-based Agent Simulation on Distributed Infrastructure (XASDI)." In this chapter, we will introduce our earlier work with a small number of agents and then describe the large-scale social simulation framework and its applications.

Keywords Agent-based Simulation · Social simulation · XASDI framework · Service system · SSME

7.1 Introduction

Service systems are complex adaptive systems and also social systems involving human beings. Service systems cannot be represented with convenient sets of mathematical equations mainly because of the complexities and vagaries of human behavior. Therefore, it is quite difficult to evaluate and understand service systems quantitatively with scientific methods. In traditional economic theories, some simplifications such as static equilibrium state, representative agent, and full rationality have been introduced to construct comprehensive and elegant theories

H. Mizuta (✉)
IBM Japan, Chuo-ku, Tokyo, Japan
e-mail: e28193@jp.ibm.com

Y. Aruka (ed.), *Digital Designs for Money, Markets, and Social Dilemmas*,
Evolutionary Economics and Social Complexity Science 28,
https://doi.org/10.1007/978-981-19-0937-5_7

with feasible solutions, but in the late twentieth century, these assumptions have received increasingly strong criticism that they cannot capture the modern economic crises and dynamic economics of the Internet era. Various approaches from multiple disciplines try to tackle these complex emerging problems (Arthur et al. 1997).

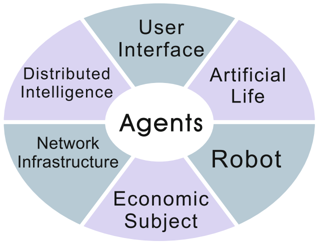
For example, researchers from the fields of physics and computer science are analyzing enormous amounts of data from real-world markets, using statistics to find hidden patterns and new economic principles. This approach is called "EconoPhysics" (Mantegna and Stanley 1999), and it seeks to explore similarities between critical phenomena from physics and market behaviors from economics, such as the now well-known concepts of the long-tail and power-law scaling, phenomena that can be found in many natural and artificial systems.

To understand such systems, we believe agent-based approaches can offer a new set of powerful tools (Mizuta 2016). Since the main method of an agent-based approach is an agent-based simulation (ABS) on computers, researchers using such approaches can model a target service system and its stakeholders with an agent-based model (ABM) and simulate the dynamic behaviors and interactions of the service system with microscopic agents emulated in computers (Namatame et al. 2002).

The word "agent" is used with a variety of meanings for different purposes, such as UI (user interface) agents, email reminder agents, physical agents, or software robots acting as agents to serve people (see Fig. 7.1). In the agent-based approach, each "agent" represents a dynamically interacting economic (or social) entity, thus abstracting the stakeholders from the real world and representing their characteristics of heterogeneity and bounded rationality that traditional economics and social science have difficulty in understanding with models. By using computational agent-based simulations (often written with object-oriented programming languages), such heterogeneous and dynamic interacting agents with bounded rationality can be intuitively implemented as objects with their interactions based on message passing.

Various software technologies such as OOP (Object-Oriented Programming), AI (Artificial Intelligence), and network communications have relationships with ABS. In addition to knowledge of computer software, researchers also need domain knowledge about objective systems such as economics, social science, financial engineering, complex adaptive systems, or behavioral economics. Hence, ABS is a multi-disciplinary science similar to Service Science, Management, and

Fig. 7.1 Various meanings of the word "Agent"



Engineering (SSME), which is the study of service systems and value cocreation phenomena (Spohrer and Maglio 2010).

Agent-based simulation evolved relatively recently as many researchers became able to conveniently use powerful PCs and popular programming languages in the last few decades. In the early period, agents were often designed to reproduce complex adaptive systems and financial markets (Izumi 1998). Then, researchers applied ABS to social behaviors such as norm emergence, game theory situations, learning, and organization.

Currently, agent-based social simulation has begun to be used for more practical and concrete problems such as traffic systems, pedestrian flows, and business operations. To evaluate the complex behavior of interacting enormous entities in a large city, such a social simulation needs to manage millions of agents with various behavior models and preferences. Moreover, the computation power is also required to analyze an enormous combination of possible situations and strategies with repeated simulations (reviews and roadmap are introduced in Noda et al. 2015). Therefore, the distributed parallel execution of large-scale agent-based social simulations on various platform (e.g., post-petascale supercomputers or cloud) is very important.

For this purpose, we have developed a large-scale distributed simulation framework "X10-based Agent Simulation on Distributed Infrastructure (XASDI)" (XASDI 2017) and its applications with the support of JST, CREST. XASDI is available as open-source software at GitHub with tutorial, API documents, sample programs, and a docker file (https://github.com/x10-lang/xasdi). XASDI is a large-scale agent-based social simulation framework with an enormous number (millions) of agents to represent Citizens in cities or countries. It enables highly parallel and distributed simulation with X10 platform (X10 2017) for post-petascale supercomputers.

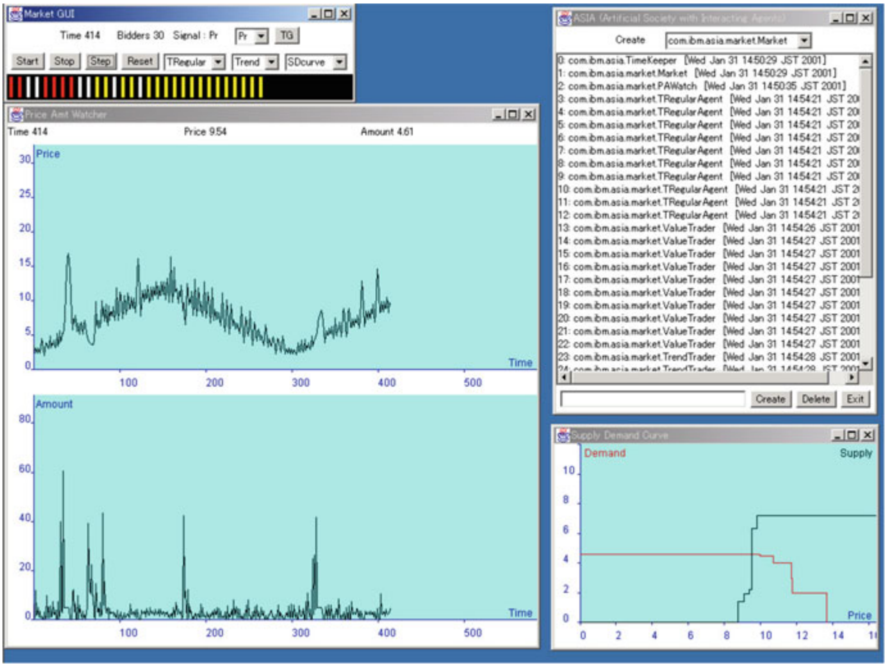
XASDI software stack contains a core runtime written in X10 language for distributed agents and execution management and Java API bridge to enable application programmer to utilize Java language. By utilizing the XASDI framework, researchers can develop various social simulation applications such as traffic, pedestrian, and evacuation simulations that support decision-making of city planners.

