Modeling Agents and Interactions for Simulating Social Systems

Daisuke Torii

Abstract

This thesis proposes a novel methodology for modeling agents and interactions in multiagent simulation for consensus building among stakeholders. When using multiagent simulations for this purpose, it is important not only that the domain experts can deeply understand stakeholders' actual behavior but also that the stakeholders can feel the simulation result as their solution. This research aims to provide a modeling methodology to satisfy these points.

The participatory method by social scientists is suitable for this purpose. The key idea is that stakeholders can understand the simulation result as their solution because they share the modeling process with the domain experts and can believe that the model reflects their actual behavior. A role playing game (RPG) where stakeholders play on a board representing their environment is used to share the modeling process and to well understand their actual thinking and behavior. "Participation" is the key aspect of our modeling methodology.

To simulate a social system as a multiagent system, two types of model are required: agents (internal models) and interactions. For example, in an economic system in agriculture described in this thesis, decision making models and negotiation models of farmers and seed suppliers are required. Hence, we have to consider a modeling method according to each character.

In modeling an agent (e.g. decision making) which is implicit in human, the identification of the model greatly depends on the modeler's ability. Therefore, a method for model validation is very important. On the other hand, in modeling interactions (e.g. negotiation) which are emerged

outside of human, it is rather important to show and capture continuously appeared interactions. RPG using a board in the participatory method is not enough to express such dynamic process as interactions, and it is important to give a method for this purpose.

In this study, to give solutions of each issue above, several technological methods are combined with the participatory method. Moreover, to confirm and discuss the proposed methods, they were applied to a case study of agricultural economics in the northeast of Thailand. The approaches and the results are as follows:

1. *Modeling agents with classification learning*

   To understand internal models of human (stakeholder) from a neutral standpoint independent with the modeler's ability, a hypothesis of domain experts is validated by a model which classification learning creates from RPG log data. The key ideas of this method are 1) a feature selection method for enhancing reliability of the learning result by classification learning, and 2) visualization of the learning result on a computer to promote understanding of domain experts in refining the learning result. The modeling process with these methods was actually applied to model farmers' decision making for seed suppliers. As a result, the domain expert's hypothesis was validated and several subjects for further investigation were found.

2. *Modeling interactions with participatory simulation*

   To well express interactions among humans, participatory simulation where user-controlled avatars and computer agents coexist is used. Here, the following two issues were tackled.

   - *Development of a platform for designing interactions with a legacy simulator*

     There exists many platforms to support developing and analyzing multiagent simulations, but they are not always suitable to design and evaluate interactions. Therefore, an effective method was proposed; first, to provide a computational model suitable for interactions, an interaction layer is constructed and connected from the outside of a legacy simulator. Next, to configure the agents interacting ability, a method for controlling the flow of information in the connection area is provided. As a concrete example, we realized an interaction layer by *Q* which is a scenario description language and connected it to CORMAS, which is often used in the field of the participatory method. Finally, we discussed the capability of our method through a concrete implementation.

- *Modeling process using participatory simulation*

  A process for modeling interactions was established, which contains both an RPG and a participatory simulation. RPG is suitable for communication between researchers and stakeholders and participatory simulation is suitable to present and capture negotiation process. For designing complicated interactions among humans and agents in participatory simulations, three kinds of interaction descriptions used in the platform mentioned above were defined and the designing process of them were established. Finally, a participatory simulation was developed on CORMAS/*Q* and we succeeded in deeply understanding and modeling negotiation among seed suppliers.

The proposed methodology has achieved the following two important points for multiagent simulations to build consensus among stakeholders. First, RPG and participatory simulation give shared experience of modeling with domain experts, which is important for stakeholders to find their solutions in the simulation results. Second, the modeling method with classification learning and participatory simulation enables domain experts to deeply understand and model stakeholders' actual behavior in the system. Indeed, the impact was confirmed in a real case study of agricultural economics in the northeast of Thailand.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Professor Toru Ishida at Kyoto University, for his continuous guidance, valuable advice, and helpful discussions.

I gratefully acknowledge valuable comments of other members of my thesis committee at Kyoto University, Professor Tetsuro Sakai and Professor Hajime Kita at Kyoto University. I am deeply indebted to my adviser, Professor Taichi Yuasa at Kyoto University for his kind advice and useful discussions.

I must express my special thanks to cooperation of Dr. Francois Bousquet, Dr. Guy Trébuil at Centre de Coopération Internationale en Recherche Agronomique pour le Développement (CIRAD) and Dr. Chirawat Vejpas at University of Ubon Ratchatani. They spent their precious time to organize our experiments and have discussions in Thailand.

I am grateful to Dr. Alexis Drogoul at Research Institute for Development for valuable advices and discussions in our collaboration with Laboratoire d'informatique de Paris 6 (LIP6). Also, I wish to thank Akishige Yamamoto at Mathematical Systems and Stéphane Bonneaud at European Center for Virtual Reality for his devotion to the project, hard work and precious discussion.

Lecturer Hirofumi Yamaki and Assistant Professor Hideyuki Nakanishi at Kyoto University always gave me kind advice and helpful comments.

Finally, I wish to thank all members of Professor Ishida's laboratory for their help and fruitful discussion.

This research was supported in part by followings:

- The young researcher development program of the JSPS 21st century COE program "Informatics Research Center for Development of Knowledge Society Infrastructure"
- The Japan Science and Technology Corporation for the Core Research for Evolutional Science and Technology Universal Design of Digital City Project.
- Grant-in-Aid for Scientific Research (A)(15200012, 2003-2005) from Japan Society for the Promotion of Science (JSPS)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Objectives

This thesis proposes a novel methodology for modeling agents and interactions in multiagent-based simulation (multiagent simulation) for consensus building among stakeholders. To this end, technologies which support model acquisition and validation are combined to the participatory method by social scientists [Bousquet 99, Bousquet 02, Gilbert 02]. Finally, the proposed methodology was evaluated with an actual case study on agricultural economics in the northeast of Thailand [Vejpas 04].

Compared to computer simulations using governing equations, multiagent simulation has advantages on the points that the target can be modeled with its complexity at a micro level and a multiagent model is straightforward to describe society consisting of humans. This is why this kind of simulation is used for various social systems (politics, finance, marketing, traffic, disaster management etc.) with various purposes (understanding of social phenomena, anticipation of social system, design of social system, training etc.).

In the existing studies of simulating social systems in multiagent models, the modeling has been based on such materials as literature, videos and surveys [Drogoul 02, Gilbert 99]. It can be said that validness of the model has been evaluated in two aspects; one is that the model is convincing enough to appropriately cut out a part of reality from such materials. Another is that the simulation reproduces the same kinds of feature as the real data.

When using multiagent simulations for consensus building among stakeholders, it is also important that the simulation actually supports solving the problems among stakeholders. Therefore, the modeling methodology should not only give the domain experts a deep understanding of stakeholders' actual behavior but also give the stakeholders an intuitive understanding of some of the possible scenarios that might evolve from their decisions in the simulation [Gilbert 02]. Especially, the existing modeling methods have not considered the latter point. The objective of this research is to provide a modeling methodology which satisfies these points.

The participatory method by social scientists is an effective approach for this purpose. The key idea is that stakeholders can understand the simulation result as their solution because they share the modeling process with the domain experts and can believe that the model reflects their actual behavior [Bousquet 99, Bousquet 02, Gilbert 02]. This differs from the traditional method in which researchers (domain experts) observe stakeholder's activity in a unilateral way by literature, videos, surveys etc. To take stakeholders into a modeling process, role playing games (RPG) and multiagent simulations have been introduced. RPG where stakeholders play on a board representing their environment is used to share the modeling process and to well understand their actual thinking and behavior. Meanwhile, a multiagent simulation is used to visualize domain experts' understanding and refine it with the stakeholders [Bousquet 99, Bousquet 02]. "Participation" is the key aspect of our modeling methodology.

To simulate a social system as a multiagent system, two types of model are required: agents (internal models) and interactions. For example, in an economic system in agriculture described in this thesis, decision making models and negotiation models of farmers and seed suppliers are required. The character differs from each other and therefore it is necessary to give a modeling method respectively. In the research of the participatory method, however, such methods considering the different character of modeling targets have not been deeply discussed. This research mainly discusses them and strengthens the existing participatory method in a technological aspect.

In modeling an agent (e.g. decision making) which is implicit in human, the identification of the model greatly depends on the modeler's ability. Therefore, a method for model validation is very important. In the participatory method, multiagent simulations are used to validate models created by the domain experts, but there has been no method for the domain experts to scrutinize the validness of the model by themselves.

On the other hand, in modeling interactions among humans (e.g. negotiation, cooperation etc.), all series of process emerge as behavior of the actor. Compared to the case of internal models, it is rather important to show and capture continuously appeared interactions. RPG using a board and papers in the participatory method is not enough to express such dynamic process as interactions, and it is important to give a method for this purpose.

In this study, to give solutions of each issue above, suitable technological methods are discussed and combined with the participatory method.

1.2 Approaches and Issues

This section describes our approaches for the objectives and issues in modeling agents (internal models) and interactions used in multiagent simulations for consensus building among stakeholders. To confirm and discuss the proposed methods, we used a case study of agricultural economics in the northeast of Thailand in co-project of IRRI (International Rice Research Institute) and CIRAD (Centre de Coopération Internationale en Recherche Agronomique pour le Développement) [Vejpas 04].

i. Modeling agents with classification learning

> A hypothesis of domain experts is validated by a model which classification learning creates from RPG log data. The merits are 1) classification learning creates an objective model without influence of the modeler's ability because the creation is based on the algorithm, and

2) classification learning is more robust than human in performance even when the number of elements to be compared is large.

What is important here is to consider a method that classification learning creates a model which logically explains the stakeholders' behavior even if the data available is sparse (it is difficult to gather enough data for classification learning because of the cost of RPG).

The evaluation was done in eliciting decision making of farmers about selection of seed suppliers in northeastern Thailand, using C4.5 [Quinlan 86, Quinlan 93] as a classification learning algorithm.

ii. Modeling interactions with participatory simulation

Participatory simulation where user-controlled avatars and computer agents coexist is introduced for modeling interactions. In participatory simulation, the actual situation (e.g. trading) can be realized on a computer because stakeholders join the simulation as avatars and interact with agents or the other avatars. Virtual spaces in computers are better than boards and papers used in RPG because it is possible to express all series of interactions and to give various roles to the agents for eliciting stakeholders' actual behavior.

Here the following two issues are tackled.

1. *Development of a platform for designing interactions with a legacy simulator*

   There exists many platforms to support developing and analyzing multiagent simulations. Although they are very useful for the domain experts to develop simulations, they are not always suitable to design and evaluate interactions. Therefore, it is important to develop a method constructing a simulator suitable for handling interaction models, where legacy simulators are effectively used as resources.

2. *Modeling process using participatory simulation*

   Stakeholders are not always familiar with computer systems and research processes, so it is not appropriate to use participatory simulation from the beginning in the modeling process. Moreover, in order to create an environment where human-controlled avatars and computer agents can negotiate and cooperate, it is necessary to give domain experts descriptions which regulate the behavior of agents and the participation of humans. The modeling process should also include the designing process of the descriptions.

The evaluation was done in modeling negotiation of rice seed suppliers in northeastern Thailand.

1.3 Thesis Outline

This thesis consists of seven chapters, including this chapter as the introduction.

Chapter 2 is dedicated to introduce the background of this thesis and describe importance of our approach. First, the characteristics of multiagent simulation will be shown: various purposes, platforms for simulation and modeling methodologies. Second, we will see the participatory method, especially Companion Modeling which is an iterative modeling process with role playing games and multiagent simulations.

Chapter 3 introduces the example which this study used to evaluate our approaches. The domain is agricultural economics in the northeast of Thailand in co-project of IRRI (International Rice Research Institute) and CIRAD (Centre de Coopération Internationale en Recherche Agronomique pour le Développement) [Vejpas 04]. Also, we will see how a role playing game is actually organized and held.

Chapter 4 describes a modeling method wherein classification learning is applied to RPG log data. This method makes it possible that researchers (domain experts) validate their pre-defined hypothesis using a model created by machine learning from RPG data. There are two points to be discussed. One is how to use classification learning in order to output reliable results for domain experts even from not enough amount of data. Another is how to refine the learning result by the domain experts in objective and intuitive way. Finally, we will show the result of applying this method for farmers' selection model of seed suppliers in the northeast of Thailand. The learning algorithm was C4.5 [Quinlan 86, Quinlan 93].

Chapter 5 explains a method to construct a suitable platform for verifying and designing interaction models with utilizing a legacy multiagent simulator. Moreover, we propose a method to express variation of personal ability in interactions and restriction of ability from environment by controlling the flow of information in the channel connecting the two systems (Connection Control). To show a concrete example of our method, two systems were actually connected: *Q* interpreter which gives interaction models in extended finite state automaton [Ishida 02b] and CORMAS which gives social and environmental modeling [Bousquet 98]. Finally, we discuss our approach through an example in CORMAS/*Q* to confirm its ability to handle fairly complex simulations.

Chapter 6 shows a process for modeling negotiation using participatory simulation where user-controlled avatars and computer agents coexist. The points to consider here are how to design a modeling process for stakeholders who are not used to a computer system and how to build a system of participatory simulation where several interactions among humans and agents are to be designed. For the latter point, we use the platform mentioned in chapter 5 to facilitate the interaction design. Finally, we will show the result of applying this method for negotiation models of seed suppliers in the northeast of Thailand.

Chapter 7 concludes the thesis summarizing the result obtained through this research and the prospect of the future research.

Chapter 2

Background

2.1 Multiagent Simulation for Social System

Computer simulation has been used in various fields (physics, biology etc.) for various purposes, e.g. a deeper understanding of the phenomenon, debug of models of systems, predicting future behavior, and performing experiments that cannot be carried our in reality [Davidsson 02]. The method most used in modeling is a top-down approach in which the features to be observed are simplified in a macro scope and expressed in governing equations. On the other hand, multiagent-based simulation (multiagent simulation) described in this section is a bottom-up approach which models an element (agent) and the interactions among elements[1]

An *Agent* is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators [Russell 02]. The interest of an agent research is its autonomy, organization, coordination and cooperation. In BDI architecture, real-time search, reinforcement learning and subsumption architecture, researchers have been investigating computational models intended to use in a real world. In blackboard model, contract net protocol, distributed constraint satisfaction, market-based model and mechanical design, researchers have been inves-

[1]Cellular automaton is also a well-known bottom-up approach.

tigating computational models for coordination and competition of many agents.

Simulations have been used in the agent research to demonstrate the effectiveness of theories or algorithms, test a developing system in an early stage, examine strategy for decision making and so on. Competition, used for a testbed to develop agents, can be said as a kind of simulations. The well-known cases are RoboCup Soccer League [Kitano 97], Trading Agent Competition (TAC) [Wellman 01] and so on. These simulations especially focus on development and verification of a new agent model. In this kind of simulation, the model tends to be complicated.

On the other hand, multiagent simulations have been started to use for deeply understanding social phenomena and analyzing/designing social systems. A multiagent model attracts attention from various application domains, especially from social scientists, because it gives a framework to directly express human society. Multiagent simulation is applied to various areas and they are currently being used in politics, economics, finance, marketing, traffic, disaster management, agricultural systems and so on. Simulations for a deeper understanding of social phenomena and analysis/design of social systems are generally called *social simulation*. The target of this thesis is social simulation by a multiagent model[2].

2.1.1 Applications of Social Simulation

The purpose of social simulation by a multiagent model can be classified as follows. Here we will clarify the feature of simulations for consensus building among stakeholders in this research.

1. **Understanding of social phenomena and discovery of theories**

[2]Computer simulations for social scientists are not limited to multiagent simulation. Micro-analytical simulation models, system dynamics and world model and queuing models are those of the examples. See the detail in [Gilbert 99].

Artificial society is one of research fields in multiagent simulation [Epstein 96]. This is simulation without reference to any specific real world target as a method of experiment, demonstration and discovery in social science. Sugarscape [Epstein 96] is a famous example. In this model, there are sugar in cellular automata and agents on the environment. This is used to describe various features of society (social networks, trade and market, cultural differentiation and evolution). For example, if agents start with an approximately symmetrical distribution of wealth (the amount of sugar each agent has stored), a strongly skewed wealth distribution soon develops. There are several other examples: a study on constraints on communication in markets in which there are some agents selling and others buying [Alvin 92], a study on the implications of agents having knowledge of future facts or events [Doran 97a] and so on.

An especially remarked concept in this purpose is *emergence*. This concept derives from complexity theory [Waldrop 92] for nonlinear systems. Emergence occurs when interactions among objects at one level give rise to different types of objects at another level. The simulation is used to understand relationship between attributes and actions of an individual (micro scope) and social aggregation (macro scope). For example, MANTA (Modeling ANthill Activity) [Drogoul 94] simulates the process that ants' society is being emerged.

A well known principle to understand social phenomena is KISS (Keep it Simple, Stupid) [Axelrod 97]. The KISS principle states that agent modeling should be simple even though the observed phenomenon is complex, and that complexity should be a result of the simulation. This is useful to enrich our understanding of fundamental processes that may appear in a variety of applications.

2. **Prediction**

If we can develop a model that faithfully reproduces the dynamics of some behavior, we can then simulate the passing of time and know the future. This is often used in such fields as economics, finance, marketing and so on. For example, in [Farrell 98] and [Said 02], it was successful that features of a real market was reproduced from behavioral models of consumers in a markets.

3. **Design of social systems**

   Multiagent simulations are used to design cooperation, control techniques and policies in a social system.

   In disaster management, Cohen *et al.* used multiagent simulation to design behavioral plans including cooperation of fire fighters and their commander for extinction of a forest fire [Cohen 89]. Helbing *et al.* modeled characteristics of crowd behaviors in panic situation from literature and videos, and finally showed appropriate ways to avoid dangerous situations and evacuate in a building by the simulations [Helbing 00]. RoboCup Rescue League is a comprehensive environment for simulations concerning to situations of a city after disaster and the rescue planning [Kitano 97, Takahashi 02].This a testbed for rescue strategies of agents in complex environmental changes such as earthquake and fire diffusion.

   Simulations in traffic generally study a navigation method to enhance the efficiency of both individuals with various destination and the whole system. For example, Balmer *et al.* proposed a method which reproduces more real traffic situation [Balmer 04] and Yamashita *et al.* studied strategies for deciding a route under the condition which shares route information [Yamashita 05].

   Simulations for institutional design aim to develop an institution or a policy which brings balance between common stake and the whole system. For example, Mizuta *et al.* uses simulations for making rules of the international greenhouse gas (GHG) emissions trading

[Mizuta 02]. The following two researches apply multiagent simulations for consensus building among stakeholders. One is water resource management in five cities in Europe (FIRMA project) [Downing 00, Gilbert 02]. Another is natural resource management such as land use, forest management and so on (Companion Modeling) [Bousquet 99, Bousquet 02]. These use gaming simulations where stakeholders participate in order that reseachers deeply understand the real behavior of stakeholders and stakeholoders share the experience of modeling. This thesis targets this type of simulation.