In the balance of this chapter, we introduce our earlier work with a small number of agents and then describe the large-scale social simulation framework and its applications based on the previous papers (Mizuta 2016; Mizuta and Imamichi 2018).

7.2 Market and Auction Simulation

Agent-based simulations (ABSs) reproduce a service system by modeling humans as heterogeneous agents. We introduce several examples from our ABS projects. At first, we applied ABS to understand phenomena in the market system. The stability of prices in asset markets is clearly a central issue in economics. From a system's point of view, markets inevitably entail the feedback of information in the form of price signals, and like all feedback systems may exhibit unstable behavior. Steiglitz and Shapiro (1998) created the price oscillation and bubbles in a simple commodity market model with producer/consumer agents and two types of speculators. In Steiglitz's market model, three types of agents (Regular agent, Value trader, and Trend trader) trade food and gold. Regular agents produce food or gold depending on the price (exchange rate between food and gold) and consume food. Value traders and trend traders are speculators with different strategies.

We considered the stability in this model with various price signals and found that the inversely weighted average of bid price stabilizes the market dramatically (Mizuta et al. 2003). Figure 7.2 shows a screenshot of the simulation where price bubbles appear with heterogeneous agents. The largest window shows two graphs showing the market clearing price and the trade volume. With the simplest market only with regular agents, the market price shows strong oscillation due to inventory cycle (Fig. 7.3). We showed that this price oscillation with producer/consumer agents is stabilized by introducing different price signals (Fig. 7.4). On the basis of the simulation, we also gave analytical results on the simplified dynamical system with different signals.



Time 414 Bidders 30 Signal : Pr

Time 414 Price 9.54 Amount 4.61

Create com.ibm.asia.market.Market

```
0: com.ibm.asia.TimeKeeper [Wed Jan 31 1450:29 JST 2001]
1: com.ibm.asia.market.Market [Wed Jan 31 1450:29 JST 2001]
2: com.ibm.asia.market.PAWatch [Wed Jan 31 1450:35 JST 2001]
3: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 20
4: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 20
5: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 20
6: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 20
7: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 20
8: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 20
9: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 20
10: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 2
11: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 2
12: com.ibm.asia.market.TRegularAgent [Wed Jan 31 1454:21 JST 2
13: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:26 JST 2001
14: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
15: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
16: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
17: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
18: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
19: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
20: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
21: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
22: com.ibm.asia.market.ValueTrader [Wed Jan 31 1454:27 JST 2001
23: com.ibm.asia.market.TrendTrader [Wed Jan 31 1454:28 JST 2001
24: com.ibm.asia.market.TrendTrader [Wed Jan 31 1454:28 JST 2001
```

Fig. 7.2 Screenshot of the market simulation

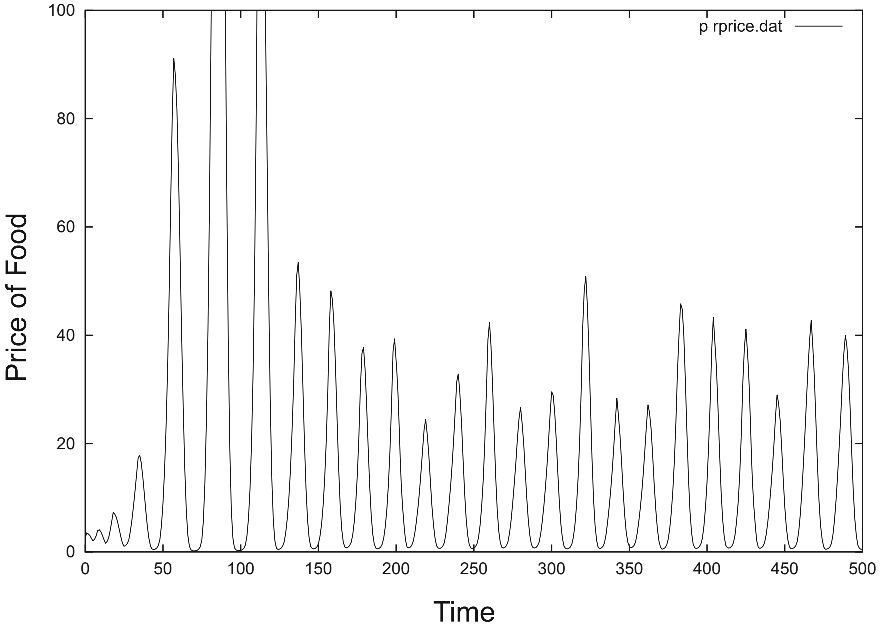


Fig. 7.3 Price vs. trading period with regular agents only

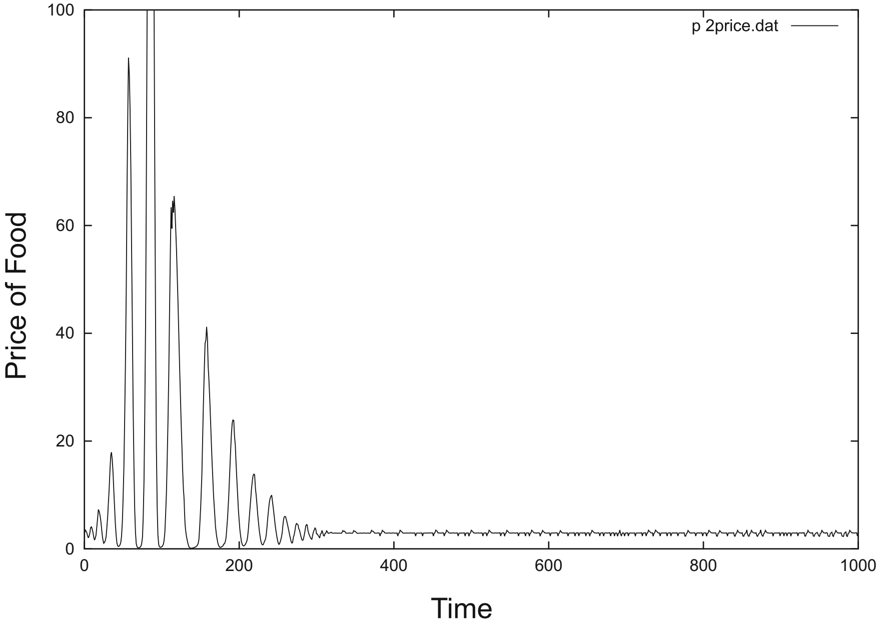


Fig. 7.4 Price vs. period with inversely weighted average bit prices as the signal

Next, we consider a simpler form of market, that is, auctions.