4. **Training**

   Multiagent simulation gives new types of application called *participatory simulation* where computer agents and user-controlled avatars. Computer games familiar in entertainment can be said as a kind of participatory simulations. For research purposes, it has been used in military training [Rickel 02] and evacuation training [Nakanishi 04a]. Training systems such as flight simulators aim to learn how to operate the equipment while participatory simulations focus on social interactions between agents and avatars.

The classification above is based on the main purpose of each simulation. Its process contains several elements. In order to design a social system, it is necessary to understand the feature of the system then give an appropriate design to the system. Moreover, in order to decide the validness of design, it is necessary to predict the influence caused from the design with some accuracy.

2.1.2 Platforms for multiagent simulation

In multiagent simulation, it is necessary to model agents and their surrounding environment. Especially, in modeling agents, we need to model not only their internal model[3] (knowledge, brief, desire, intention etc.) but also interactions with other agents and the environment. There are many systems for multiagent simulation because it is very costly to model them from the scratch. Some simulators provide visualization tools for observing a process of simulation and analysis tools of simulation results. These days, multiagent simulation has been used in various application domains such as social science and some simulators aim to provide a platform to facilitate domain experts who are not familiar with programming multiagent models. Here, we will see various multiagent simulators developed today.

Swarm[4] is a software package for multiagent simulation of complex systems developed at the Santa Fe Institute [Minar 96]. It is intended to be a useful tool for researchers in a variety of disciplines, especially artificial life. The basic architecture of Swarm is the simulation of collections of concurrently interacting agents: with this architecture, a large variety of agent-based models can be implemented.

RePast[5] is a free software framework for creating agent-based simulations using the Java language [Collier 02]. It provides a library of classes for creating, running, displaying and collecting data from agent-based simulation. RePast borrows much from the Swarm simulation toolkit and can properly be termed 'Swarm-like'.

NetLogo[6] is a programmable modeling environment for simulating natural and social phenomena. It is particularly well suited for modeling complex systems developing over time. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many individuals. NetLogo let students open simulation and 'play' with them, exploring their behavior under various conditions. It is also an authoring environment that enables students, teachers and curriculum developers to create their own

[3]We call an internal model "agent" in contrast of "interaction" among agents as individual entities.

[4]http://wiki.swarm.org

[5]http://repast.sourceforge.net/

[6]http://ccl.northwestern.edu/netlogo/

models.

CORMAS[7], which is used in this study, is programming environment dedicated to the creation of multiagent systems, especially for the domain of natural resources management [Bousquet 98]. It provides a framework for developing simulation models of coordination modes between individuals groups that jointly exploit common resources.

MadKit[8] is a Java multiagent platform built upon an organizational model [Ferber 98]. It provides general agent facilities, such as lifecycle management, message passing and distribution, and allows high heterogeneity in agent architectures and communication language, and various customizations. Madkit communication is based on peer-to-peer mechanism which allows developers to develop distributed application quickly using agent principles.

The UMASS Multi-Agent System Simulator (MASS) is designed for creating sophisticated autonomous agents that are reactive to their environment, and that perform goal-oriented decision-making under constraining conditions such as deadlines and resource tradeoffs [Vincent 01]. MASS agents are created using the component-based Java Agent Framework (JAF), and environment simulation uses the Task Analysis, Environmental Modeling and Simulation language TAEMS [Decker 98]. MASS also has exploited the notion of making seamless, flexible transitions between simulated and real-world tasks. Using JAF, the MASS-specific components in an agent can be replaced with components that interact directly with the real world, allowing for a separation between the agent's logic and its environmental interactions. MASS, JAF and TAEMS are freely distributed[9].

Caribbean[10] is a Java framework including a run-time environment for building Web applications that provide asynchronous, event-handling, and monitoring services, as well as synchronous services [Yamamoto 01]. Each user can keep a persistent, event-driven object, such as an agent, at server

[7] http://cormas.cirad.fr/indexeng.htm

[8] http://www.madkit.org/

[9] http://dis.cs.umass.edu/download.html

[10] http://www.alphaworks.ibm.com/tech/caribbean

side. Caribbean agents are implemented as event-driven object. When the object is active, it is allocated in a memory but if there are too many objects, inactive objects are swapped out into the secondary storage device. In this way, Caribbean can realize a large scale multiagent simulation.

MACE3J is a Java-based MAS simulation, integration, and development testbed, with a supporting library of components, examples, and documentation, distributed freely [Gasser 02]. MACE3J currently runs on single and multiprocessor workstations, and in multiprocessor cluster environments such as Sun multiprocessor clusters or the SGI Origin. The MACE3J design is multi-grain, but gives special attention to simulating very large communities of large-grain agents. It exhibits a significant degree of scalability, and has been effectively used in fast simulations of over 5,000 agents, 10,000 tasks, and 10M messages, and on multiprocessor configurations of up to 48 processors.

FreeWalk[11] is a platform where human participants and autonomous characters can socially interact with one another in a virtual city space [Nakanishi 04b]. The applications of FreeWalk include 3D chat, multi-user training, and visual simulations. FreeWalk has such three capabilities as human behavior visualization, model-based simulation, and multi-user participation. The multi-user multi-agent architecture of FreeWalk allows Internet users to take part in a large-scale online simulation, e.g. an evacuation from a virtual central railway station. The simulation scenario of FreeWalk can be written in Q language [Ishida 02b].

2.1.3 Modeling for Social Simulation

For simulating a social system, it is necessary that the agent model should appropriately reflect reality. Here, we will see various processes for modeling.

The following process is the ideal set of steps in using simulation in the social science introduced by Gilbert and Troitzch [Gilbert 99].

[11] http://www.lab7.kuis.kyoto-u.ac.jp/freewalk/

1. *Observation of the target*: After defining the target to be modeled, the target is observed in order to provide the parameters and initial conditions for the model.

2. *Designing a model*: The target is simplified and the conceptual model is designed. The most difficult thing here is to decide what needs to be left out and what needs to be included.

3. *Building the model*: The conceptual model is transformed to a computer program.

4. Simulation: The simulation is executed and the results are recorded.

5. *Verification*: The program is checked if the simulation is actually doing what is expected. It is essential to 'debug' the simulation carefully, preferably using a set of test cases.

6. *Validation*: Validation concerns whether the simulation is a good model of the target. Validity can be ascertained by comparing the output of the simulation with data collected from the target. Also, it is necessary to check how sensitive the model is to slight changes in the parameters and initial conditions: *sensitivity analysis*.

The following examples of multiagent simulation are basically organized in the steps above.

Said *et al.* uses simulations to study the effect of marketing strategies in a market with the element of competition [Said 02]. That requires having a consumers' behavioral model allowing the representation of observed individual behaviors and the simulation of a large population of consumers. The authors firstly constructed behavioral model of consumers based on literature of sociology, psychology, economics and business science (marketing). This model is behavioral attitude consisting of social processes which include *imitation process* and *conditioning process* and personality traits which include *mistrust*, *opportunism* and *innovativeness*. These five parameters are adjusted by generic algorithm to build a first virtual agent consumer population which is compliant with a real market. The virtual consumers created in this process were successful in illustrating the feature of a real market.

Helbing *et al.* uses simulations to explore ways to avoid dangerous situations and ways to appropriately lead evacuees in panic [Helbing 00]. One of the most disastrous forms of collective human behavior is the kind of crowd stampede induced by panic, often leading to fatalities as people are crushed or trampled. Authors firstly extracted the features of evacuees in panic from literature of socio-psychology, video, socio-psychological literature, reports in the media and available video materials, empirical investigations, and engineering handbooks. Next, they modeled the crowd behavior from the perspective of socio-psychology and physics. Finally, they confirmed that the model could simulate the real situation then proposed an appropriate evacuation method.

Balmer *et al.* proposed a new agent-based modeling method to reproduce urban traffic flows [Balmer 04]. The existing main method outputs the features of an average travel time etc. based on the O-D (Origin-Destination) matrixes which express emergent traffic in each area and the destination. Authors tried to reproduce a real situation by modeling agents based on the O-D matrixes with a probabilistic method. Finally, their simulation succeeded in gaining a better recall ratio than results in the past using real traffic data in the area of all of Switzerland.

In training, it is necessary that virtual humans take appropriate actions for the training and crowds move with reality (e.g. the variability of moving). It is also important for users to feel the scene alive. For example use of simulations as training, Rickel *et al.* gave virtual humans (agents) roles by scenarios for educate users about realistic situations at a battlefield. What is important here is how agents react to actions of the users and how realistic situations in a battlefield are created [Rickel 02]. Murakami *et al.* proposed a method to model agents' behavior in an evacuation training based on data of a real controlled experiment. The authors succeeded in reproducing the feature of the controlled experiment in a 3D space [Murakami 03].

In this thesis, we intend to use simulations for consensus building among stakeholders. In such simulations, it is more important that stakeholders can understand the simulation result as their solution than that the simulation precisely reproduce the reality. We will see the participatory method for this purpose in the following section.

2.2 Participatory Method for Modeling

2.2.1 Characteristics of Participatory Method

Recently, a purpose of multiagent simulation as a participatory method by social scientists is to give participants an intuitive understanding of some of the possible scenarios that might evolve from their decisions in a complex domain where various related things have to be considered. Therefore, the simulation model is designed not to deliver a specific prediction, which is an over-ambitious aim as a consequence of the complexity of the system. At the same time, the simulation is not used to evaluate whether a policy is good or bad but to make an institution or a policy.

In traditional modeling process, since all contacts with the model are mediated by the researchers, stakeholders are just shown the simulation result and have to take on trust that the simulation is a good model of the situation. Therefore, the results will not give them any intuitive understanding of the dynamics of the model and they hardly understand the result as their solution [Gilbert 02].

In all these, the stakeholders (or their representatives) and the researchers work together as a team towards shared objectives, with each contributing their own expertise. To this end, role playing games and multiagent simulations are used.

Several original characteristics of the participatory method are summarized below. This method offers new approaches in making a model which reflects reality and evaluating the model by potential users.

1. The RPG makes it possible to extract an unconscious understanding of stakeholders and to construct a model that reflects reality.

2. Simulations that have the same features as the RPG are organized and stakeholders who do not have expert knowledge can evaluate the model and the simulations result by themselves.

3. The shared experience of modeling enables stakeholders to understand the simulation results as what may occur in the future.

Moreover, this method is effective for extracting reality, even if applied to stakeholders in farm villages who are unfamiliar with the research process or the internal working of computers.

The participatory method by social scientists are systematized as Companion Modeling by a research group around CIRAD. There are many applications to natural resource management [Bousquet 02, Tébuil 02, Vejpas 04]. Another example is the FIRMA project applied to environmental management in water resource [Downing 00, Gilbert 02]. The next section will explain the modeling process of Companion Modeling.

2.2.2 Companion Modeling

The Companion Modeling methodology [Bousquet 99, Bousquet 02], which is one approach in Participatory Modeling, requires the validation and refinement of an initial model constructed from relevant literature, field surveys, etc., with stakeholders through role-playing games (RPG) and a multiagent simulations which are iteratively held, providing an improved understanding of reality after each loop (iteration. see Figure 2.1.). Finally, a multiagent simulations of the model created from this methodology are organized and the process shows a conceivable outcome in the future. This ComMod process is used to enhance collective learning for adaptive management of common resources.

Fig. 2.2 shows detail processes of Companion Modeling. After several iterations of RPG, the process using multiagent simulation is usually done.

In a process using RPG, an initial agent model is firstly created from relevant literature and surveys. Next, a RPG using a board that represents the stakeholders' surrounding environment is organized. The RPG can be

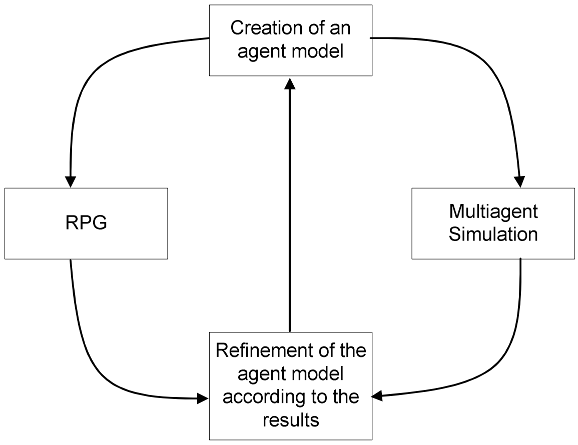


Figure 2.1: Overall process of companion modeling: companion modeling is iterative process of RPG and multiagent simulation.

used for the validation and improvement of the initial model. After this RPG, the stakeholders are interviewed to better understand the motivations behind their decision-making. The initial model is improved through the analysis of the interview and the RPG log data (This process is shown in Fig. 2.2 (A)).

After the RPG, a multiagent simulation is organized using a graphical interface that has the same characteristics as the board used in the RPG. The stakeholders then evaluate the results of simulations. The stakeholders can easily understand this simulation and can give suggestions on modifications to the model because of their experiences with the RPG. In many cases, after more than one RPG, a model of the multiagent simulation is made, using the model refined from the results of the RPG (This process is shown in Fig. 2.2 (B)). In this way, it is possible to reach an in-depth and shared understanding of reality through the iteratively held RPG and multiagent simulation.

There are many application of Companion Modeling in agricultural fields of Africa and Southeast Asia (mainly Thailand and Vietnam). Scenarios to effectively solve their problems are proposed by positive exchange

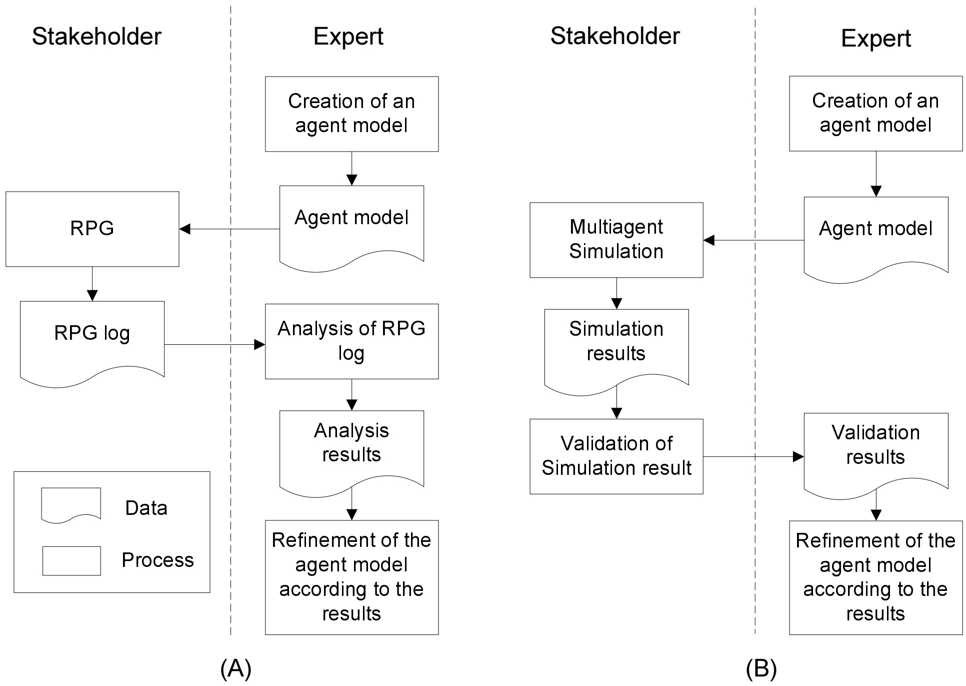


Figure 2.2: Detail processes of companion modeling: the left is the one using RPG and the right is the one using multiagent simulation.

of opinions with stakeholders through RPGs and multiagent simulations. The concrete examples are as follows:

- Appropriate solution of farmers' land use and distribution of labor power in the mountain region of Vietnam (Bac Kan) [Bousquet 01a, Castella 01]

- Solution to conflict of stakeholders (government, farmers, forest managers and so on) about use of forest resource and land in a high land of North Thailand (Maehae of Chiang Mai distinct) [Bousquet 01b, Tébuil 02]

- Agricultural economics in northeast of Thailand: understanding of rice production of farmers and the seed supply chain [Vejpas 04]

Chapter 3

Agricultural Economics in Northeastern Thailand

In this chapter, we will explain agricultural economics in the northeast of Thailand in co-project of IRRI (International Rice Research Institute) and CIRAD (Centre de Coopération Internationale en Recherche Agronomique pour le Développement) [Vejpas 04], and describe how a role playing game is actually organized and held.

3.1 Background

The lower northeast Thailand subregion contains nine provinces covering 8.4 million ha, with 17,357 villages and 11.5 million people. According to a survey done by the Rice Research Institute during 1982-86, more than 1,500 rice varieties were grown in northeast Thailand. The government has been making a high investment for a long time to release better varieties (according to rice scientists' criteria) as recommended varieties and produce and supply seeds to farmers. About 77% of the farmers in the northeast have adopted these three recommended varieties.

It has been reported that most farmers in the northeast are still using their own rice seed, but more farmers tend to buy seeds and also to change seeds more frequently. However, the production capacity of the government for rice seed is only 3-5% of the demand. Therefore, more and more organizations and projects are becoming involved in the rice seed supply system, but no integrated information on rice seed supply systems of different agents and its linkage with variety and seed management system have been reported. Especially, the deficiency in rice seeds required by farmers is a problem. Therefore, it is necessary to investigate current rice supply system and decision making process of farmers and improve rice seed production systems. However, at the same time, it is important to consider conservation of rice biodiversity.

There is no common platform for stakeholders, particularly for farmers, who should have their required varieties match their consumption needs and field conditions and should have good-quality seed for agronomic and marketing aspects, while the public institutions conserve rice biodiversity as valuable genetic sources and as alternative varieties. To understand the complexity of the system, the participatory modeling approach is used for better knowledge integration and communication of different perception. A main purpose is to provide a better understanding of the system's behavior, to be able to identify the key constraints that should help finding possible solutions. This can lead to establishing a decision support system for serving farmers' needs in rice production systems and to harmonizing stakeholders' roles and objectives as well as conserving biodiversity under dynamic and multilevel circumstances.

In summary, subjects researchers are studying are as follows: 1) understand farmers' needs concerning rice varieties and the selection model of rice varieties and 2) to identify the problems of the current delivery system of rice varieties by acquiring a farmer's selection model of seed suppliers and a flow model of rice seeds among the seed suppliers and farmers.