The use of online auctions has grown rapidly since the 90s, and in general, many segments of the economy are becoming granulated at a finer and finer scale. Thus, understanding behavior of auctions, and especially the interaction between the design of auctions, agent behavior, and the resulting allocations of goods and money, has become increasingly important—first not only because we may want to design auctions that are as profitable as possible from the sellers' point of view but also because we may want to bid in auctions or design computer systems that respond well to the loads that auctions generate.

To investigate such dynamic interactions between heterogeneous bidders and also the price formulation through successive auctions, we developed an agent-based simulation of dynamic online auctions (Mizuta and Steiglitz 2000).

In a dynamic auction simulation, we investigated the behavior on popular online auctions with heterogeneous bidder types, e.g., Early Bidders and Snipers. The model considers a single auction involving the sale of one item by one seller to one of many bidders, who submit their bids over time in the interval $[0, T)$ to an auctioneer, who awards the item to the highest bidder at closing time. A bidder can submit more than one bid during the auction. In experiments, the starting bid price is fixed at 1, and the duration of the auction is $T = 500$ time units.

At the beginning of each auction, each bidder determines his first valuation of the item.

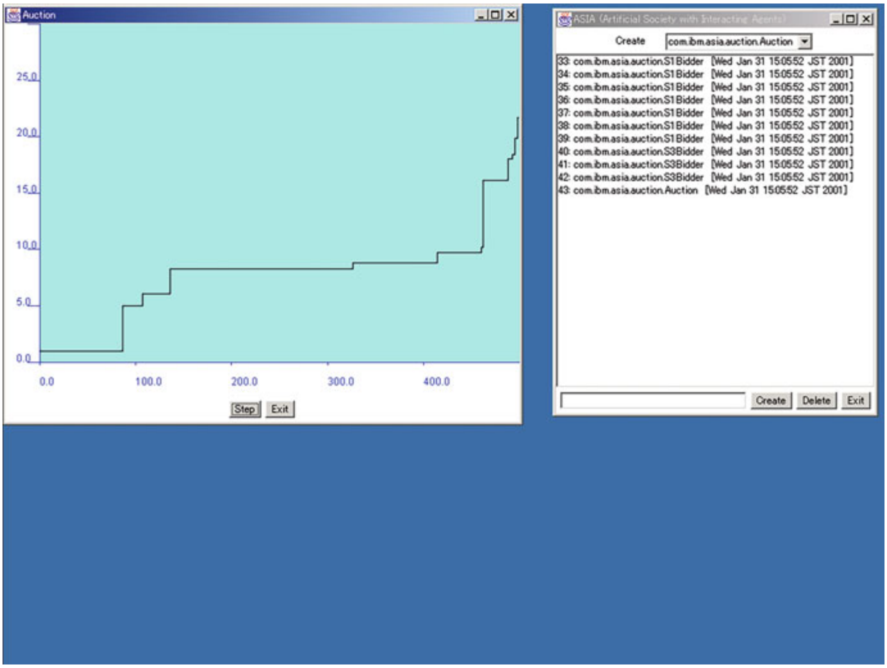
At each time period $0 < t < T$, each bidder receives the status of the auction, can update his estimation on a fixed schedule or probabilistic, and can submit bids if the conditions for his strategy are satisfied. In this model, early bidders can bid any time during the auction period, update their valuations continuously, and compete strongly with each other, and snipers wait until the last moments to bid.

An example auction simulated is shown in Fig. 7.5. The graph for the second highest bid price shows price jumps at the last few moments. We also indicate histograms of winning prices by Early Bidders and Snipers in Figs. 7.6 and 7.7. In most cases, Snipers win the item with broadly distributed prices. However, there are small chances for Early Bidders win the item with very low prices or very high prices.

7.3 International Emissions Trading Simulation and Gaming

In a series of works (Mizuta and Yamagata 2001a,b, 2002), we considered agent-based simulations in a computer and gaming experiments with human players for the international CO2 emission trading.

We have been investigating the CO2 emission trading under the Kyoto Protocol. Nation agents correspond to participating countries or regional groups and COP agent is a central auctioneer and manages the international trading. In this model, we created 12 nations' agents; 6 are Annex I countries that are developed countries and assigned reduction targets in the level of emission in 1990, and 6 are Non-Annex



```
33: com.ibm.asia.auction.S1Bidder   [Wed Jan 31 15:05:52 JST 2001]
34: com.ibm.asia.auction.S1Bidder   [Wed Jan 31 15:05:52 JST 2001]
35: com.ibm.asia.auction.S1Bidder   [Wed Jan 31 15:05:52 JST 2001]
36: com.ibm.asia.auction.S1Bidder   [Wed Jan 31 15:05:52 JST 2001]
37: com.ibm.asia.auction.S1Bidder   [Wed Jan 31 15:05:52 JST 2001]
38: com.ibm.asia.auction.S1Bidder   [Wed Jan 31 15:05:52 JST 2001]
39: com.ibm.asia.auction.S1Bidder   [Wed Jan 31 15:05:52 JST 2001]
40: com.ibm.asia.auction.S3Bidder   [Wed Jan 31 15:05:52 JST 2001]
41: com.ibm.asia.auction.S3Bidder   [Wed Jan 31 15:05:52 JST 2001]
42: com.ibm.asia.auction.S3Bidder   [Wed Jan 31 15:05:52 JST 2001]
43: com.ibm.asia.auction.Auction   [Wed Jan 31 15:05:52 JST 2001]
```