3.2 Targets of Modeling

To realize this economic system in multiagent simulation, four models are needed: a) decision making model of farmers for rice varieties, b) decision

Table 3.1: Seed suppliers in the northeast of Thailand

Supplier	The formal name	Explanation
RRC	Rice Research Center	A government agency which manages and supplies rice seed. It supplies the best quality of seed.
SC	Seed Center	A government agency which multiplies the seed from RRC and distributes it to suppliers and farmers in the province. They also technically support AC.
DAO	Distinct Agricultural Office	A supplier which provides information and seed to CSC or farmers.
ST	Seed Trader	A retail distributor which purchase seed and sells it to farmers.
AC	Agricultural Cooperative	An agricultural cooperative which multiplies the seed purchased and distributes it to farmers.
CSC	Community rice Seed Center	A supplier which multiplies seed from DAO and SC, and sells it to farmers.
BA	Bank for Agriculture	A bank which gives loan to farmer members and also distribute seed.

making model of farmers for seed suppliers, c) negotiation model of suppliers for rice seeds and d) negotiation model of farmers for rice seeds.

As described above, a multiagent simulation for a system involved with several kinds of stakeholders, we have to understand internal models and interactions and a modeling method suitable for each target should be developed.

The modeling method for an agent internal model will be described in chapter 4 and be applied to the case of b). The one for agent interactions will be described in chapter 6 and be applied to the case of c).

Table 3.1 shows the actual suppliers and their roles in the northeast of Thailand. In chapter 4, the target is farmers' decision making model how to select one supplier from the suppliers shown in table 3.1. In chapter 6, the target is a supplier's (Seed Center: SC) negotiation model how to access suppliers and how much seed to buy or sell.

Figure 3.1: A farmer playing RPG (the left man standing): he is making a decision about rice varieties and the fields to put them

3.3 An Example of Role Playing Game

Now we will introduce the detail of a RPG practiced to understand the behavioral model of farmers. The log data for this RPG is actually used in chapter 4.

In this project, a RPG was organized to research the characters of the farmers in the urban and rural areas. The first RPG took place in '03 sep. at Ubon Ratchathani, a city located in the North Eastern part of Thailand, with 12 farmers participating. The next was in '04 Jan. at Ban Bua Ngarm, a small poor town located 80 kilometers south to Ubon Ratchathani, with 13 farmers consisted of three towns and 2 ethnics (Lao and Khmer).

In the RPG, a 3D board (60cm square) reflecting the landscape of the farmland is used. The board contains the landscape of fields in higher, middle, and lower levels. It is said that lower leveled fields are easier to keep water and therefore is better to grow rice. As the variety of rice and their place to obtain are decided, a label indicating them is attached by assistants

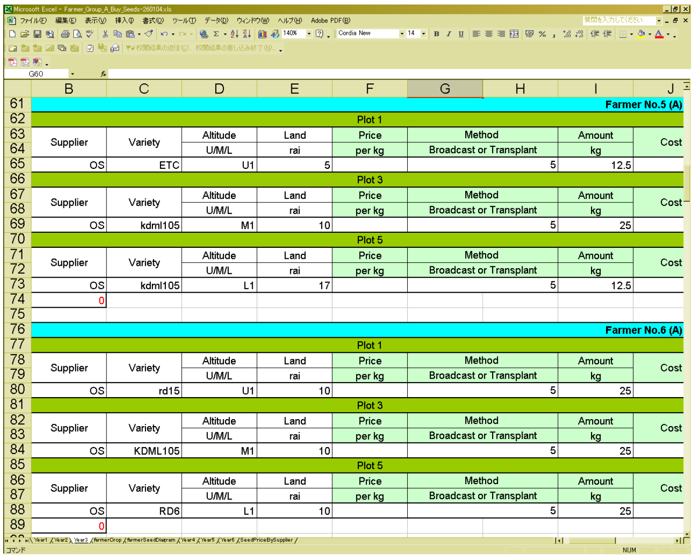


Figure 3.2: An Excel sheet for recording selection of farmers

(students of Ubon Ratchathani University) (Figure 3.1). Around 6 farmers are assigned to each board.

Each farmer is given the farmland on the 3D board, close to actuality as possible. Generally, rich farmers own their land in diversity from the higher to the lower levels. On the other hand, poor farmers own most of their land in the higher levels which are said not to be favorable to grow rice. To add more reality, each farmer is given some fictitious money to buy rice plants at the beginning of the game.

The RPG was conducted in 6 steps for '03 Sep. and 4 steps for '04 Jan. (One step stands for one year.) For each step, the participants choose their rice varieties (out of 4 kinds plus others) and where to buy them (out of 10 traders). Then they decide where to grow which variety (within the farmland they own) and how to grow them (Whether to start from plants or seeds). In the end of the year, they make a decision how much of their crop to be consumed for their families immediate needs, to be kept for the next year or to be sold off, for each variety. These decisions are summarized in a

Figure 3.3: A farmer being paid his gain from selling

Excel sheet for each step (Refer to figure 3.2). Gain from selling their rice is paid to the participants by fictitious money. Figure 3.3 is a picture taken during the RPG. The man is being paid his gain from selling. In the end of the RPG, each group has an interview to reveal the process of their decision making.

Chapter 4

Modeling Agents with Classification Learning

In this chapter, we will discuss the issues and the solutions in using classification learning to model agents (internal models), and explain the modeling process. This method was evaluated with an example: decision making of farmers for seed suppliers. The decision making model is represented in a decision tree and we selected the well-known decision tree learning algorithm "C4.5" for classification learning [Torii 05b, Torii 05c, Torii 06b].

4.1 Introduction

In modeling an agent (e.g. decision making) which is implicit in human, the identification of the model greatly depends on the modeler's ability. Our approach is to apply machine learning to RPG log data and create an objective model without influence of the modeler's ability. The learning result can be used to validate the domain experts' hypothesis and find subjects for further investigation. Therefore, we did not choose deductive learning which treats a hypothesis by domain experts as prior knowledge, but did choose classification learning, one kind of inductive learning, which creates alternative classification knowledge from data without prior knowledge.

A task of classification learning in this method is acquiring classification knowledge that logically explains the internal model of the RPG participants from the RPG log data. Many applications on classification learning aim to acquire a meaningful learning result from a large amount of data, but we seek to acquire the one recognized as a human internal model by the domain expert, even from insufficient amounts of data. To solve this task, we have to develop a method more than simply applying a learning algorithm with RPG log data. It is indispensable to effectively use the expert knowledge.

The related works are as follows. U-MART [Kita 03] is a related work that applies machine learning to log data of a gaming simulation. U-MART is a test-bed system for an artificial market and the computer agent learns an optimal trading strategy in a simulation environment where humans and agents participate together. Our research differs in that stakeholders' models, which reflect reality, are acquired from the RPG log data using a machine learning system. A related work, that selects features using expert knowledge for classification learning is written about in [Ishino 95]. This method acquires a necessary feature subset through questionnaire data that has many features and the data gathered through interactions between a machine learning system and an expert. In this work, the expert does not know which features are important before applying the machine learning system. In our research, the expert knows important features beforehand, and our approach is different in that the important features are used as prior knowledge in a feature selection method, which narrows down the features.

Section 4.2 will explain a problem in applying the classification learning to RPG data and an approach to solve this problem. Section 4.3 will describe a process of model construction that uses classification learning (C4.5 [Quinlan 86, Quinlan 93]) with an actual example of case study on rice production in the northeast of Thailand. After discussing our proposed method in section 4.4, the final section will summarize this chapter.

4.2 Issues and Approaches

4.2.1 Definition as a Classification Learning Problem

The Characteristics as a classification learning problem are as follows:

1. Although the domain experts create a hypothesis from relevant literature and survey findings, they believe that other factors influencing classification can exist, but do not know how they relate to the classification. Therefore, the number of features tends to be large because the experts take into account all elements that they think might be related to the classification. However, the more features that are included, the more data sets tend to be needed to assure reasonable learning performance [Almuallim 94, Kohavi 97]. Unfortunately, it is difficult to gather enough data because the cost of gathering data through RPG is high (e.g. an RPG session usually engages 12-15 players for 2 days).

2. Noise data can affect the learning result because the amount of data may be insufficient.

3. The learning result gained through learning should not only be able to correctly classify data, but should also logically explain the real behavior of stakeholders. This means that the classification condition, which consists of the learning result, should have a relevant context that explains reality and satisfies the domain expert.

4.2.2 Approaches

In our approach for these issues, expert knowledge is effectively used for obtaining a model even from insufficient amount of data sets. Here, the unneeded features are winnowed in two ways:

Feature Selection Method

The wrapper approach[1] [Kohavi 97] is used for eliciting a model with higher classification accuracy for unknown data; that is, a model with more generality. This method identifies a feature subset that maximizes the prediction accuracy as determined by cross validation[2], by eliminating irrelevant features from the initial feature set through hill-climbing search. In this method, however, when the quantity of data sets is insufficient, there is a possibility that some features needed for adequately reproducing reality will be eliminated due to noise data. Therefore, in our approach, the features contained in the hypothesis (in a decision tree) are regarded as being candidates to adequately explain reality and retained in the process (e.g. five features in figure 4.3 are not eliminated in this process). These features are always contained in the feature subset, but the learning algorithm selects a part of features from the subset, so the important features are not always contained in the learning result. This means that the wrapper approach can winnow the features to prevent the learning from selecting irrelevant features which do not ensure generality.

Visualization Method

In order to obtain a result which logically explains the stakeholders' behavior, the learning model is refined by eliminating irrelevant features contained

[1]A method using a searching algorithm that identifies a feature subset which gives the best performance of a model outputted from an inductive learning. In many cases, cross validation is used to estimate the performance and hill-climbing is used to search a feature subset with the best estimated performance. In hill-climbing, there are two searching direction: one is backward elimination which begins at the full set of features and eliminates one feature which makes the estimated performance the best. Another is forward selection which begins at the empty set of features and adds one features which makes the estimated performance the best.

[2]Data sets are divided into n groups without overlapping. A machine learning program is applied to the n-1 groups and the result is tested by one unused group. The classification accuracy of the test data is used as criterion for the performance estimation. Switching the group used for the test data, this process is repeated n times. We assigned n to 10 which is the frequency generally used.

in it and applying the classification learning again. Such conditions are selected by the domain expert. Here, It is important to consider an evaluation method that gives domain experts intuitive understanding of the model and excludes their bias to the model. To this end, the environment and participants of the RPG are reproduced on a computer using a graphical interface and computer agents. The expert is shown the results made by the agents who have the learning model and the RPG log data itself (the expert is not told which is from the RPG log data). Then he puts comments to each result. After this, the model itself is directly shown to the expert. He selects irrelevant features, comparing his comments to the visualized result with the model.

There are three merits on this method. First, a visualized method helps the expert to find apparently strange results and eliminate irrelevant features. Second, the learning result can be neutrality evaluated, while the expert tends to criticize the learning result when the model is shown from the beginning because he is obsessed with his hypothesis. Third, he can scrutinize the learning result because a model is re-evaluated based on his comments to the visualized result. The reason why RPG log data is also shown to the expert is to convince him of validity of the learning result (he can accept a part of the learning result when the output of it resembles the one from the RPG log even if it does not support his hypothesis.).

4.3 Modeling Process with a Real Example

4.3.1 Overview of Process

We constructed five steps in which the approaches above are combined. In Figure 4.1, the corresponding numbers with each step are shown.

STEP1: Survey and Interview to Stakeholders

An initial model and data items to be corrected for RPG are defined from relevant literature and interviews with stakeholders. Also, important features are extracted from the initial model for STEP3.

STEP2: Role Playing Game

RPG sessions are organized and Log data is recorded.

STEP3: Output of a Reliable Result from Classification Learning

Learning data is created by transforming the RPG log data into a format for the classification learning. Next, irrelevant features are elim-

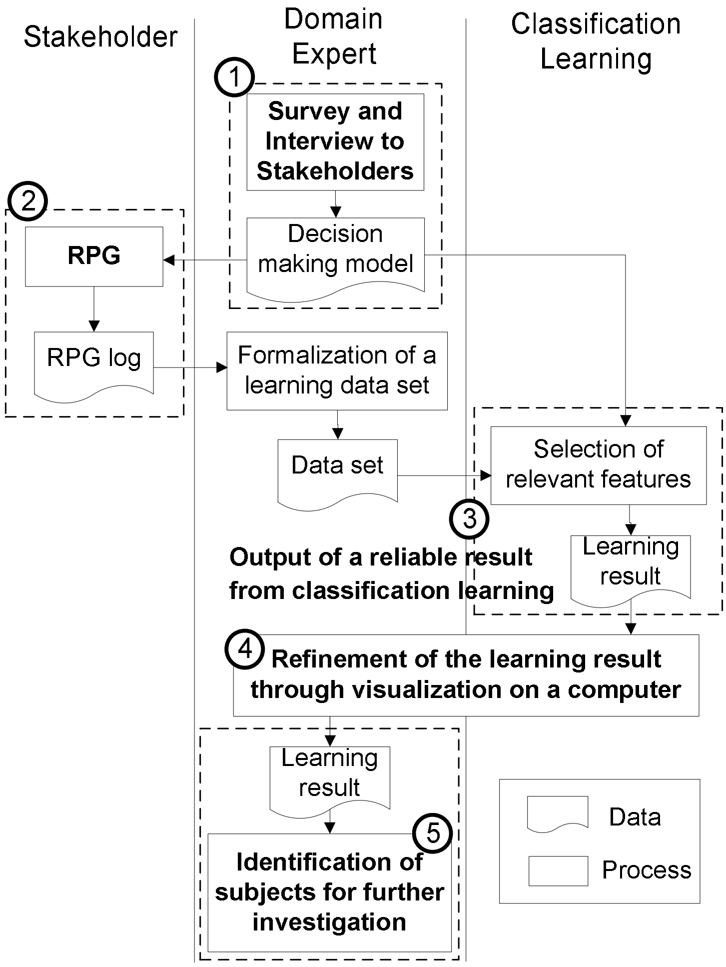


Figure 4.1: Overview of modeling process for agents (internal models)

inated by the feature selection method described in section 4.2.2. Finally, a learning result is outputted using the feature subset.

STEP4: Refinement of the learning result through visualization on a computer
Using the visualization method described in section 4.2.2, the learning result is assessed by domain experts. They eliminate irrelevant features and the classification learning is applied with the reduced set of features. This process is repeated until they are satisfied with the result.

STEP5: Identification of subjects for further investigation
By comparing the hypothesis (the initial model) with the final learning result, the domain experts find subjects for investigating in the future.

Figure 4.2 shows the detail process for modeling decision making model used in our example with decision tree learning algorithm "C4.5". The following section will explain the detail of the process with the results gained by applying them to a farmer's selection model of rice suppliers concerning the rice varieties widely distributed for commercial production.

4.3.2 STEP1: Survey and Interview to Stakeholders

Domain experts create an initial model (decision tree) from relevant literature and interviews with stakeholders. Important features are extracted from the initial model for STEP3. Also, data items to be corrected in RPG are defined (these will be also features in classification learning).

In our example, Figure 4.3 is the initial model. From this decision tree, important features ("*Change Variety*", "*Original Supplier is RRC*", "*Previous Seed Class*", "*Close to Ubon*" and "*Accessibility of RRC*") were extracted. The data items to be corrected in RPG (these will be the features for classification learning) are described in Table 4.1.

The abstract of this hypothesis is as follows (Table 3.1 in chapter 3 explained the role of each supplier). A seed supplier who takes an important role is *RRC* (Rice Research Center: a government agency which manages

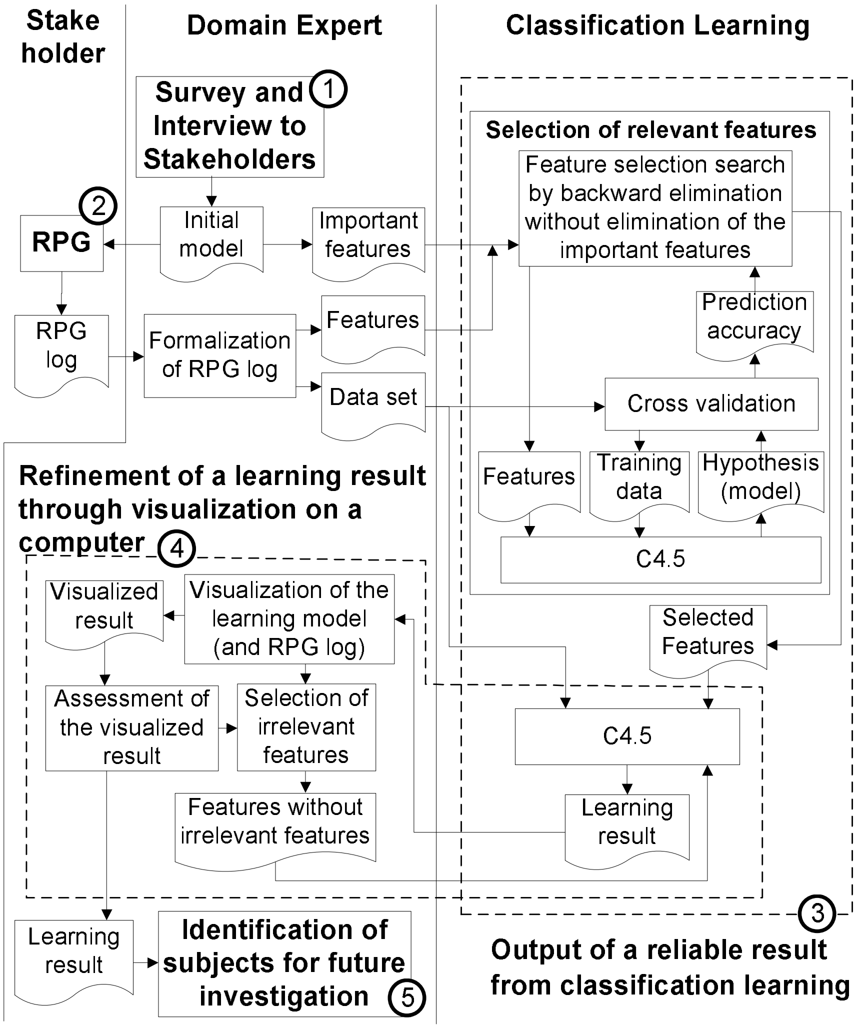


Figure 4.2: Detail modeling process for internal models with C4.5

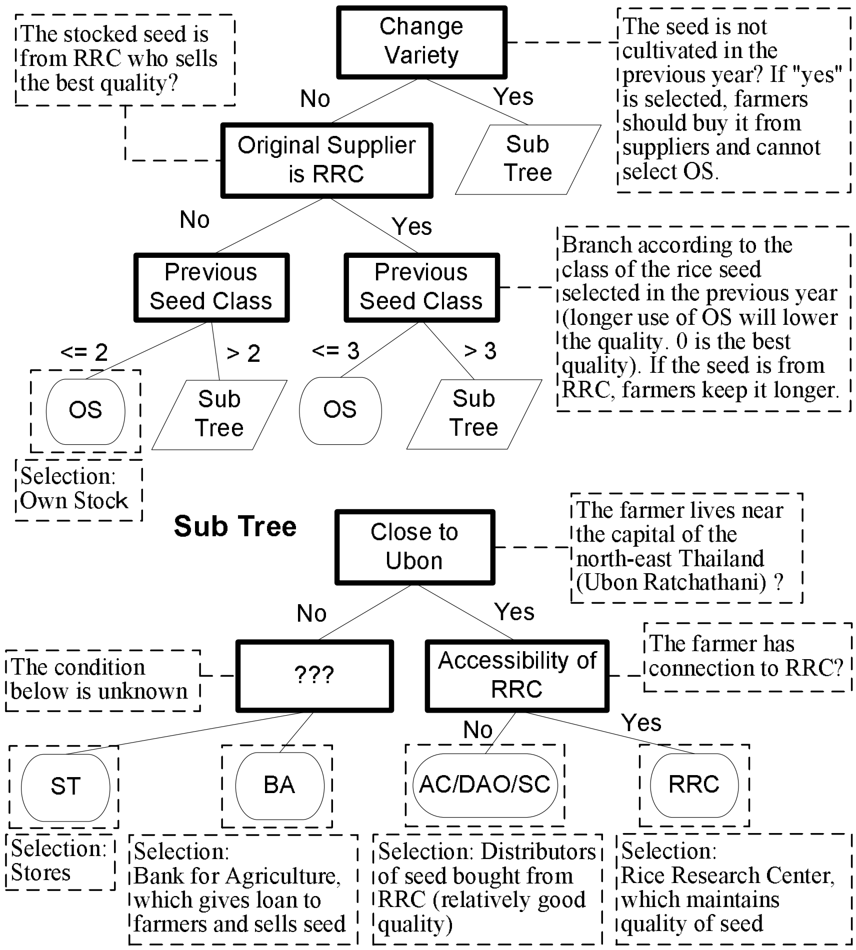


Figure 4.3: The initial decision tree model: The rectangles surrounded by bold lines are features. Five kinds of feature except "???" are the important one.

Table 4.1: Features and its selection process: the selected features ("s" in the table) and deleted features ("d" in the table) through this modeling process. In STEP1 (the item name is "1"), "s" means selection of the domain expert. In STEP3 (the item name is "3"), "s" means selection by C4.5 and "d" is deletion of the feature selection method. In STEP4 (the item name is "4"), "s" means selection by C4.5 and "d" is deletion of the domain expert.

Attribute	Explanation	1	3	4
Change Variety	The selected rice is not cultivated in the previous year?	s		
Original Supplier is RRC	The stocked rice is from RRC?	s		
Previous Seed Class	Seed class of the rice selected in the previous year.	s	s	s
Close to Ubon	The farmer lives near Ubon Ratchathani?	s	s	s
Accessibility of RRC	The farmer has connection to RRC?	s	s	s
Farmer Name	ID of the participant (farmer).		d	
Step	One step represents one year.			
Ethnicity	Ethnicity of the participant			
Family Number	Family number.			
Accessibility of SC	The farmer has connection to SC (Seed Center).			
Total Size of Land	The total size of land assigned to the farmer.			
Size of Upper Land	The size of upper land assigned to the farmer.			
Size of Middle Land	The size of middle land assigned to the farmer.		d	
Size of Lower Land	The size of lower land assigned to the farmer.			
Previous Supplier	The supplier selected in the previous year.		d	
Original Supplier	The supplier which the stocked rice was bought from.		s	d

and supplies rice seed. It supplies the best quality of seed.). *OS* is also important selection (Own Stock: this means that farmers use their own stock of seed. If they continue to use this, the "*Seed Class*" will be worse every year.) . *AC/DAO/SC* is a group of relatively large suppliers in the suburbs of Ubon Ratchathani which is the capital of northeastern Thailand. They also mediate seed to the other suppliers. *BA* (Bank for Agriculture) and *ST* (Seed Trader) are a supplier which retails seeds to communities far from the city and farmers. *OF* (Other Farmer) means that seeds are brought from the other farmers. The model shows a relationship between features (elements of decision making) and these suppliers (or seed sources).

The sub tree can be said as a pure selection model of seed suppliers in case of not using their own stocks (*OS*). "*Close to Ubon*" shows that the farmer is close to Ubon Ratchathani which is the capital of northeastern Thailand. *RRC* and *AC/DAO/SC* exist relatively near the city. The following *Accessibility of RRC* shows if the farmer has a route to be retained from *RRC* which provides good quality of seeds. In our example, it is uncertain how to decide a suppliers in the case that the farmer lives far from the city ("*Close to Ubon* = no") and so the condition is shown as *???*.

In the main tree, the first element to classify is "*Change Variety*", which shows whether the farmer selects the same seed as the previous year or not. In case he (or she) changes, he must get seeds from the other supplier and can not select *OS*. In case he uses the same variety, he can select whether to use his own stock or to buy from a supplier. The value of "*Previous Seed Class*", the class of seeds he used in the previous year (O is the best value), has an basic influence on this decision. The period that the farmer keeps using the same seed depends on "*Original Seed Supplier is RRC*" which shows if the seeds the farmer bought was from *RRC* which provide the best quality (If he bought from *RRC*, he kept using for three years.

4.3.3 STEP2: Role Playing Game

RPG sessions using a board that represents the stakeholders' environment are organized. Log data for each stakeholder is recorded.

In our example, two RPG sessions were held. A board representing farm land (which has lower, middle and upper fields) was created (see Figure 4.4-(a)). Totally, we got log data of 25 stakeholders.

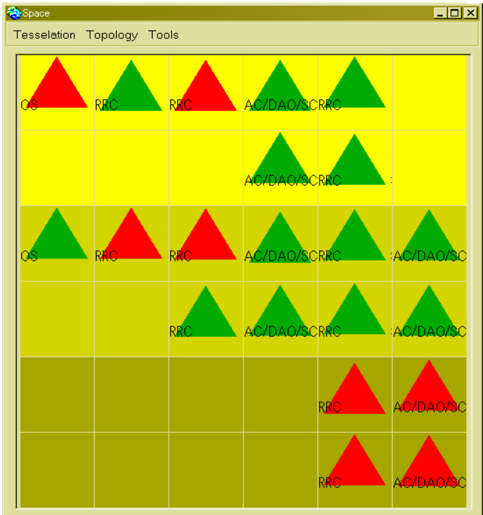
4.3.4 STEP3: Output of a Reliable Result from Classification Learning

Learning data is created by transforming the RPG log data into a format that the classification learning algorithm can understand. Next, the feature subset that offers the best prediction accuracy is identified from all the features and data sets created in the initial setting. The search method for the feature subset is a hill-climbing search that eliminates features one by one with a backward search. With regards to performance estimation, which is a criterion for selecting an eliminated feature, the prediction accuracy from the cross validation method is used. During each step of the search, all patterns of a feature subset that can be created by eliminating one feature from the current feature subset are created, and each prediction accuracy is outputted from the cross validation with C4.5 at each pattern. The one that improves prediction accuracy the most becomes the next feature set. When prediction accuracy cannot be improved any further, the search is terminated and the feature subset is outputted. What is important here is that the important features selected by the expert are not eliminated at any point.

In our example, first, the data sets were transformed from the RPG log data. Each data set expresses a farmer's selection at each step (which corresponds to one year). The number of the features was 16 and the number of the data sets was 80. The classification accuracy of the initial decision model (see Figure 4.3) was 58.0%.

Next, the feature selection method was applied while five of the features identified in STEP1 were not eliminated. Finally, the feature selection method eliminated 3 features shown in Table 4.1 and we got the first result shown in Figure 4.5 from the remaining 13 features. The feature selection method increased the prediction accuracy from 66.2% to 71.2%.

(a) RPG board [Vejpas 04]



(b) Visualized interface

Figure 4.4: Interfaces for modeling: These are examples used in a case study of rice production in the northeast of Thailand. Both show upper, middle, and lower lands. The cards on (a) and the triangles on (b) represent selected rice varieties and suppliers.

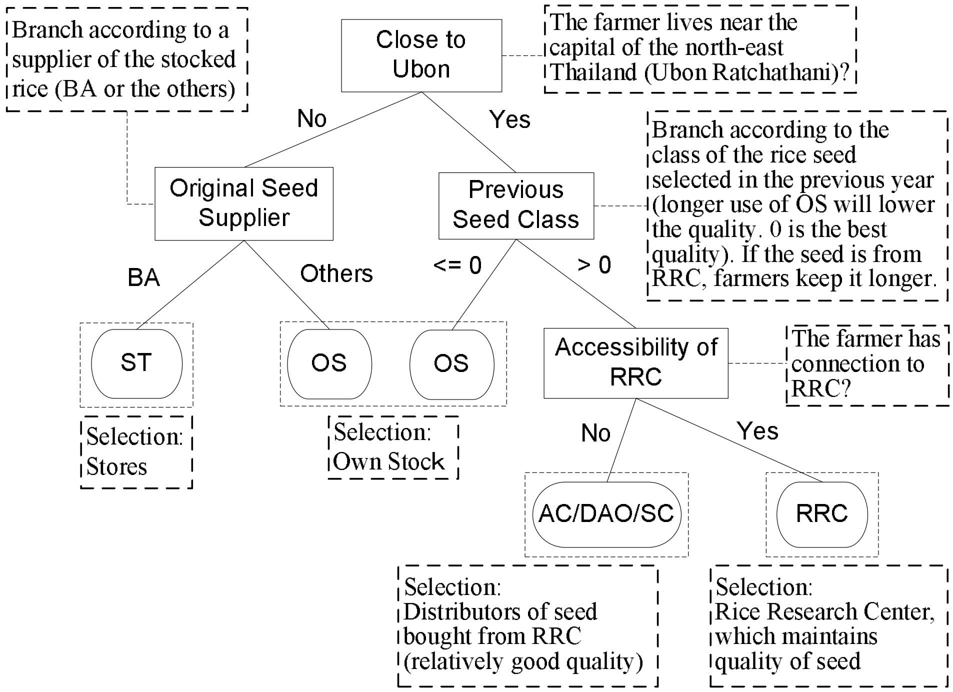


Figure 4.5: The first learning result

4.3.5 STEP4: Refinement of the learning result through visualization on a computer

The environment and participants are reproduced on a computer using a graphical interface and computer agents. The learning model is given to the experts to process using a kind of Turing test; the experts are shown two types of results: results from the learning model and RPG data. After this, the model is directly shown to the experts. When there are irrelevant classification conditions in the model, the expert will eliminate that feature which consists of the conditions and C4.5 will be applied using the resulting feature subset. This process is repeated until the expert is satisfied with the learning result.

In our example, we implemented the first learning result shown in Figure

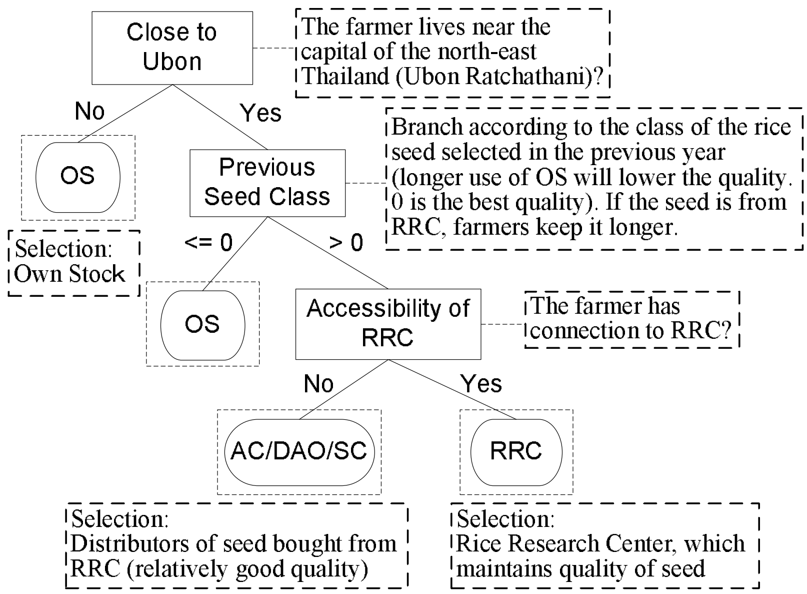


Figure 4.6: The finalt learning result

4.5 in CORMAS [Bousquet 98] (Figure 4.4-(b)) and evaluated the model by showing the results to the expert. At each step (equivalent to one year), the expert carefully observed the results and gave comments about the decision making process of each agent.

After this, the first learning result was evaluated by directly showing the decision tree shown in Figure 4.5 to the expert. The expert could then understand the reasons for the phenomena in the simulation. After investigating all the conditions of the decision tree, the expert decided that a condition consisting of the "*Original Seed Supplier*" was not relevant to reproduce reality. Therefore, this feature was removed and C4.5 was applied again. The result is shown in Figure 4.6. This time the expert had no problems with the evaluation, so we decided on this as the final learning result. In this result, the prediction accuracy was 68.8% and the classification accuracy was 75.0%.

4.3.6 STEP5: Identification of subjects for further investigation

By comparing the experts' hypothesis (Figure 4.3) with the learning result, the experts investigated their hypothesis and find subjects for investigating in the future. These will be discussed among the domain experts or investigated by repeating the process from STEP1.

In our example, we interpreted the final results in the following way. The condition "*Close to Ubon = no*" means that many farmers who live far from the city tend to use their own stock seed (*OS*). Farmers near the city use *OS* in the case of the "*Previous Seed Class <= 0*", which refers to farmers who bought seeds from a supplier in the previous year. (When seed is bought from a supplier, the seed class is set to the best value 0. Otherwise, the seed class is increased and the rice quality will be down.) These are the two branches of choice in the case of "*Previous Seed Class > 0*", which means that the farmers used *OS* in the previous year. *RRC* (who supplies the best quality of rice seeds) is chosen if they have a connection to *RRC*, otherwise *AC/DAO/SC* is chosen.

The experts evaluated that the learning model does not perfectly classify farmers' decision making, but it logically explains the behavior of the farmers in the RPG. This is because the essence of the expert's hypothesis partially emerges in the learning model. "*Close to Ubon = yes*" leads conditions "*Accessibility of RRC*" and the larger value of "*Previous Seed Class*" leads not *OS* but *RRC* or *AC/DAO/SC*. Moreover, the prediction accuracy (75.0%) and the classification accuracy (68.8%) of the learning model were improved from the accuracy which the hypothesis of the expert correctly classifies the RPG data (58.0%, which can be the classification accuracy or the prediction accuracy). Therefore, we consider that generality could be enhanced in a numerical value.

These results highlight the following subjects for further investigation. The subjects mentioned below need to be investigated through a follow-up survey with stakeholders (farmers), discussion among the experts and some future RPG. First, an important difference between the results and the hypothesis is the threshold value of the "*Previous Seed Class*". The value is 0 in the result, but 2 or 3 in the hypothesis. It is necessary to investigate that this is peculiar to the RPG or generally explained phenomena. Second, two features that were contained in the hypothesis "*Change Variety*" and "*Original Seed Supplier is RRC*" are not contained in the result. There is the possibility that this result was caused by insufficient data, so we cannot conclude that these two features are unnecessary, and we need to investigate the importance of these two features. Finally, it seems that the result does not perfectly explain the farmers' decision making process. (For example, one third of the data was wrongly classified as *OS* when it was actually "*Close to Ubon = no*".) There might be important features that the experts do not recognize except in the initial feature set in this experiment; therefore, we need to investigate such features further.

4.4 Discussion

We can explain the effectiveness of our approach in the following way: first, the feature selection method reduced the cost of the expert refining the learning model. In our example, when the expert eliminates features without the feature selection method, it is necessary for the expert to iterate the refinement process three times, but when using the feature selection method, just one round of refinement was enough, because the feature selection method eliminated three features. In our approach, considering the preparation costs of the visualization method used by the expert for the refinement of results, it was desirable for the expert to reduce the refinement process. Therefore, the feature selection method was indispensable. Moreover, we confirmed the effectiveness of our proposed method in which the features the expert thinks important are not eliminated in the feature selection method process. In our experiments, important features like these were often eliminated when this method was not used, and the experts were not satisfied with the results.

Second, we consider that the evaluation method of the learning model using the visualization was useful for gaining a better understanding of the characteristics of the model and for facilitating the collaboration with the expert. The expert was able to make various comments about the decision making of agents who used the learning model because the interface had the same characteristics as the RPG. This is effective because it leads to more comments than when the learning model was shown directly to the expert, which had been tried in the early stages of our research. In this example, the threshold value of the "*Previous Seed Class*" is 0, which led to a result where *OS* and one of the other suppliers were displayed in turn. The expert made many comments concerning this phenomenon. Although this differs from the hypothesis, it was finally understood as being what occurred in reality. This is because the expert found the same phenomenon in the results from both the learning result and the RPG data, so it can be said that this visualization method was effective that the expert could neutrally evaluate.

It is impossible to certificate that the model gained from our method is the most general and the best one extracted from the RPG log data. However, our method was successful in convincing the domain expert that the learning result could explain (the part of) the real behavior of the stakeholders, using the feature selection method and the visualization method. This is especially important point and therefore the experts can seriously investigate the subjects identified in our method in the future.

We still need to evaluate our method in other examples, but we think that the effectiveness of our method is confirmed insofar as it can be used as tool for validating the hypothesis of an expert from RPG data.

4.5 Summary

In this chapter, we proposed a novel method for modeling agents (internal models). Here, a hypothesis of domain experts is validated by a model which classification learning creates from RPG log data. This method has made it possible to understand internal models of human (stakeholder) from a neutral standpoint independent with the modeler's ability. This method was actually evaluated with farmers' selection model of seed suppliers in the northeast of Thailand.

There are three problems in applying classification learning for this purpose: 1) It is difficult to gather enough data for the number of features because the cost of gathering data is high. 2) Noise data may affect the learning result because the amount of data may be insufficient. 3) The learning result should be logically explained as a human internal model.

There are two key ideas for approaching these problems:

1. A feature selection method for enhancing reliability of the learning result by classification learning was proposed. In this process, the important features chosen by the expert were always included. In our experiment, this approach succeeded in creating learning results which satisfied with the domain expert from RPG data which has large number of features and not enough amount to guarantee the learning performance. Moreover, the refining cost of domain experts was reduced because this approach automatically eliminates irrelevant features, which is very important because the cost of visualization for refinement by domain experts is relatively high.

2. A visualization method on a computer was proposed for refinement of the learning result by domain experts. Here, decision based on the learning result and RPG data itself are shown to the domain experts (they do not know which is RPG data). This approach enables domain experts to evaluate the learning result in objective and intuitive way. In our experiment, this approach was successful in gaining more comments to the learning result than the way to directly show it. Moreover, the domain experts could accept that the learning result explained the situation of RPG, because they were shown two types of data.

Five modeling steps combined with these methods was established: "Survey and Interview to Stakeholders", "Role Playing Game", "Output of a reliable result from classification learning", "Refinement of the learning result through visualization on a computer", "Identification of subjects for further investigation”. These steps were actually applied to the real example. Finally, the domain experts’ hypothesis was validated and several subjects for further investigation were found.

Chapter 5

Creating Interaction Layer on Social Simulation

As mentioned in chapter 2, various simulators have been developed for various purposes, but they are not always suitable enough to design and verify interaction models. In this chapter, we will see a method to construct a simulator fit to handle interaction models, making an effective use of systems already developed [Torii 05a, Torii 06a].