Fig. 7.5 Screenshot of auction simulation

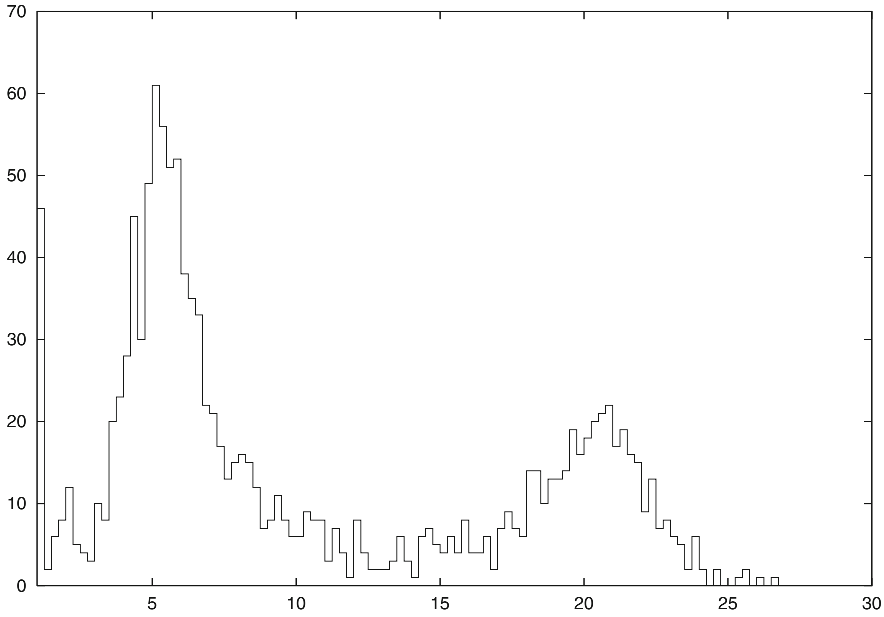


Fig. 7.6 Histogram of winning prices by early bidders

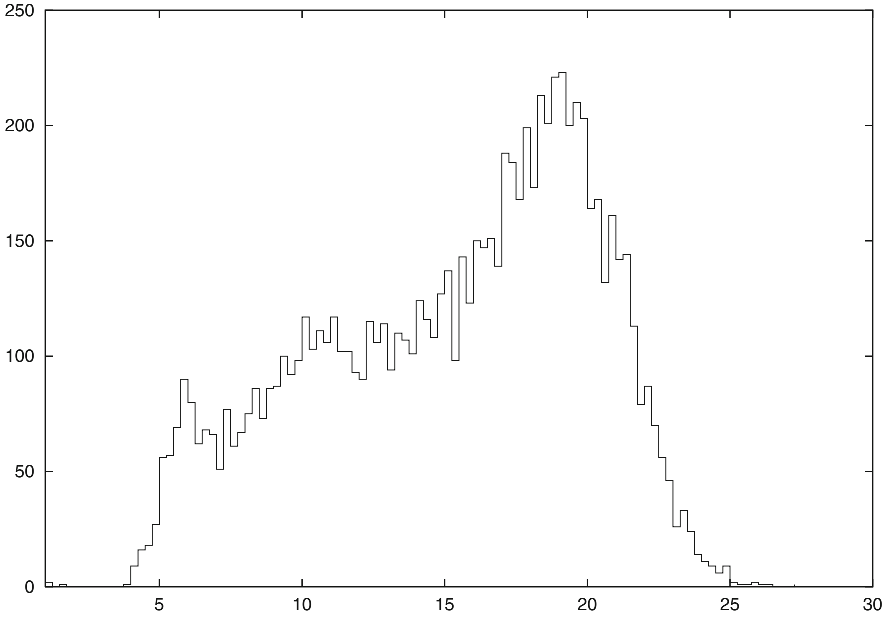


Fig. 7.7 Histogram of winning prices by snipers

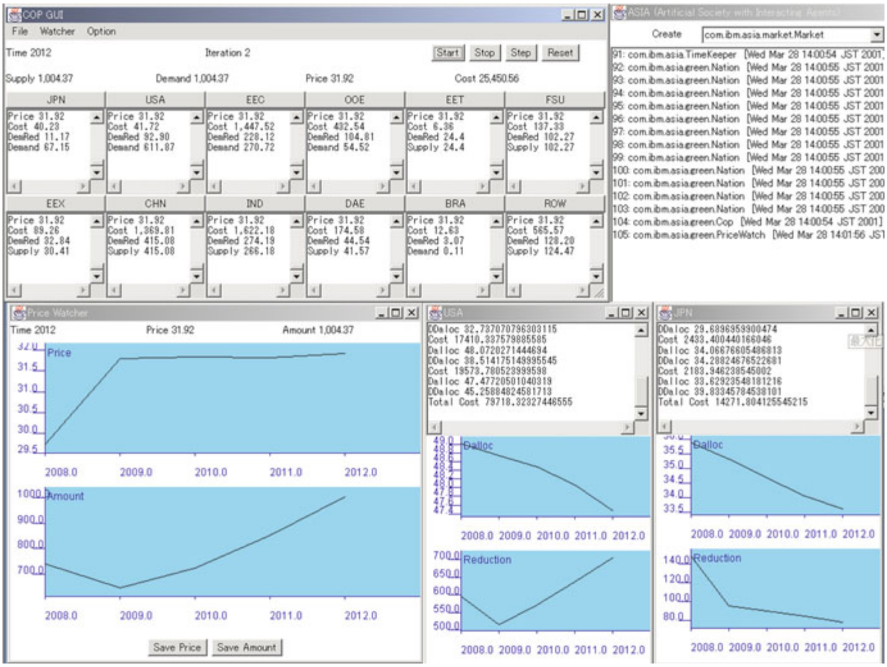
I countries that are developing countries and not assigned targets for reduction as in the CERT model (Grütter 2000).

We consider dynamic market development through the first commitment period 2008–2012. In each trading year, the COP agent sends Request for Bid (RFB) messages to all nations which have an asking price. Upon receiving the RFB message, a nation agent examines the asking price and his Marginal Abatement Cost (MAC) to decide the amount of the domestic reduction. Then, he sends back a bid message to the COP agent which says how much he wants to buy or sell at the asked price. After repeating this RFB-BID process, the COP model will find the clearing price where the demand and the supply balance and send the trade message to approve the trades for the year. Thus, the equilibrium price for each year is determined when the MAC functions and the assigned reductions of all of the participants are given.