5.1 Introduction

In multiagent simulation, agents change their own state and make actions to their environment based on negotiation/coordination with the other agents or observation of their environment. If designing of interactions is the central interest in the simulation, it is preferable that various interaction descriptions can be tested based on computational model for interactions.

However, most of the existing simulators which provide a programming framework for agents and environment are valuable because they equip convenient tools facilitate simulation (representation, analysis etc.). However, most of them provide a framework to describe behavioral models of agents in general using purpose-built API which help easy construction of agent functions but do not provide description framework based on a computational model for interactions (the architecture is shown in Figure 5.1). It is not realistic to develop such a platform from a scratch.

Our approach is to create an interaction layer which provides interaction descriptions based on a computational model over the existing simulator well used in the domain. This approach gives the domain experts an environment suitable for design of interaction models, and besides, the representation and analysis tools equipped with the simulator are still available. In our approach, moreover, in order to configure the agents interacting ability and refine interaction model in more realistic situation, a method for controlling the flow of information in the connection area is provided (Connection Control).

This chapter will mainly describe the technical aspect to archive this approach. For realizing our approach, a scenario description language *Q* was layered on CORMAS, which is well used in a domain of natural resource management. CORMAS/*Q* is also used in a participatory simulation in the next chapter.

The following sections are as follows: First, the proposed method will be described. Next, we will see integration of CORMAS and *Q*. Finally, capability of this method will be discussed by implementing a simulation of forest fire [Cohen 89].

5.2 Creating Interaction Layer on Social Simulation

5.2.1 Architecture

Figure 5.2 shows the architecture where an interaction layer is created outside of a multiagent simulator. This simulator consists of a social layer and an environmental layer (In the figure, the border is a solid line). The social layer consists of two layers: one is an interaction layer which interprets descriptions for negotiation/coordination (interaction description) and sends interpreted results to the agents. Another is an agent layer where the

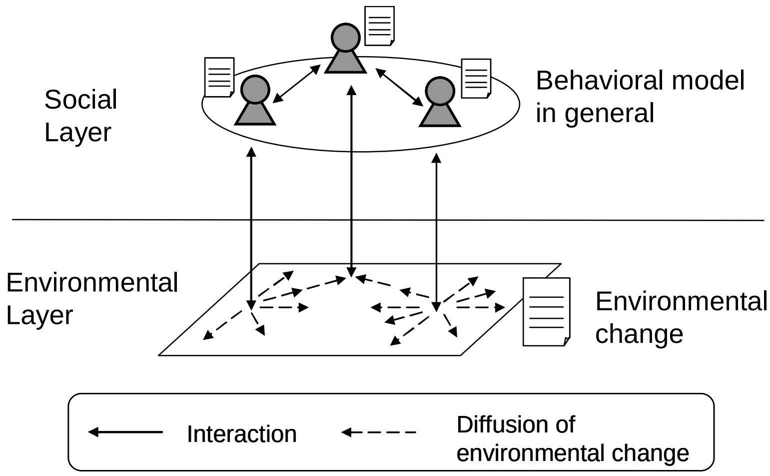


Figure 5.1: Normal architecture of social simulation

agents who receive the interpreted results make actions like communication of messages, observation from environment and action to environment in the simulator. In the interaction layer, interaction description is provided by a language based on a computational model suitable for describing interactions. In the agent layer, it is necessary to describe how agents act in the simulator according to the order from the interaction layer. The simulator also needs to give descriptions of diffusion of environmental change in the environmental layer.

As described in Figure 5.2, a single interaction model is assigned to a single agent. In this architecture, each agent is controlled by the interaction description, so even if the description is made in a view of a few agents, the interaction layer finally should make orders for a single agent.

5.2.2 System Modules

Interaction description requires the model interpreter which interprets the description according to the computational model and the execution result from an agent as an input. The interpreted result consists of interactions to

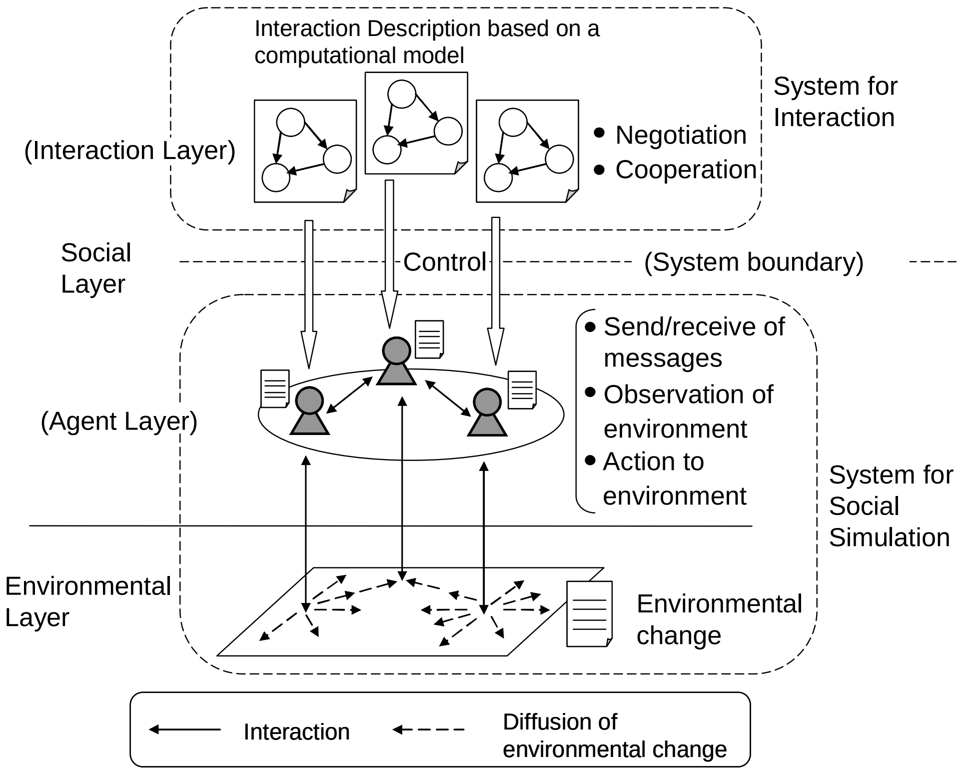


Figure 5.2: Architecture of social simulation where interaction layer is created

be executed and their execution order. For example, when there are several transitions from a state in a state transition model, interactions to be executed are a set of conditions for transition and the execution order is "parallel". The amount of interpreted results to be sent to an agent should be smaller as possible because more such results need more complex mechanism for processing in the simulator and especially it is difficult to change the inside in the existing simulator.

On the other hand, when receiving the execution request, the agent executes a corresponding action in the simulator and then sends back the execution result to the model interpreter. It is necessary to build these functions

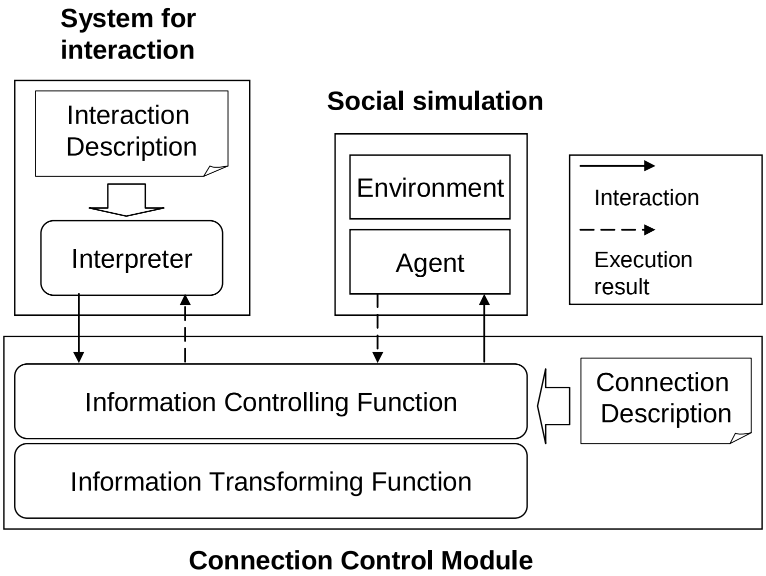


Figure 5.3: System modules

into agents in a simulator in order to realize the architecture.

Between the two systems, a Connection Control Module is necessary to intermediate the input and output from both systems. This consists of functions which exchange and adjust input/output data. An information transforming function has roles for exchanging information between different systems based on specification of serialization and protocol (e.g. a file sharing and TCP protocol). An information controlling function realizes Connection Control described in the next section.

5.2.3 Data Sequence in an Example

This paragraph explains how agents communicate each other in the architecture, using the example of Fire Fighter in section 5.5.1. The data sequence is shown in Figure 5.4. In this example, a fire fighter agent and a fireboss agent exist in the simulator. In the interaction layer, the model interpreter reads interaction scenarios and sends out commands to the agents in the simulator.

This example starts when the fire fighter encounters fire on their route. First, the interpreter in the interaction layer interprets the scenario and asks the fire fighter to send an inquiry of a modified route to the fireboss. Then, the fire fighter actually sends the inquiry to the fireboss in the simulator. After sending the inquiry, the fire fighter sends a completion report to the interaction layer. Next, the interpreter reads the next part of the scenario and asks the fire fighter to wait the reply for the fireboss. Meanwhile, when the fireboss who has been asked from the interaction layer to observe for any message from fire fighters receives the inquiry from the fire fighter, he will send a report of receiving inquiry of modified route from the fire fighter to the interaction layer. Then, the interpreter returns a request of sending a modified route to the fire fighter. Receiving this request, the fireboss recalculates a suitable route and send it to the fire fighter. Finally, the fireboss returns a completion report to the interaction layer. In the same way, when the fire fighter, waiting for a message from the fireboss, receives the modified route and returns a completion report to the interaction layer.

5.3 Connection Control

5.3.1 Interactions on Simulation

In multiagent simulation which assumes multiagent and environment, we can generally classify the types of interactions involved by agents into the following three groups:

1. communication with the other agents
2. observation from the environment
3. action to the environment

Information consisted by these three interactions flows between the interaction layer and the simulator regardless of the computational model adopted for interaction description.

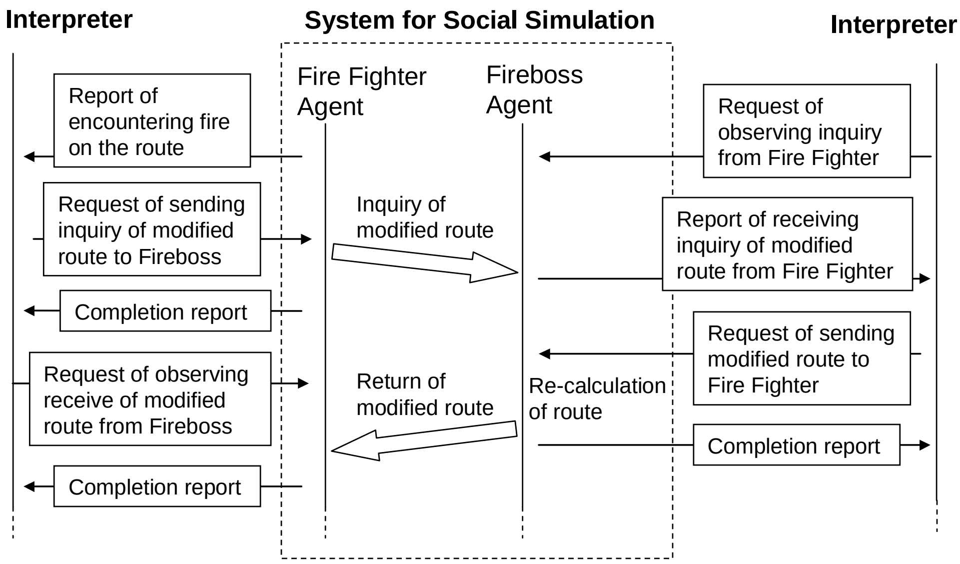


Figure 5.4: Data sequence in a case of the Fire Fighter model

5.3.2 Definition

We propose a method for designing more realistic interaction models without changing the inside of a simulator by controlling the flow of information in the channel connecting the two systems (Connection Control). This can express variation of personal ability in interactions and restriction of ability from the environment. This control is configured by "Connection Description", which is prepared except the interaction descriptions.

The following two operations can be performed on each interaction in section 5.3.1 and are defined in Connection Description[1]

1. to lose information (accuracy)

2. to delay information transmission (delay)

[1]we cannot assure these two cover all possible operations. However, they will make it possible that the agent ability is variously adjusted and interaction models are verified from various views.

The accuracy is defined in “%”. For example, when it is 80%, one in five messages is lost at the connection bridge. The delay is defined in a number based on unit time used in the simulator. For example, if it is discrete time, the delay is written in a natural number, and if the delay is “3”, information will be delayed for three unit times.

The accuracy can be implemented by not transforming the request from the interaction layer to agents with a probability to be defined. In the delay, in order to lead the anticipated effect, the timing of processing in the connection bridge needs to be defined in Connection Description. This is defined according to the characteristic of an interaction as follows.

(a) to delay information transmission to an agent for defined time.

(b) to delay information transmission of an executed result from an agent for defined time.

Table 5.1 shows examples of effect of this method and the timing defined in Connection Description to each interaction classified in section 5.3.1.

Users have to note that the effect created by Connection Control continues during whole simulation. For example, about restriction of ability from environment, interference always received from environment can be expressed and the one received from a specific situation (ex. communication failure when the distance is very far) cannot be expressed. Such a situation should be expressed as a model in the simulator (in the agent layer). Next, this method is not enough to express all kinds of restrictions because this controls information before or after an agent executes the request. For example, in the fire fighter domain, when firepower has a power gauge, it is difficult to express the situation that the power is reduced to the half even in failing to extinguish in accuracy and the power is gradually reduced to 0 in double times as usual.

5.3.3 Implementation of Connection Control

To realize the Connection Control, the command must not include several successive interactions. If it does, (example; doing *B* after *A*) the infor-

Table 5.1: Effect of connection control and examples of describing Connection Description

	Communication with other agents	Observation from environment	Action to environment
Effect on agent's ability	Equivalent to fine-tuning the information transmission ability	Equivalent to fine-tuning the ability to collect information from the environment	Equivalent to fine-tuning the ability to act toward the environment
Specification by parameter	"Accuracy" as the failure rate of message transmission, and "Delay" as the time being spent more than usual to transmit the information	"Accuracy" as the success rate of collecting information from the environment, and "Delay" as the time being spent more than usual to collect the information	"Accuracy" as the success rate of acting toward the environment, and "Delay" as the time being spent more than usual to complete the action
Example of effects	1. Accuracy; case of conversation interrupted by noise. 2. Delay; case when more time is spent to transmit the conversation completely.	Case when someone is figuring out about the object faraway. 1. Accuracy; occasional failure of collecting information. 2. Delay; condition of constantly receiving past information because of time-lag.	Failing to achieve or losing more time to take an action which could be instantly done by a more talented person.
Timing specification of delay by Connection Description for the example.	First, the communication must be begun and the similar effect is provided by delaying the return of action results to the interpreter. Accordingly, both (b) is true for information sender and receiver.	First, the information must be obtained at present time and the similar effect is provided by delaying the return of action results to the interpreter. Accordingly, (b) is true.	It is necessary to stop the transmission before the action because action toward the environment will change the conditions of the environment. Accordingly, (A) is true.

mation controlling function will not be able to figure out the timing for the successive interactions to start and therefore cannot realize the delay of which the user intends to happen. In the example, the information controlling function sends "doing B after A" to the agent at once and therefore it can only give the delay of A and cannot give the delay of B. For solving this problem, the information controlling function must separate to a single command even if it receives successive interactions from the interpreter. (e.g. "doing B after A" must be separated to A and then B). This separation can be realized in either of the information controlling functions shown in Figure 5.3.

If the simulator works in discrete time, this time marching must be grasped by the Connection Control Module to process a delay. Therefore, Connection Description should be read by the information controlling function situated in the side of the simulator.

5.4 Connecting *Q* and CORMAS

To put the method shown in the previous section into practical use, we actually constructed the interaction layer using Q which provides a description for interactions and its interpreter on a multiagent simulator CORMAS. In this section, we will show each feature of Q and CORMAS then the implementation of their connection.

5.4.1 Scenario Description Language *Q*

Q[2] [Ishida 02b] is suitable for describing complex social interactions and has been applied to social psychological simulations in a virtual space [Nakanishi 03] such as evacuation simulations [Ishida 02a]. Q can describe interaction scenarios between (legacy) agents (See Table 5.2). This feature makes it possible to control the scenario execution of a large number of agents by attaching Q to already existing agent systems. The computational

[2]http://www.lab7.kuis.kyoto-u.ac.jp/Q/index.htm

Table 5.2: Q and CORMAS

	Q	CORMAS
Goal	• A language for describing interaction scenarios among (legacy) agents. • A large number of agents can be controlled by attaching Q to already existing agent systems.	• System for simulation whose models of coordination modes between individuals and groups who jointly exploit the resources. • The changes of natural environment and the observations or actions of agents are defined.
Computational Model	• Agent scenarios by extended finite state automaton	• Environmental change by cellular automaton
Description	• Scenarios are described, using sensing functions (cues) and acting functions (actions). • Interaction Pattern Card (IPC) enables non-computer professionals to create scenarios. • Participatory simulation is easily realized by replacing a part of software agents by human controlled avatars.	• At every step (unit time), the simulation is executed by describing functions of each cell and agent. • Modeling tools (space, agent, communication between agents), a management tool (simulation) and visualization tools (communication between agents, statistical information) are available. • It is optionally possible to import the spatial map from GIS (Geographic Information Systems) like MapInfo, ArcView.
System	• Event driven. An event triggered by a interaction leads the corresponding process to be executed. • The Q interpreter has no responsibility on time management.	• Time driven. At every step (a unit time), the diffusion between neighboring agents is calculated. • Time management is done by allocating appropriate time length to each step.
Application	• Evacuation simulation • Social psychological simulation	• Resource management of water, wood and pasture • Multiple uses of land and resources
Implementation Language	• Scheme	• Smalltalk

model behind a *Q* scenario is an extended finite state automaton, which is commonly used for describing communication protocols. By using *Q*, users can directly create scenario descriptions from extended finite state automata.

In *Q* scenarios, we can use sensing functions (cues) and acting functions (actions) provided by already existing agent systems. Scenarios are interpreted by *Q*, while cues and actions are executed by the legacy agent systems. The mapping between cues/actions in the *Q* scenarios and those in agent systems is given by the *Q* Agent Adapter. Moreover, *Q* provides an end user language called Interaction Pattern Card (IPC). Since IPC is domain dependent and implemented using Excel, it enables non-computer professionals to create scenarios easily. Scenarios described in IPC are translated into *Q* by the IPC translator.