For the multiple trading periods, we considered a partition of the assigned reduction as a strategy of agents. The dynamics of MAC is given by considering the available technologies for reduction.

As a simple dynamic process for the reduction technology, we adopt reusability and deflation. Once the technology whose cost is lower than the certain price is used, the reusability of the technology will be restricted. On the other hand, the technical innovations and deflation decrease the cost of the technology.

Figure 7.8 shows an example of the simulation views. We can see brief reports on 12 agents and price changes from 2008 to 2012.



Time 2012 Iteration 2

Supply 1,004.37 Demand 1,004.37 Price 31.92 Cost 25,450.56

JPN	USA	EEC	OOE	EET	FSU
Price 31.92 Cost 40.23 DemRed 11.17 Demand 67.15	Price 31.92 Cost 41.72 DemRed 92.90 Demand 611.87	Price 31.92 Cost 1,447.52 DemRed 228.12 Demand 270.72	Price 31.92 Cost 432.54 DemRed 104.81 Demand 54.52	Price 31.92 Cost 6.36 DemRed 24.4 Supply 24.4	Price 31.92 Cost 137.33 DemRed 102.27 Supply 102.27

EEX	CHN	IND	DAE	BRA	ROW
Price 31.92 Cost 89.26 DemRed 32.84 Supply 30.41	Price 31.92 Cost 1,369.81 DemRed 415.08 Supply 415.08	Price 31.92 Cost 1,622.18 DemRed 274.19 Supply 266.18	Price 31.92 Cost 174.58 DemRed 44.54 Supply 41.57	Price 31.92 Cost 12.63 DemRed 3.07 Demand 0.11	Price 31.92 Cost 565.57 DemRed 128.20 Supply 124.47

com.ibm.asia.market.Market

```
91: com.ibm.asia.TimeKeeper  [Wed Mar 28 14:00:54 JST 2001,
92: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 2001
93: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 2001
94: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 2001
95: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 2001
96: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 2001
97: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 2001
98: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 2001
99: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 2001
100: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 200
101: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 200
102: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 200
103: com.ibm.asia.green.Nation  [Wed Mar 28 14:00:55 JST 200
104: com.ibm.asia.green.Cop  [Wed Mar 28 14:00:54 JST 2001]
105: com.ibm.asia.green.PriceWatch  [Wed Mar 28 14:01:56 JST
```

Time 2012 Price 31.92 Amount 1,004.37

```
DDaloc 32.737070796303115
Cost 17410.337579885585
Dalloc 48.0720271444694
DDaloc 38.514175149995545
Cost 19573.780523999598
Dalloc 47.47720501040319
DDaloc 45.25884824581713
Total Cost 79718.32327446555
```

```
DDaloc 29.6896959900474
Cost 2433.400440166046
Dalloc 34.06676605486813
DDaloc 34.28824676522681
Cost 2183.946238545002
Dalloc 33.62923548181216
DDaloc 39.83345784538101
Total Cost 14271.804125545215
```

Fig. 7.8 Screenshot of international emissions trading simulation

Gaming simulations with human players in an environment similar to the agents' environment are expected to help us in constructing plausible behavior models and extracting the essential dynamics. We have developed a web application for gaming so that most client PCs with web browsers can easily access it.

In this gaming simulation, players determine the amounts of the domestic reduction of CO2 and the amounts of the excess demand for international emission trading according to the presented price in the RFB at each iteration. Information such as the cost graph, the MAC, the total reduction target, and the trading history is also given.

We show the results of the gaming experiment (Mizuta and Yamagata 2005).

In the experiments, we tried two types of trading model: Walras equilibrium price and double auction (DA). For trading among computational agents, we used Walras trading for one or five trading years to find the equilibrium price and cost-effective strategies. On the other hand, we introduced DA trading for human players since the iterative process of Walras trading is too troublesome for human traders to use and will not converge with dishonest and irrational bids. In an experiment with real human bidders, we tried the Walras trading with students in a preliminary gaming experiment, which did not reach equilibrium. With DA trading, gaming players enjoyed free trading, and sell/buy permits to achieve the target positively.

Table 7.1 Reference and results of gaming experiments

Reference					
Nation	JPN	EEC	OOE	EET	FSU
Cost	47,307	130,852	24,308	−25,351	−84,997
Gaming 04-1					
Nation	JPN	EEC	OOE	EET	FSU
Cost	7783	265,821	12,600	−10,106	−41,655
Perf	84%	−103%	48%	−60%	−51%
Gaming 04-2					
Nation	JPN	EEC	OOE	EET	FSU
Cost	112,143	387,817	30,361	−58,142	−128,823
Perf	−137%	−196%	−25%	129%	52%

The most characteristic behavior emerged in the game was price control by sellers. Sellers (EET: Economies in Transition of Eastern Europe and FSU: Former Soviet Union) were unwilling to sell until the market price became very high, and buyers (JPN, EEC: 15 EU members, OOE: Rest OECD) were forced to pay more than the equilibrium. Even after we changed the assigned countries of the players, this tendency of high price controlled by sellers was sustained and sellers obtained greater revenue than the equilibrium trading of computational agents.

Players and the game controller accessed the online gaming system with their web browsers. The game controller predefined the game's nation parameters and controlled the procession of each game. Each game consists of 5 trading years and one trading year takes about 10 minutes of real time. This web-based gaming system collaborated with the agent-based simulation framework. Hence, we can investigate the behavior of trading using the computational agents with the same factors given to game players.

Two samples of gaming results are shown in Table 7.1. This gaming experiment was held at the University of Tokyo with 10 undergraduate students.

In the experiments (Gaming 04-1 and Gaming 04-2), five countries/areas (JPN, EEC, OOE, EET, and FSU) are assigned to players.

We consider the relative performance of students by comparing their total cost (M$) to achieve the Kyoto targets with the results of the agent-based simulation with Walras equilibrium price.

In Gaming 04-1, JPN showed excellent performance. Investigating the recorded activities in this game, we can see how JPN players achieved such a performance. From the log data, we found that JPN made a large trade with FSU at the early stage. Therefore, JPN need not trade emissions after price rises rapidly and got the high performance. But, FSU and EET studied from the previous game and obtained huge profit by hesitating to trade until the later stage in Gaming 4-2.