It is possible to replace *Q* controlled software agents by user controlled avatars, i.e. participatory simulation is easily realized by replacing some of the agents by humans. Simulations are carried out in an event driven fashion. The *Q* interpreter has no responsibility for time management: time is controlled by the agent systems that execute the bodies of cues and actions.

Q is an extension of Scheme, a Lisp programming language dialect. Concurrent execution of scenarios of multiple agents is realized easily by using Scheme's continuation for controlling process switching.

5.4.2 CORMAS

The goal of CORMAS[3] (COmmon pool Resources and Multi-Agents Systems) is to build simulation models of coordination modes between individuals and groups who jointly exploit the (same) resources [Bousquet 98]. In CORMAS, users can define the diffusion of environmental changes, and agents' observations/actions with regard to the environment (see Table 1). The computational model behind CORMAS simulations is a cellular automaton. A natural environment is modeled as a two dimensional mesh, and the diffusion between neighboring cells is calculated at each unit time (called a step). Modeling tools for space, agent and communication, a

[3]http://cormas.cirad.fr/indexeng.htm

simulation management tool, and visualization tools are available. It is possible to import map information from GIS (Geographical Information Systems) like MapInfo[4] and ArcView[5]. CORMAS has been applied to renewable resource management of water, wood and pasture [Perez 02], natural resources marketing systems, and multiple uses of land and resources [Bah 03].

The CORMAS system is time-driven. At every step, the diffusion from neighboring cells is calculated, and agents' observations/actions are performed. Time management is done by allocating the appropriate step duration. CORMAS provides two types of simulations. In synchronous simulations, changes of all entities (cells, agents) are committed at the end of each step. In asynchronous simulations, on the other hand, the changes are immediately reflected in ongoing calculations. Users can construct a model of agents and natural environments by using prepared Smalltalk classes. Users can add new classes as necessary.

5.4.3 Implementation

Q scenarios are described for controlling agents. *Q* provides sensing (Cue) and acting (Action) function for describing the three kinds of interaction mentioned in section 5.3.1. Communication with the other agents is described in two parts: receiving of messages from other agents (Cue) and sending of messages to other agents (Action). In the same way, observation of environment is described in Cue and action to environment is described in Action. *Q* only provides language functions and a scenario interpreter and therefore agents in CORMAS execute the actual observation/action in the simulator. The instance of Cues and Actions in *Q* scenario needs to be defined in CORMAS.

The information transforming function in Figure 5.3 is implemented in shared files. This is one of general approaches to connect different systems as well as TCP/IP, CORBA etc. In our case, the *Q* interpreter is executed

[4]http://www.mapinfo.com
[5]http://www.esri.com/software/arcview/

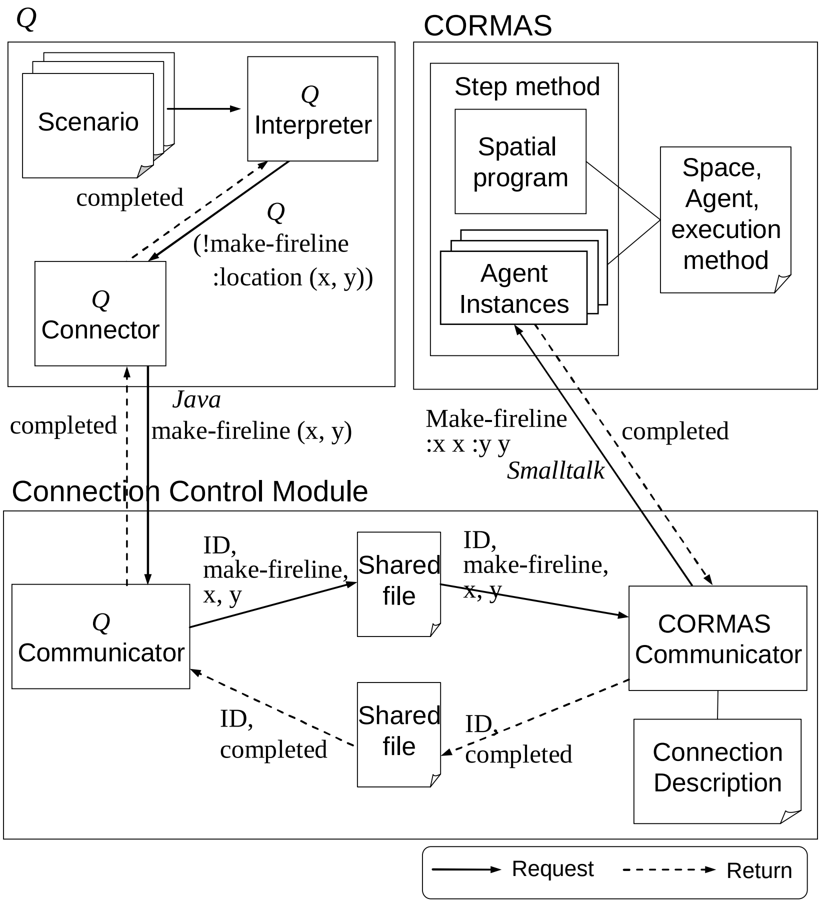


Figure 5.5: Implementation of CORMAS/*Q*

on a Java virtual machine[6] and CORMAS on a Smalltalk virtual machine, which means that both systems can be run on various platforms. Using shared files allows us to retain this benefit and make the integrated system platform free.

For handling the shared files, the Q module/ CORMAS module is extended with the Q Communicator/CORMAS Communicator; they are implemented in Java and Smalltalk, respectively. The format of the shared files is fixed in advance. Q Communicator has two roles. One is to write execution requests of Cue/Action to the shared file. The other is to receive the execution results and return them to Q. The CORMAS Communicator also has two roles. One is to receive execution requests from the Q module and call the corresponding agent instance. The other is to write the execution results to the shared file.

The information controlling function is implemented as follows. Q interpreter thorough Q Connector outputs a set of interactions, conforming the condition mentioned in section 5.3.3[7]. For Connection Control, Connection Description is equipped with CORMAS Communicator. The flow of information is controlled based on the Connection Description when CORMAS Communicator calls the agent method. CORMAS Communicator needs to grasp the time marching in CORMAS in order to realize “delay” because CORMAS carries out the simulation in discrete time.

Figure 5.5 describes an example which action “make-fireline” is executed.

5.5 Example: Fire Fighter Domain

5.5.1 Problem

To discuss our approach, we ran a simulation using the model of a forest fire in Yellowstone National Park developed by the Phoenix Project at the University of Massachusetts Amherst in 1989 [Cohen 89].

[6] Q is the Kawa version.

[7] Precisely saying, the format of output can be configured in the Q Connector

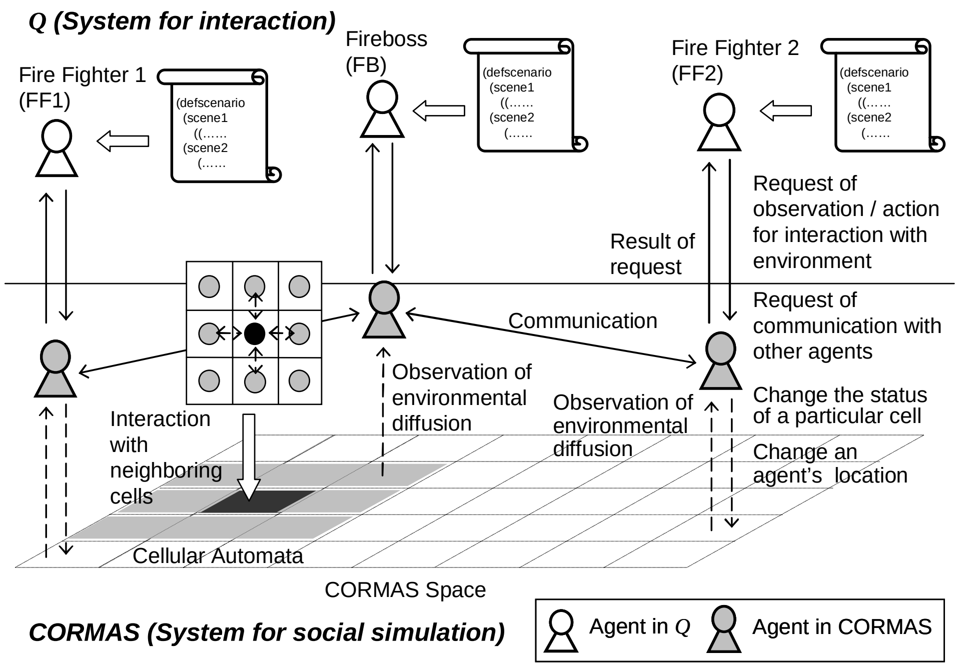


Figure 5.6: Fire Fighter simulation by CORMAS/*Q*

In this problem, two types of agents exist. One is a fireboss who gives a direction based on the overall environmental information. The other is a fire fighter who follows the fireboss's direction and moves in the environment. The fire fighter is a bulldozer and builds a line around continuously burning fires to prevent the fires from spreading.

In the environment, each cell has a type such as a river, a plain, a road, wood, and fire. Fires spread in irregular shapes and at variable rates, determined by ground cover, elevation, moisture content, wind speed and direction, and natural boundaries.

This simulation starts when the fireboss is alerted to a new fire and dispatches two fire fighters to a rendezvous point which is determined by a calculation based on to the location of the fire and wind speed/direction. Next, the fireboss calculates a route for the fire fighters and instructs them

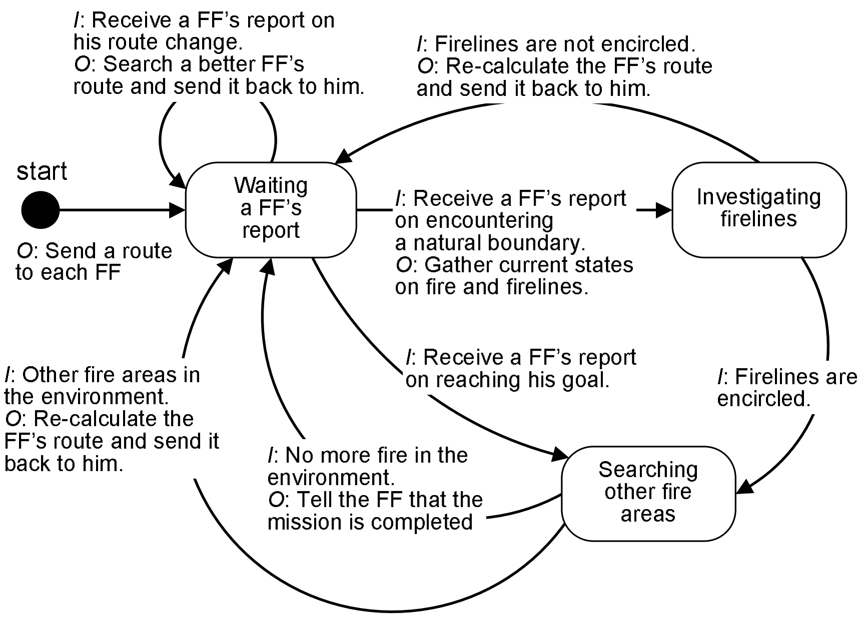


Figure 5.7: The interaction model of the fireboss

to follow the given path and encircle the fire. They are given clockwise and counterclockwise encircling directions. When they find a fire on their route, they can change their route by themselves. If the fire threatens them due to an abrupt environmental change, they move away from the fire. When such a route change is reported to the fireboss, he sends back a better route plan calculated from the location of the fire fighter and overall environmental information. The fire fighter changes his own route plan as suggested by the fireboss when the difference between his own plan and the fireboss's plan is large. When a fire fighter encounters a natural boundary like a river or a fire line created by the other fire fighter, he asks the fireboss for the next action. If a fire leaps outside the fire circle, the fireboss gives the fire fighters a new route plan. When the fire is enclosed and flying sparks are insignificant, the fireboss concludes that the mission is finished and stands down the fire fighters.

```
(defscenario fireboss
   (&pattern ($FF_name "") ($FF_info "") ($env_info "") ($route ""))
   (Waiting-FF-report
     ((?route_change_report :from $FF_name :info $FF_info)
      (!get_env_info :info $env_info)
      (!calculate_route :info $FF_info :info $env_info :result $route)
      (!send_route :to $FF_name :result $route)
      (go Waiting-FF-report))
     ((?natural_boundary_report :from $FF_name :info $FF_info)
      (!get_env_info :info $env_info) (go Investigating-firelines))
     ((?goal_arrival_report :from $FF_name :info $FF_info)
      (go Searching-other-fire)))
   (Investigating-firelines
     ((?firelines_encircled :info $env_info)
      (go Searching-other-fire))
     ((?firelines_not_encircled :info $env_info)
      (!calculate_route :info $FF_info  :info $env_info :result $route)
      (!send_route :to $FF_name :result $route)
      (go Waiting-FF-report)))
   (Searching-other-fire
     ((?no_more_fire :info $env_info)
      (!send_mission-complete :to $FF_name)
      (go Watching-FF-report))
     ((?other_fire :info $env_info)
      (!calculate_route :info $FF_info  :info $env_info :result $route)
      (!send_route :to $FF_name :result $route)
      (go Waiting-FF-report))))
```

Figure 5.8: The scenario of the fireboss

In this way, the fire is controlled by the interaction of the fireboss and the fire fighters.

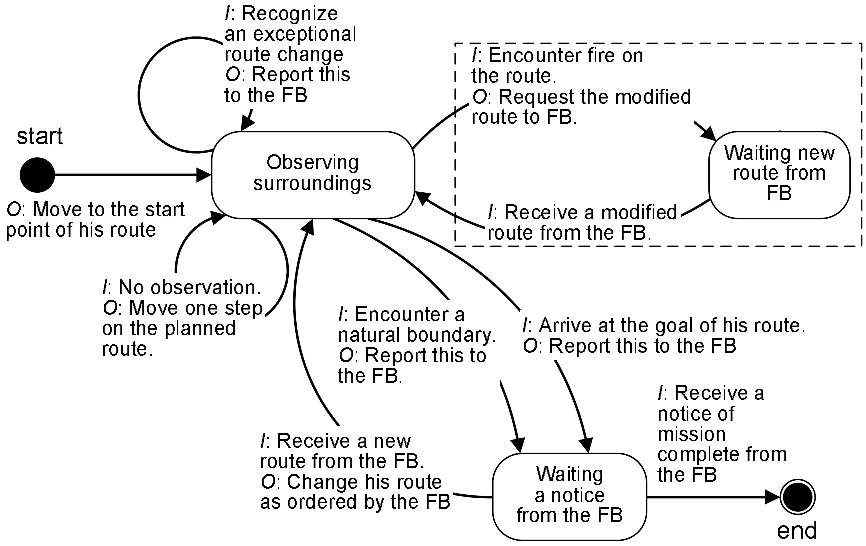
5.5.2 Models

Figure 5.6 illustrates the implementation of this problem in CORMAS/*Q*. In CORMAS, the environment is expressed in a cellular automaton, and each cell has ground information (e.g. fire, wood, water etc.) and weather information of wind speed and the direction.

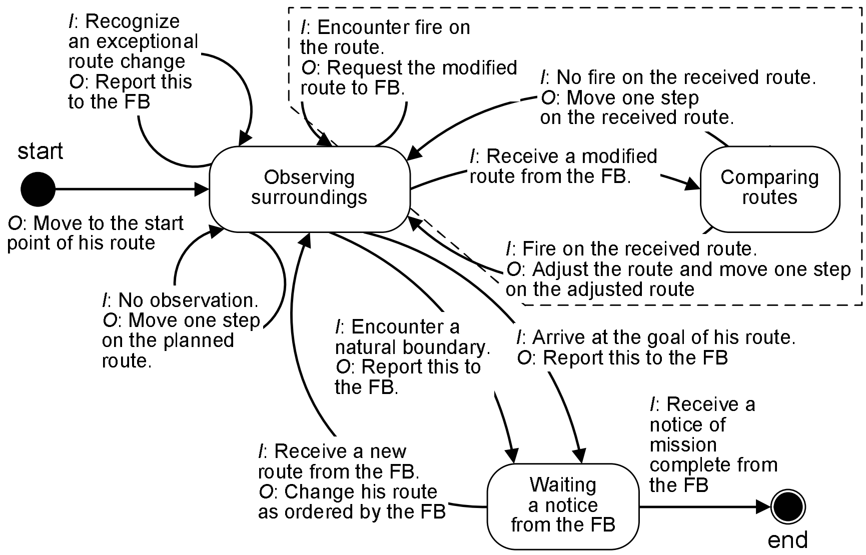
We implemented a simulation in which fires spread minute by minute in forest with a river running through it. At the beginning of this simulation, there is a small fire in the upper area of the forest and fire is spread by the north wind. The fire spreads according to the cellular automaton model shown in Figure 5.10. In this model, each cell calculates its next state at every step, referring to the eight neighboring cells. This rule is followed by all cells in the environment. The rule of fire spread is described as follows: First, if the state of the central cell is 'a tree', the next condition will be applied. Next, if the state of the north cell (i, j+1) is 'fire', the state of the central cell is changed to 'fire' at the probability of 30%. The probability is 10% at the north-east cell (i+1, j+1) and the north-west cell (i-1, j+1).

On the other hand, the fireboss agent and the fire fighter agents are placed in both *Q* and CORMAS; *Q* controls the agents in CORMAS. There are two types of interaction in the *Q* scenario. One is social interaction between the agents. The other is environmental interaction between the fire fighters and the environment.

The behavioral plans of the fire fighter and the fireboss are expressed in a state transition diagram. Figure 5.7 is the model of the fireboss and Figure 5.9 is the model of the fire fighter. You may understand the correspondence between the problem explanation in section 5.5.1 and the state transition diagram of these two figures. The *Q* scenario is directly coded from the state transition diagram (a scenario shown in Figure 5.8 corresponds to a diagram shown in Figure 5.7).

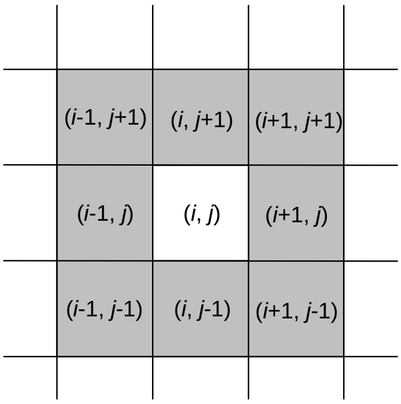


(a) The initial interaction model of the fire fighter



(b) The modified interaction model of Fire Fighter

Figure 5.9: The interaction models of the fire fighter




```
if State(i, j) = #tree
then
  if State(i, j+1) = #fire
     and
     Random(0.3) = 1
  then State(i, j)-> #fire
  else if State(i+1, j+1) = #fire
         and
         Random(0.1) = 1
  then State(i, j)-> #fire
  else if State(i-1, j+1) = #fire
         and
         Random(0.1) = 1
  then State(i, j)-> #fire
```

Figure 5.10: Environmental diffusion model by cellular automaton in CORMAS

5.5.3 Results

Figure 5.11 indicates the screenshot of the simulation. There is a spread of forest and a river running through it. First we set a small fire in the upper area of the forest and then let it spread by the north wind. The fire fighters are placed on the north side of the fire (see Figure 5.11 (a)).