Through these gaming experiments, students who did not have previous knowledge quickly studied during the short-term gaming experiments and behaved more effectively. Thus, such a gaming simulation also seems efficient for teaching interaction strategies in complex social systems.

7.4 Agent-Based Simulation Framework XASDI

In this section, we introduce the X10-based Agent Simulation on Distributed Infrastructure (XASDI 2017). This framework is published as an open-source software under the Eclipse Public License (EPL).

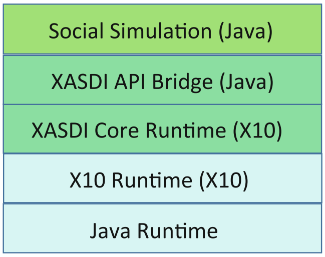
XASDI is a large-scale agent-based social simulation framework with an enormous number (millions) of agents to represent Citizens in cities or countries. XASDI enables distributed simulation with the X10 language for post-petascale supercomputers. The X10 programming language (X10 2017) is the APGAS (Asynchronous, Partitioned Global Address Space) language that provides highly parallel and distributed functionalities with Java-like syntax. On the other hand, XASDI provides easy-to-use API with Java that is familiar to application programmer of social simulations and can be developed with powerful IDE functionalities (e.g., Eclipse refactoring and debugger).

XASDI software stack contains a core runtime written in X10 language for distributed agents and execution management and Java API bridge to enable application programmer to utilize Java language (Fig. 7.9). By utilizing XASDI framework, users can easily develop their social simulator with Java on distributed parallel environment without studying X10 language.

The agent in XASDI is referred to as Citizen and Citizen has corresponding CitizenProxy managed in the simulation environment to exchange messages. To manage CitizenProxy, XASDI provides a hierarchical container structure called Place, Region, and World (see Fig. 7.10). CitizenProxies belong to a Place and Places belong to a Region. World can contain several Regions, but usually there is only one Region in the World.

Here, we note the confusing terminology of X10 programming language and this framework. X10 also uses the term “Place”; however, the meaning of the term is different compared with the term of XASDI. The Place of X10 is used to denote the distributed execution environment for multicore or multi-node. To distinguish “Place” of X10 and XASDI, we will use “X10 Place” for Place of X10 and “Place” for Place of XASDI. Only one World instance exists in one X10 Place and manages lists of entities in the World including Region and Citizen. The World can also contain IDs of Citizens in other X10 Places.

Fig. 7.9 XASDI software stack



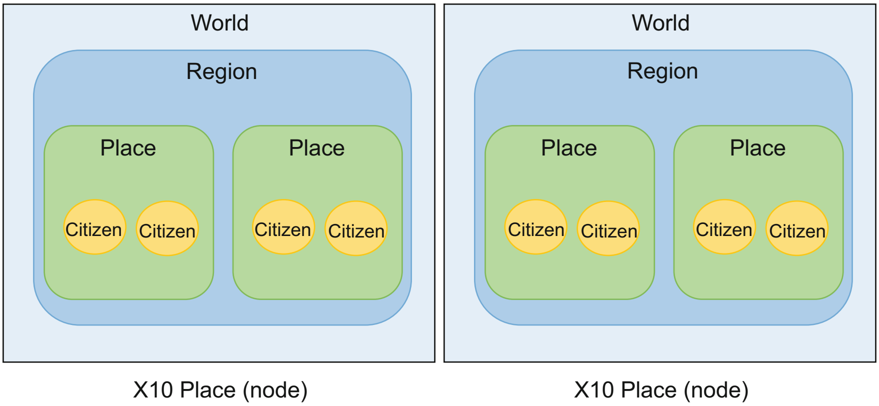


Fig. 7.10 XASDI hierarchical structure to manage agents

Other important classes in XASDI are Message, MessageRepository, and Driver. MessageRepository manages message exchange among CitizenProxy and environment and this class also works as an interface between Java environment and X10 environment to exchange messages in distributed X10 Places. Driver manages execution of the simulation with a corresponding thread. Each Driver is related to Places (and Citizens in the Places) where it has a responsibility for execution.

Finally, XASDI provides a logging mechanism. By preparing log definitions for the application, it can output simulation logs in each X10 Places.

7.5 IBM Mega Traffic Simulator

In this section, we introduce an application of XASDI for the city traffic flow.

We developed a traffic simulator with drivers' behavior models by analyzing the real probe car data that are obtained from GPS equipped on taxies in Tokyo (Osogami et al. 2012, 2013). The agent-based traffic simulator considers each microscopic vehicle as agent, which travels through a given road network with Crosspoints (nodes) and Roads (links). With the XASDI framework, the Vehicle class extends the Citizen class, the Road class extends the Place, and the Crosspoint class extends the Driver class.

Each agent is assigned an origin, a destination, and a departure time as a trip according to an origin-destination (OD) table obtained from population and traffic census survey data. The simulator creates the agent on the origin at the departure time. The agent chooses a route from the origin to the destination, according to a model of the route choice, and travels along that route. In this simulator, heterogeneous agents (vehicles) select a route with their probabilistic preference distribution estimated from probe car data and change their car speed and lane

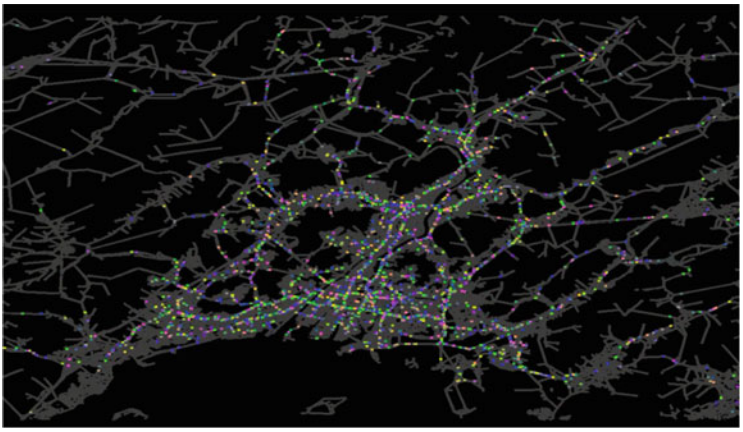


Fig. 7.11 Screenshot of traffic simulation in Hiroshima city

based on the car-following model (Gipps 1981) and Integrated Lane-Changing Model (Toledo et al. 2003) that represent the dynamic interaction with surrounding cars.