We could modify the interaction model, using the Connection Control (in Figure 5.9, (a) was modified to (b). The area framed by dotted lines was modified.)

First, the condition in which the fireboss's message is interrupted by noise from the environment is described by the Connection Description. This was realized by imposing 3 steps of delay to all transmission from the fireboss to the fire fighters. The timing of this delay mentioned in section 5.3.2 was set to (b) for both the fireboss and the fire fighter. As a result, the fire fighter's moving was severely disturbed and ended to let the fire spread over to most of the lower part of the environment (see Figure 5.9 (b)). Then, re-tracing the fire fighter's behavior in the simulation, we found out the cause of this result to be the following. Because of the transmission delay, by the time the fire fighter completely recognizes the modified route from the fireboss, the fire has spread further and the modified route is already on fire. Accordingly the fire fighter must require another route. Such case was frequently seen in the simulation.

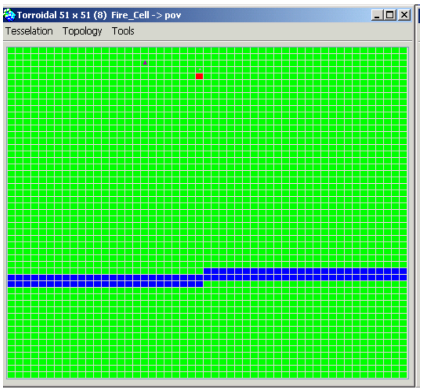
Next, we made a model change for the fire fighter so that when the modified route is already on fire, it can adjust its route according to the surrounding available information (see Figure 5.9 (b)). By this model change, the fire fighter managed to adjust its route by itself and saved the number of times having to transmit with the fireboss. As a result, although the fire spread more further compared to the case without delay, the fire was held within the fireline and the river. (see Figure 5.11 (c)).

5.5.4 Discussion

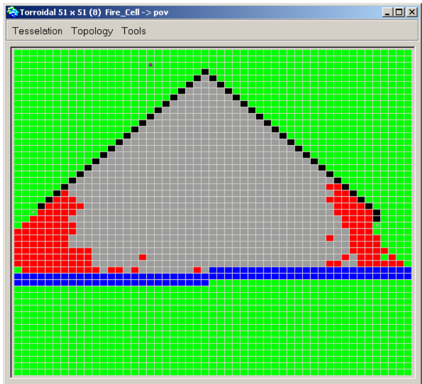
This Fire Fighter simulation verifies that according to our proposed method, a simulator with the architecture and the system framework has worked out without any problem.

Further more, we believe that our proposed method will be effective in a domain where environment and people are intricately-involved and therefore requiring a complicating interaction model, such as the Fire Fighter simulation. *Q*, using state transition model as a computational model, gives description of parallel observations and their corresponding action, which is suitable for this domain where parallel observations from the environment and the other agents are occurred. It would be difficult for CORMAS to archive the description and readability given by *Q* scenario because it is necessary to squeeze the same mechanism as *Q* interpreter into the structure of CORMAS.

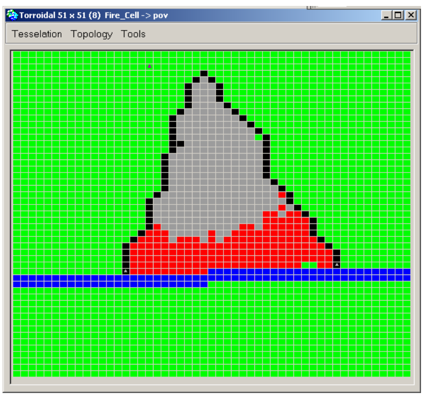
Moreover, we consider the concrete example indicates that "Connection Control" is effective for designing more realistic interaction models. It is difficult to realize Connection Control without the proposed connecting method because it would be necessary to reconstruct the inside of a simulator. In other words, this method makes it easy to realize a platform where users adjust ability of agents and restriction from the environment. How-



(a) Initial state



(b) The simulation result with the initial interaction model



(c) The simulation result with the modified interaction model

Figure 5.11: Screenshots of the simulation

ever, we still need to consider the effect of this method in a simulation with domain experts. At the same time, we also need to refine this method in the other domains.

5.6 Related Works

Our approach is a kind of combination of different systems. In this view, the related works are as follows. Robocup project proposed a framework where various agents participate in the soccer simulation environment [Noda 03]. The same kind of framework was also adopted in U-Mart where agents and users participate in a virtual market [Kita 03]. As the extension of Robocup soccer framework, Robocup rescue [Takahashi 02] proposes a framework where agents and various kind of disaster simulators can join together. High level architecture (HLA) [Kuhl 00] is a very generic, language-independent architecture specification for an infrastructure that integrates simulation components ("federates") into robust and controllable aggregates ("federations"). Robocup and U-Mart mainly intend to give an environment where various agents created by different researchers and various users can join. Robocup rescue and HLA intend to re-use and configure the existing systems. However, this research mainly intends to give an environment for design and modeling of interactions, and the related works do not have this perspective.

On the other hand, there are several languages for describing coordination among agents. For example, AgenTalk [Kuwabara 95], Cool [Kolb 95] and COOL [Barbuceanu 95] are well-known. We can select a suitable language according to the purpose, using our proposed method.

5.7 Summary

There exists many platforms to support domain researchers to develop and analyze multiagent simulation, but they do not always give environment to design and evaluate interactions. In this chapter, we proposed a method to construct a platform for designing interactions with a legacy simulator.

First, to provide a computational model suitable for interactions, an interaction layer is constructed and connected from outside of a legacy simulator. We showed the architecture and the system modules. Especially, two functions in the connection area were shown: information controlling function, which realizes the Connection Control, and information transforming function, which realized communication between different two systems. Next, to configure the agents interacting ability and refine interaction model in more realistic situation, a method for controlling the flow of information in the connection area is provided (Connection Control). Two types of operations to the information were defined: accuracy and delay. Especially in delay, two types timing of operation are needed. We also showed the effect of these operations in three kinds of interactions in multiagent simulation and examples of setting a timing of delay.

As a concrete example, we realized an interaction layer by *Q* which is a scenario description language and connected it to CORMAS, which is often used in the field of the participatory method. Finally, we discussed the capability of our method and showed a process to refine an interaction model through a concrete implementation in the Fire-Fighter domain [Cohen 89].

Chapter 6

Modeling Interactions with Participatory Simulation

In this chapter, we will discuss the issues and the solutions in using participatory simulation to model interactions and explain the processes involved with an example: negotiation of seed suppliers in the northeast of Thailand. The negotiation model is represented in a state automaton and we implemented our participatory simulation in CORMAS/Q described in chapter 5 [Torii 06b, Torii 06c].

6.1 Introduction

Role playing game (RPG) is useful to promote communication between stakeholders and the domain experts and effective to extract stakeholders' thinking, using a board which represents environment surrounding stakeholders. However, RPG is not enough to model interactions (e.g. negotiation in economic activities) because interactions are a series of action and it is difficult to express such a dynamic process in boards or papers. Moreover, RPG, where many stakeholders are gathered together, is expensive to run and is difficult to concentrate a specific stakeholder and deeply understand him/her.

Here, we will introduce a participatory simulation for the following two purposes: 1) It is relatively easy to express the process of interactions on a computer. Stakeholders can grasp the effect of their decision on the simulation world. 2) Computer agents can play instead of absent stakeholders then domain experts can concentrate the surveyed stakeholders and deeply understand them.

In this chapter, we will consider the following two issues:

(i) *Construction method of a participatory simulation system*

Participatory simulation is such a complicated environment as where stakeholders as avatars and agents participate. What is important here is to create an environment that supports the domain expert in designing agent-avatar interaction so as to elucidate the behavior of the stakeholder.

(ii) *Process for modeling interactions*

It is not appropriate for stakeholders to play participatory simulation from the beginning of the modeling process. We need to consider a process, applying the notion of the participatory method.

Section 6.4 explains the related works using participatory simulation. Section 6.3 shows our example (seed supply chain in the northeast Thailand [Vejpas 04]) to be applied with this method. Section 6.4 describes a method to construct a system for participatory simulation. Section 6.5 shows our modeling process with the real example and our approach is discussed in section 6.6. Finally, we summarize this chapter in section 6.7.

6.2 Related Works

There are two researches to use participatory simulation in agent modeling. 1) One applied machine learning to the logs of participatory simulations for developing diverse agent models in a virtual training system [Murakami 05]. 2) The other allowed agents to learn effective negotiation strategies for the participatory simulations of a virtual market [Kita 03]. The former targeted behavior models that reflect reality and focused on developing diverse agent models for virtual training systems with increased realism. This standpoint differs from ours which aims to deeply understand human behavior for designing social systems. The latter does not aim to deeply understand real human behavior, either.

We know of two studies that attempt to deeply understand actual human behavior. 1) One developed a web forum of agents and stakeholders for holding a continuous RPG free from the constraints of time and place [Gilbert 02]. 2) The other used participatory simulations in which only actual users participated so as to eliminate a drawback with the RPG approach in which roles of participants are fixed and emergence of new roles cannot be handled [Guyot 05]. In the first study, participatory simulation is intended to create a place for continuous meeting of stakeholders and this differs from our use of participatory simulation: to exploit behavioral models of stakeholders. The second also mentions design processes, but they are not for constructing an environment where agents and avatar participate.

6.3 Seed Supply Chain in the Northeast of Thailand

In this study, we aim to recognize the negotiation model of seed suppliers in the northeast of Thailand. The negotiation model here is the process of how a supplier deals with other traders (e.g. the access order, the amount of seed to buy/sell etc.). In Section 6.5, we focused only on the Seed Center (SC), which plays a main important role in the organization. Therefore, in the participatory simulation, the staffs of SC took part in as the only avatar, and the rest of the suppliers are all computer agents. SC is a wholesale, selling seed to many other suppliers.

Figure 6.1 shows the structure of the organization we are going to address here. It is a typical organization structure of suppliers in the northeast of Thailand, scaled down to a smaller size. The suppliers are underlined by their actual name (Table 3.1 in chapter 3 explained the role of each sup-

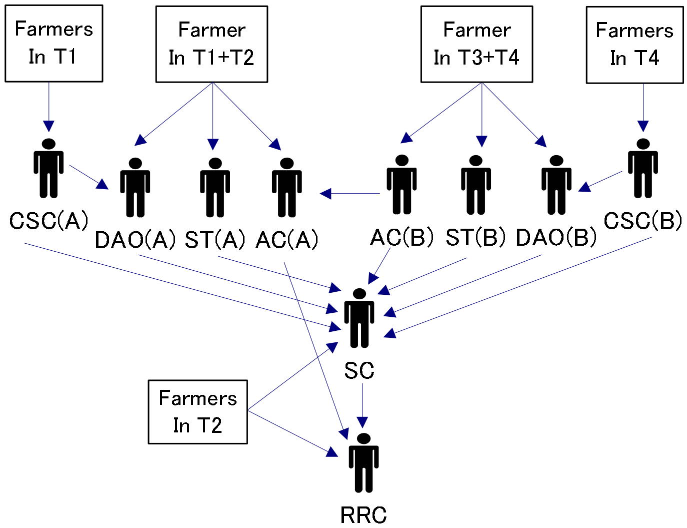


Figure 6.1: Organization of seed suppliers in the northeastern Thailand: the suppliers have links to the other suppliers and farmers. "T" means Tambon which is a name of unit expressing a small region in Thailand

plier). RRC (Rice Research Center) and SC (Seed Center) operate in the whole Province. AC (Agricultural Cooperative), ST (Seed Trader) and DAO (Distinct Agricultural Office) operate within a Amphoe (which consists the Province) and CSC (Community rice Seed Center) operates inside of a Tambon (which consists a Amphoe). We assume 2 types of Amphoes (A and B) here. The letter in the brackets adscript with AC, ST, DAO and CSC indicate which Amphoe they belong to. Arrows in the figure show the channels of trading. (For example, SC is provided only from RRC.)

6.4 Approaches

6.4.1 Agent Control by Interaction Description

In participatory simulation, interactions among agents and avatars are very important. We use system architecture shown in Figure 6.2 where an inter-

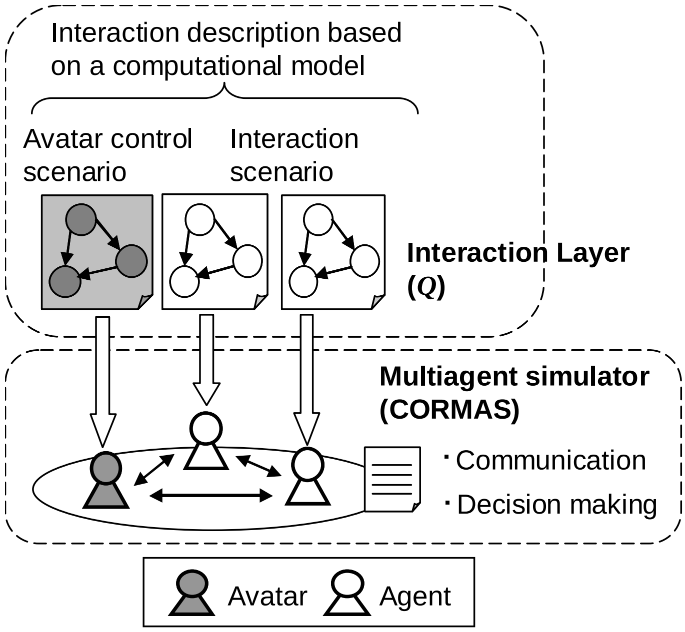


Figure 6.2: Agent control by interaction descriptions

action layer constructed outside the simulation system controls the agents in the system as described in chapter 5. This architecture makes it possible to use interaction descriptions based on a computational model that makes it simple to write descriptions to control agents. Therefore, users can focus on interaction design.

As an actual simulation system based on this architecture, we used the scenario description language *Q* [Ishida 02b] for describing interactions and combined it with the multiagent simulator CORMAS [Bousquet 98] which is often used in the research field of the participatory approach (the system is called CORMAS/*Q* as described in chapter 5).

Table 6.1: Classification of interactin description

Classification	Definition of each interaction description
Interaction protocol	Procedures which should be abided to negotiate with another agent or avatar
Interaction scenario (negotiation model)	Procedures in which access priority of trading or condition for completion of the deal etc. are described within the framework of the interaction protocol
Avatar control scenario	Procedures of UI and avatar control along the interaction protocol

6.4.2 Definition of Interaction Descriptions

For realizing participatory simulations where agents and avatars interact, we define the three kinds of interaction descriptions shown in Table 6.1. These descriptions are designed according to the modeling process shown in the next section. To construct an appropriate system for a participatory simulation which elucidates the behavior of stakeholders, it is necessary that domain experts design these descriptions but they are not the experts for developing such a system so it is important to design through collaboration of domain experts and agent designers (computing professionals). In the architecture above, these interactions (interaction scenario and avatar control scenario) are to be described for a single agent. The negotiation model targeted to acquire is described in the form of the interaction scenario.

The reasons we assume these three descriptions are as follows. First, interaction protocols are very important because they define how user-controlled avatar communicates with computer agents in the simulation world. Second, an interaction scenario is a negotiation model for agents who access the other agents and avatars. Third, an avatar control scenario is an extended scenario of the interaction protocol. This is required in the participatory simulation for negotiation/coordination because users must make decisions according to the procedures regulated by the interaction protocol, and the actions except the protocol should be restricted. At the same time, it should regulate behavior of an avatar after the user's decision making done through user interfaces (UI). Therefore, the avatar control scenario is described in four elements: 1) control of UI (display etc.), 2) observation of input by a user through UI, 3) sensing from other agents, and 4) acting to other agents which is triggered by input to the UI by a user.

For an actual example, the three kinds of interactions were described in *Q* scenarios. The computational model behind a *Q* scenario is an extended finite state automaton, which is commonly used for describing communication protocols. By using *Q*, users can directly create scenario descriptions from extended finite state automata. In *Q* scenarios, we can use sensing functions (cues) and action functions (actions) provided by CORMAS. Scenarios are interpreted by *Q*, while cues and actions are executed by CORMAS. Cues/actions become the interface between domain experts and computer scientists, which enables collaboration in the design of interactions [Ishida 02b, Murakami 03].

6.5 Modeling Process with a Real Example

In this section, we will see a process to model negotiation, apply this process to the real example mentioned in section 6.3 and show the results. In this example, we aimed to acquire a negotiation model of SC (Seed Center) who is one of the important suppliers in the northeast of Thailand. The simulation system is CORMAS/*Q* described in the previous section.

6.5.1 Overview of Process

The overview of a modeling process is as follows. Each in four steps corresponds to the number in Figure 6.3. This process is operated by three roles: stakeholders, domain experts, system developers.

STEP1: Survey and Interview to Stakeholders

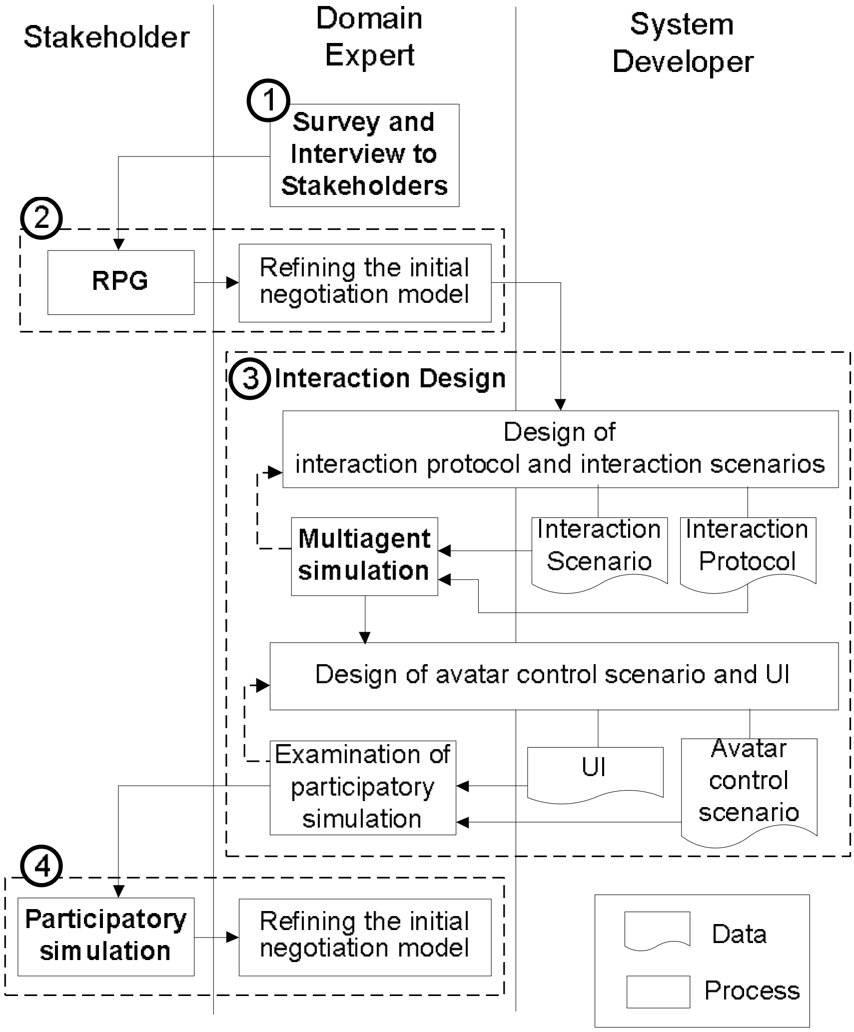


Figure 6.3: Modeling process for interactions

Domain experts create an initial model from relevant literature and interviews to stakeholders.