At each time step (typically, 1 sec), the microscopic car behavior in roads is controlled by connected Crosspoints, each of which is assigned an individual thread to effectively simulate the fine-grained car movement in and across roads even in a distributed HPC environment. The simulator tracks the location of each agent and records information of vehicles (position and speed), roads (average speed, number of vehicles, CO2 emissions on the road), and trips (travel time and total CO2 emissions of each vehicle) into log files which are used for analysis and visualization. The screenshot of the simulation is shown in Fig. 7.11.

7.6 Pedestrian and Shopping Simulation

In this section, we introduce the pedestrian and shopping simulation for shopping malls and city areas (Mizuta et al. 2016; Mizuta and Imamichi 2018).

By extending XASDI framework, we create Consumer agents (Citizen) that move and shop around the mall. In the mall, there are Zone and Shop (Place) at which a Consumer agent is located.

For the shopping mall simulation, we define Attraction, Move, Shop, and Purchase actions. Each action is related with models possessed by a Consumer agent to allow different behavior for each person. We implemented Attraction, Move, Shop, and Purchase models based on Hui's model (Hui et al. 2009) and combined them with a Walk model.

We consider a three-layer representation of the mall. One map layer is a concrete geospatial representation defined with 3D coordinate data and another is an abstract

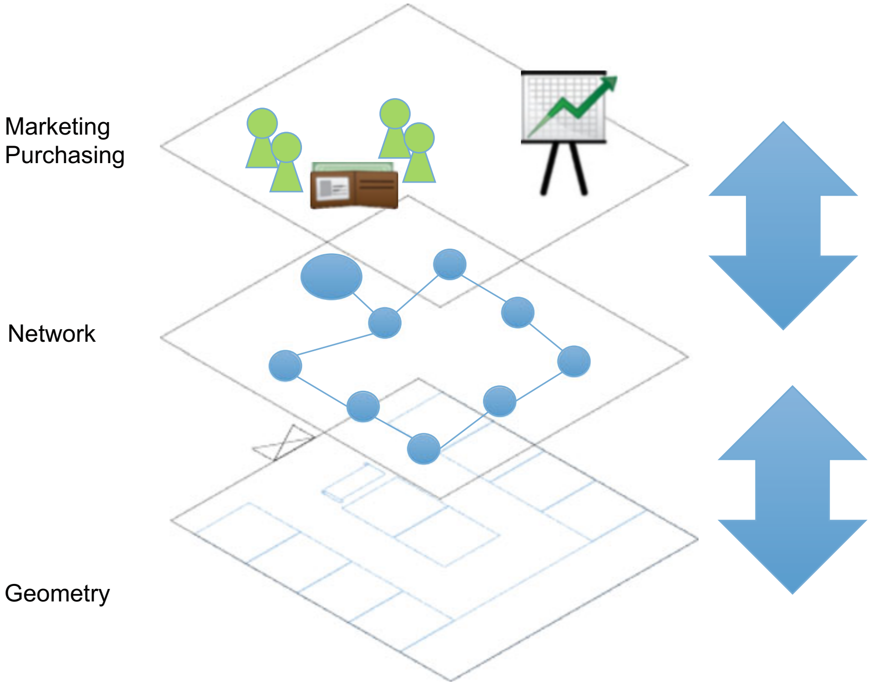


Fig. 7.12 Three-layer structure of shopping mall simulation

network representation defined with nodes and links. Each zone is a node and the adjacency relationship between two zones is a link. On top of these two map layers, we also consider a third economic layer that includes marketing and purchasing activities.

Figure 7.12 shows the three layers of the shopping mall simulation.

For economic activities including destination decision-making in the mall, we use the network representation. For pedestrian activities, we use the geospatial representation. This integration of economic behavior and moving behavior is a key factor of this agent-based simulation.

At the beginning of the simulation, the simulator reads the zone definition file and generates zone objects (including Entrance and Shop) and related GeoZones that represent geospatial locations simultaneously. Consumer agents enter the mall through the Entrance zone at uniform random times during the simulation.

At each time step, agents update their attraction for products and zones, as described later. The attraction is a numeric value associated with each agent (i.e., customer), where the value represents the level of interest of the customer to the zone, or products sold in the zone, at which the agent is located at a given time. Each agent may or may not have a destination. A destination is a zone in the network

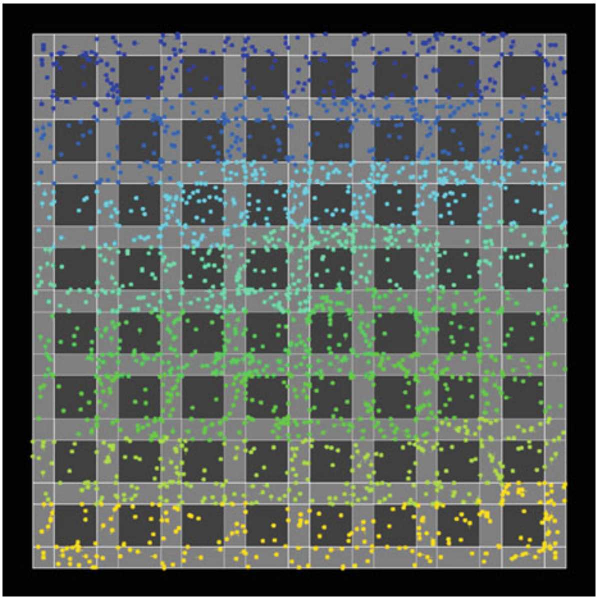


Fig. 7.13 Screenshot of shopping mall simulation

representation. If an agent does not have a destination, it selects a destination with the probability determined on the basis of zone attractions.

If an agent decides to stay in the current zone, or when it reaches the destination, it walks around the current zone using the pedestrian model described in the next section. After the agent decides to make one of the zones its destination, the shortest path on the network is chosen as a trip by Dijkstra's algorithm (Dijkstra 1959), and then the agent walks toward the next zone in the trip.

Figure 7.13 shows the screenshot of the walking consumer in a simple grid mall.

By extending the simulator for shopping mall, we develop a pedestrian simulator in a city. For the shopping mall simulation, we use very simple definition of zone map. Now, we utilize more realistic map definition that can be used for city planning or evacuation.

We extract road network data and building data in Sendai city of Japan from OpenStreetMap (OSM 2017) and convert it to zone map. We choose an area in front of Sendai station for our simulation case.

Figure 7.14 shows the screenshot of the pedestrian simulation in Sendai.

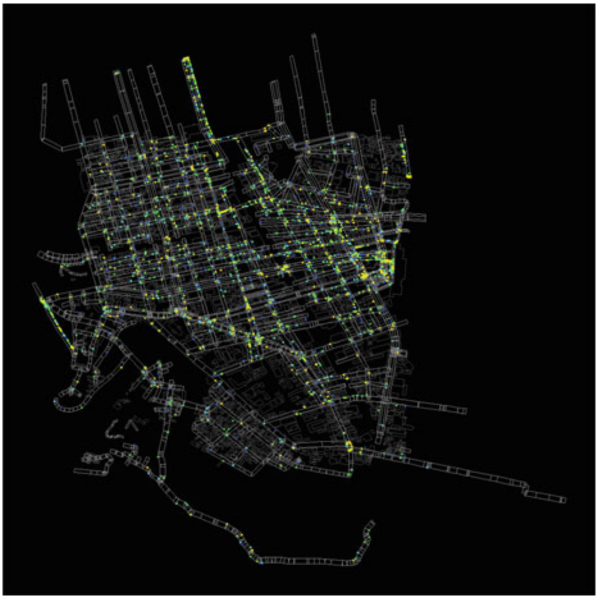


Fig. 7.14 Screenshot of pedestrian simulation in Sendai

7.7 Conclusion

In this chapter, we introduced a computational approach for complex service and social systems called agent-based simulation. Agent-based simulation can become a powerful tool for Service Science, Management, and Engineering (SSME).

In addition, agent-based social simulations are utilized to support the decision-making of city planners for various real social issues, recently. For such a purpose, we developed a large-scale social simulation framework "X10-based Agent Simulation on Distributed Infrastructure (XASDI)." XASDI is available as open-source software at GitHub with tutorial, API documents, and sample programs (https://github.com/x10-lang/xasdi).

We introduced our earlier work with a small number of agents and then describe the details of XASDI framework and its applications. We believe that we can support government and business decision-making through what-if analysis for various scenarios with these simulations.

Acknowledgments The work with XASDI framework and its applications was supported by CREST, JST.

References

Arthur WB, Durlauf SN, Lane DA (1997) The economy as an evolving complex system II. Addison-Wesley, Reading

Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1(1):269–271

Gipps PG (1981) A behavioural car-following model for computer simulation. Transp Res Part B Methodol 15(2):105–111

Grütter J (2000) World market for GHG emission reductions. In: The World Bank's National AIJ/JI/CDM strategy studies program, 2000

Hui SK, Bradlow ET, Fader PS (2009) Testing behavioral hypotheses using an integrated model of grocery store shopping path and purchase behavior. J Consum Res 36(3):478–493

Izumi K (1998) An artificial market model of a foreign exchange market. Ph. D. thesis, The University of Tokyo

Mantegna RN, Stanley HE (1999) Introduction to econophysics: correlations and complexity in finance. Cambridge University Press, Cambridge

Mizuta H (2016) Agent-based simulation for service science. In: Global perspectives on service science: Japan. Springer, Berlin, pp 177–191

Mizuta H, Steiglitz K (2000) Agent-based simulation of dynamic online. auctions. In: Proceedings of the 32nd conference on Winter simulation Society for Computer Simulation International, pp 1772–1777

Mizuta H, Imamichi T (2018) Large-scale social simulation framework "x10-based agent simulation on distributed infrastructure (XASDI)". In: The twenty-third international symposium on artificial life and robotics 2018 (AROB 23rd 2018), pp 784–789

Mizuta H, Yamagata Y (2001a) Agent-based simulation and greenhouse gas emissions trading. In: Proceedings of the 33rd conference on Winter simulation, pp 535–540

Mizuta H, Yamagata Y (2001b) Agent-based simulation for economic and environmental studies. In: Annual Conference of the Japanese Society for Artificial Intelligence. Springer, Berlin, pp 142–152

Mizuta H, Yamagata Y (2002) Transaction cycle of agents and web-based gaming simulation for international emissions trading. In: Proceedings of the Winter simulation conference, vol 1. IEEE, Piscataway, pp 801–806

Mizuta H, Yamagata Y (2005) Gaming simulation of the international co2 emission trading under the Kyoto protocol. In: Agent-based simulation: from modeling methodologies to real-world applications. Springer, Berlin, pp 72–81

Mizuta H, Steiglitz K, Lirov E (2003) Effects of price signal choices on market stability. J Econ Behav Organ 52(2):235–251

Mizuta H, Maeda K, Yoshihama S, Nakasuji H (2016) Agent-based simulation of movement and purchasing behavior of walking shoppers. In: The twenty-first international symposium on artificial life and robotics 2016 (AROB 21st 2016), pp 498–503

Namatame A, Terano T, Kurumatani K (2002) Agent-based approaches in economic and social complex systems, vol 2. IOS Press, Amsterdam

Noda I, Ito N, Izumi K, Yamashita T, Mizuta H, Kamada T, Murase Y, Yoshihama S, Hattori H (2015) Roadmap for multiagent social simulation on HPC. In: 2015 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology (WI-IAT), vol 3. IEEE, Piscataway, pp 22–25

OpenStreetMap (2017) http://openstreetmap.org/. Accessed 13 Dec 2017

Osogami T, Imamichi T, Mizuta H, Morimura T, Raymond R, Suzumura T, Takahashi R, Ide T (2012) IBM mega traffic simulator. IBM Research–Tokyo, Research Report

Osogami T, Imamichi T, Mizuta H, Suzumura T, Ide T (2013) Toward simulating entire cities with behavioral models of traffic. IBM J Res Dev 57(5):6-1

Spohrer JC, Maglio PP (2010) Toward a science of service systems Springer US, Boston, MA, pp 157–194

Steiglitz K, Shapiro D (1998) Simulating the madness of crowds: price bubbles in an auction-mediated robot market. Comput Econ 12:35–59

Toledo T, Koutsopoulos H, Ben-Akiva M (2003) Modeling integrated lane-changing behavior. Transp Res Rec J Transp Res Board 1857:30–38

XASDI (2017) X10-based agent simulation on distributed infrastructure (XASDI). https://github.com/x10-lang/xasdi. Accessed 27 April 2017

X10 (2017) The x10 parallel programming language. http://x10-lang.org/. Accessed 27 April 2017