STEP2: Role Playing Game

RPG based on STEP1 is held with stakeholders. Through RPG, domain experts make much account of communication with stakeholders and are mainly interested in understanding problems of stakeholders or among stakeholders. The initial model is improved by the knowledge acquired through RPG.

STEP3: Interaction Design

The three kinds of interactions (interaction protocol, interaction scenario, avatar control scenario) are designed through collaboration of domain experts and system developers based on observation in RPG. First, interaction protocols and interaction scenarios of each agent are designed and tested through several runs of multiagent simulation. Next, avatar control scenarios and UI are designed, and a system for participatory simulation is implemented and adequately tested.

STEP4: Participatory Simulation

Participatory simulation constructed in the previous step is held with specific stakeholders whose negotiation model need to be deeply understood. Then, the domain experts refine the negotiation model by the knowledge acquired through communication with the stakeholders and the operation logs of the stakeholders.

The stakeholders, villagers in southeastern Asia and Africa, are unfamiliar with the research process and sometimes unfamiliar with computer systems. To this end, we took RPG into this process instead of holding a participatory simulation from the beginning. RPG has been a good place for communication between domain experts and stakeholders in researches of the participatory method. We considered that such an opportunity enables stakeholders to comfortably join the participatory simulation.

The following process describes the details of each step with the results that the process was applied for modeling negotiation in the seed supply chain mentioned in section 6.3.

6.5.2 STEP1: Survey and Interview to Stakeholders

First, domain experts investigate literature and interview stakeholders. This step is important for deciding how to organize RPG and participatory simulation.

In our example, domain experts investigated several reports about the northeast of Thailand and interviewed the stakeholders. The problems which exist as a whole were identified and the initial model was not concrete but very abstract which just specified a contact list of each supplier (see Figure 6.1) [Vejpas 04].

6.5.3 STEP2: Role Playing Game

RPG based on STEP1 is designed and held with stakeholders. Through RPG, the actual behavior of stakeholders is observed and the reason behind the behavior is exposed by interviews (debriefing). The initial model is improved by the knowledge acquired through RPG. During RPG, it is also important to consider how to design interaction protocols to be needed in the next step.

In our example, RPG was held with the seed suppliers (also, representatives of the government organization) mentioned in section 6.3. Figure 6.4 shows the negotiation model of SC from the results through RPG. The refined model of SC in our example is that rice seed is supplied according to a priority order (DAO, CSC, ST and AC).

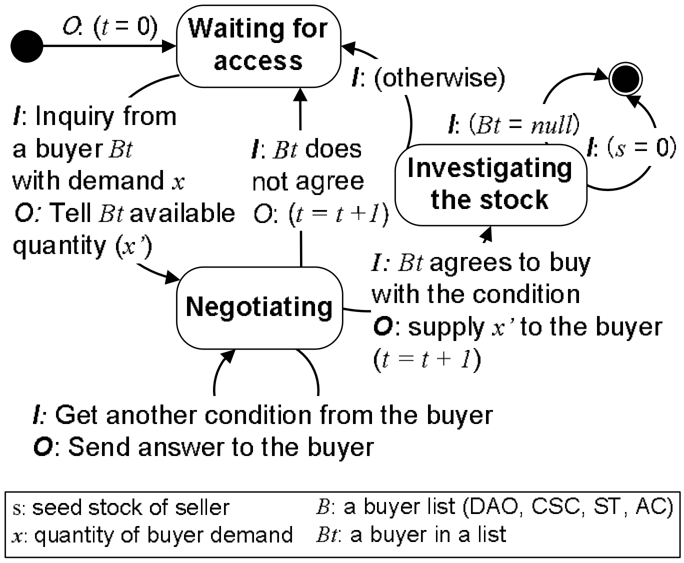


Figure 6.4: The negotiation model of Seed Center after the RPG, which shows supply procedure (DAO, CSC, ST, AC) by the order in “buyer list.

6.5.4 STEP3: Interaction Design

The three kinds of interaction are designed in the following procedures.

i) Design of interaction protocols Interaction protocols are to be defined. This defines procedures for avatars and agents to adhere in participatory simulation. This should be designed to elucidate negotiation models of stakeholders, referred from the result of RPG.

ii) Description of interaction scenarios Interaction scenarios of agents acting as stakeholders who do not attend the participatory simulation are designed based on the interaction protocols. What is important here is that several kinds of interaction scenarios are created for each agent to observer reaction of a stakeholder in participatory simulation.

iii) Multiagent simulation System developers implement agents who play according to the interaction scenarios and the environment. The domain experts run the simulation and test behavior of the agents. If necessary, the protocols and scenarios are modified.

iv) Description of avatar control scenarios Avatar control scenarios and UIs are designed based on interaction protocol.

v) Test of participatory simulation Avatar control scenario is combined to the system and is tested over several runs. If necessary, the interaction scenarios designed heretofore are modified.

The process i), ii) and iv) are done through cooperation of domain experts and system developers.

In our example, interaction protocols (an example for sellers is shown in Figure 6.5) and interaction scenarios for each agent (Figure 6.4 and Figure 6.8 are the examples) were designed by a state transition model which is a computational model of Q. Negotiation in our case was proceeded in the relation between one seller and one buyer then the interaction protocols were described in such a manner (on the other hand, the interaction scenarios were described in the relation between one to many).

To observe reactions of the stakeholders in several angles, three types of interaction scenarios are prepared: a) access in a fixed order, b) access in a cheap price order and c) access in a short distance order. a) is the one which follows the real activity. Next, the domain expert and the system developer defined cues/actions of Q and the system developer implemented the corresponding behavior of the cues/actions in CORMAS (this part of design process are described in []Murakami:03:aamas). Finally, through multiagent simulation, the domain expert tested all kind of scenarios and adjusted them.

Next, UI and the avatar control scenario were designed. UI was not a single big panel which has all functions for participatory simulation but a small one which has necessary functions in each procedure. They appear or disappear according to the avatar control scenario. In this way, the designer can provide users actions according to an interaction protocol. The avatar

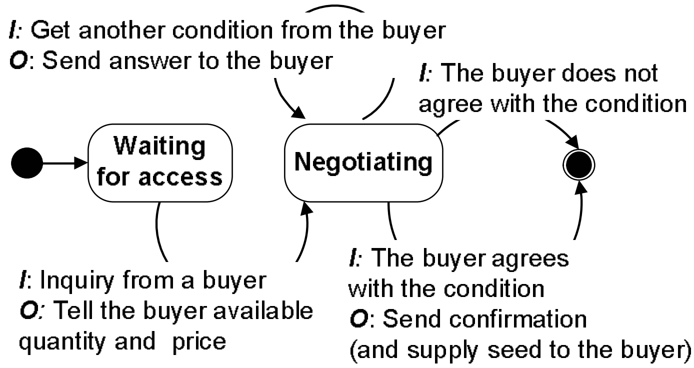


Figure 6.5: An example of interaction protocol (for sellers): Conditions are exchanged between the seller and the buyer in enough times to agree and the negotiation is settled if the buyer agrees/disagrees with the condition.

control scenario was designed in a state transition diagram (the example of SC is shown in Figure 6.5) and translated to *Q* scenario. Then, cues/actions are designed and corresponded to the control of UI or avatar behavior in CORMAS. Finally, the domain expert tested the system of participatory simulation.

6.5.5 STEP4: Participatory Simulation

i) Participatory simulation A participatory simulation with stakeholders whose interaction model the domain experts want to deeply understand is held. The domain experts run several simulations in different interaction scenarios to assign an agent.

ii) Refinement of a negotiation model The domain experts refine the initial interaction model from discussion during the simulation and the log data.

In our example, the participatory simulation was held at Seed Center (SC) No.10 on July 19th 2005 in Ubon Ratchathani which is the capital of

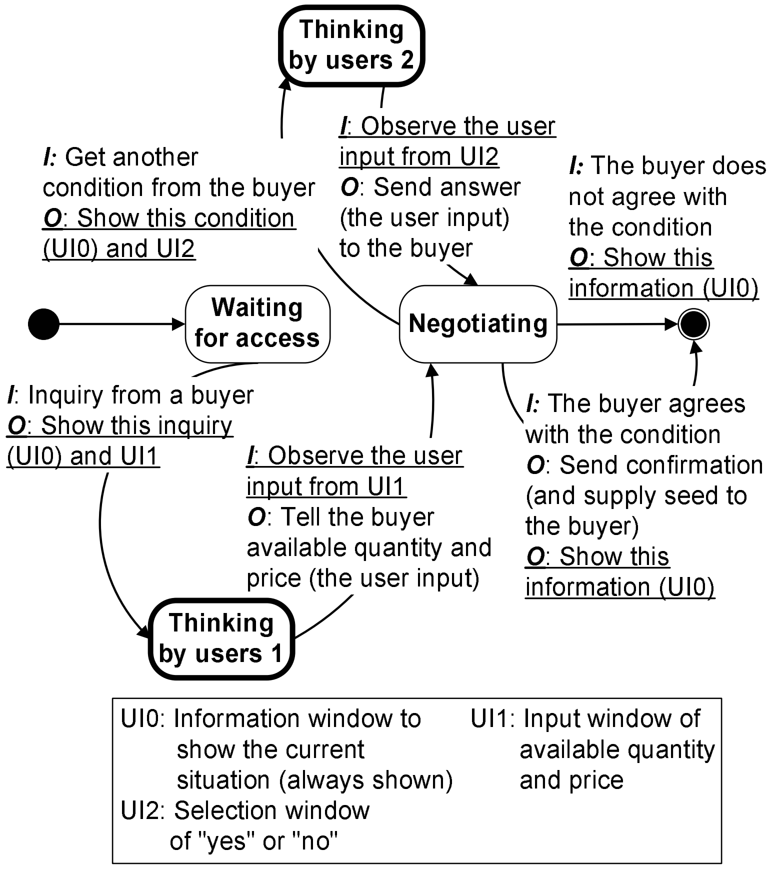


Figure 6.6: An example of avatar control scenario (for Seed Center as an avatar): the interaction protocol shown in Figure 6.5 is extended by putting controls concerning to UI (states surrounded by a bold line and conditions/actions with underlines)

Figure 6.7: A picture of the actual participatory simulation: two in left are stakeholders (staffs in the Seed Center) and a man in upper-right is a researcher of this domain.

the northeast of Thailand. One domain expert (Dr. Chirawat Vejpas) and one system developer (the author) organized this simulation and we asked two staffs of the SC to take part in this simulation. The stakeholders did not speak English and so Dr. Vejpas translated my statement to Thai. After explaining the background of this simulation, we discussed several parameters used in the simulation (e.g. the average amount of the rice production) with them and modified the parameter because this may make influence to their decision. In the simulation, they made each decision making from their discussion (see Figure 6.7). The other stakeholders except SC were represented as agents controlled by the interaction scenarios. The session spent about three hours in the morning (9 a.m. to the noon).

The refining process of the model is as follows. CSC, AC, ST and DAO are the names of the seed supplier. The result after several times of participatory simulation was that CSC was always selected first then the selection of AC, ST and DAO was not fixed. This was observed every time, though we changed the interaction scenario given to the agents in every run of sim-

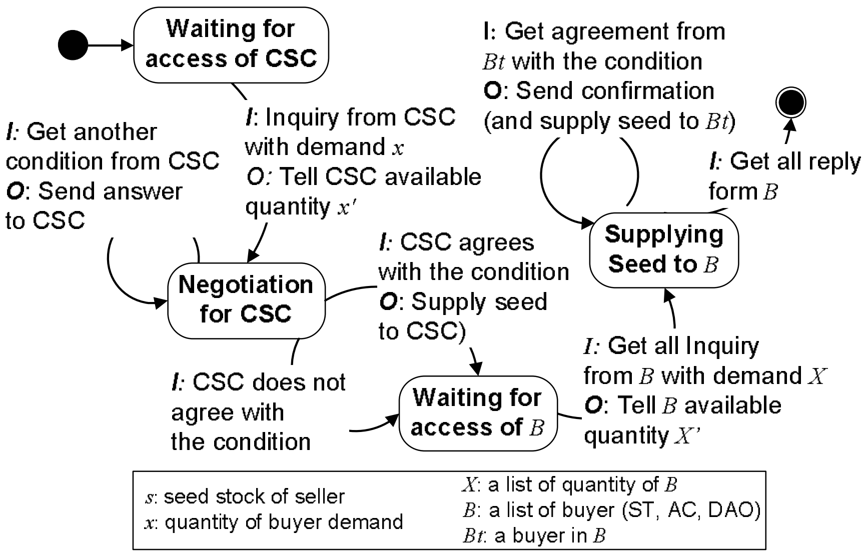


Figure 6.8: The refined negotiation model of Seed Center after the participatory simulation: CSC is first chosen then AC, ST and DAO are equally distributed. If the stock remains, the other buyers will be supplied.

ulation. After the domain expert recognizes this phenomenon, he tried one more around of simulation while he was asking questions to the stakeholder. From the conversation with the stakeholder, two reasons were identified: 1) CSC is a priority for SC because a project decided to support CSC (DAO is a priority before this simulation). 2) The next order of priority is DAO, ST and AC in their policy, but there are no priority and SC equally provides seed to the three suppliers in reality (SC divides the remains in a ratio of their each inquiry). Finally, the domain expert could refine the model as shown in Figure 6.8.

6.6 Discussion

The stakeholders could complete the simulation without any problem. Although they do not usually work with computers and are not familiar with research processes, they did not need much explanations from the domain expert. We consider that the "RPG" helped stimulating the communication between the experts and stakeholders, and therefore led the stakeholders to fully understand the issue addressed in our study. The "Survey and Interview to Stakeholders" placed in the beginning was not only a step for making a direction of the whole study but also an important part of the preparation for the RPG.

We have recognized the following four merits of the participatory simulation. 1) Generally an RPG costs a great amount of work and cost to assemble a large number of stakeholders. However, in this case the participation was limited only to the SC (for about three hours) and saved a lot of cost for both the experts and stakeholders. 2) A computer could represent the timing of making decision and the influence to the simulation world by the decision, and negotiation process which is usually difficult to express in boards and sheets was expressed comprehensively for both the experts and stakeholders. 3) Unlike the RPG, most of the stakeholders were played by the agents and the simulation was automatically processed by the computer program. This enabled us to concentrate on just a few stakeholders. (just the SC for this time.) 4) The experts were able to ask questions to the stakeholders during the simulation and stakeholders could answer them more specifically by recognizing their present condition and result of their decision at real time.

In addition, the architecture we adopted for participatory simulation was favorable to concentrate on the designing of the interaction, the most important part for a simulation. Therefore experts and system developers could well consider the design of interactions. This also led the complicated interaction to be described systematically by clearly defining the three interactions and explaining the process of designing them. (at step 3: interaction design)

In the architecture adopted here, interaction model should be described at the point of view of a single agent. However in some cases of designing interaction protocol, it is more convenient to describe considering a number of agents. Therefore it is a task in the future to provide a more user-friendly tool for designing interactions. However, we consider it is significant in a point of modeling to introduce a mechanism in which agents are controlled by scenarios described in a computational model dedicated to interactions which is important for participatory simulations.

In participatory simulations, the UI handled by the stakeholder is also an important factor. Although our UI was very simple and cost little work to design for this time, research on the design of the UI will be our future work.

6.7 Summary

In this chapter, a modeling method with participatory simulation was established for deep understanding of stakeholders' interactions (e.g. negotiation). We aimed to overcome the drawback of RPG used in the participatory method which is difficult to express a series of process in negotiation. The issues we tackled are as follows:

(i) *Construction of a system for participatory simulation*

The system architecture where the interaction layer controls agents was adopted in order that domain experts can focus on interaction descriptions. Three kinds of interaction descriptions (interaction protocol, interaction scenario and avatar control scenario) were defined and their modeling process was also established.

(ii) *Process for modeling negotiation*

Four modeling steps are established for both stakeholders and domain experts to make the best use of participatory simulation and acquire negotiation models: "Survey and Interview to Stakeholders", "Role Playing Game (RPG)", "Interaction Design", "Participatory Simulation". The point is that RPG is held before participatory simulation because RPG gives a good opportunity for stakeholders to understand background of the domain through communication with the domain experts.

Finally, a participatory simulation was developed on CORMAS/*Q*, and we succeeded in capturing a negotiation model among suppliers of rice seeds. In our experiment, the stakeholders could well understand the negotiation process and the influence of their decision in the simulation world. The domain expert could concentrate to observer one specific stakeholder and ask questions to the stakeholder in any time, which brought a deep understanding of negotiation by the domain expert. This is because computer agents take roles of the other stakeholders in the simulation and the computer program automatically made progress of the simulation.

Chapter 7

Conclusion

7.1 Contributions

This thesis proposed a novel methodology for modeling agents and interactions in multiagent simulation for consensus building among stakeholders. To this end, technologies which support model acquisition and validation are combined to the participatory method by social scientists. Finally, the proposed methodology was evaluated with an actual case study on agricultural economics in the northeast of Thailand. Contributions are summarized as follows:

i. Modeling agents with classification learning

A modeling method wherein a hypothesis of domain experts is validated by a model which classification learning creates from RPG log data was proposed. This method has made it possible to understand internal models of human (stakeholder) from a neutral standpoint independent with the modeler's ability. This method was actually evaluated in modeling farmers' decision making for seed suppliers in the northeast of Thailand.

There are two key ideas for that classification learning creates a model which logically explains stakeholders' behavior even if the data available is sparse.

1. A feature selection method for enhancing reliability of the learning result by classification learning was proposed. In this process, the important features chosen by the expert were always included. In our experiment, this approach succeeded in creating learning results which satisfied with the domain expert from RPG data which has large number of features and not enough amount of data set which cannot guarantee the learning performance. Moreover, the refining cost of domain experts was reduced because this approach automatically eliminates irrelevant features, which is very important because the cost in the visualization method is high.
2. A visualization method on a computer was proposed for refining the learning result by domain experts. Here, decision from both the learning result and RPG data itself are shown to the domain experts (they do not know which is from the RPG data). This approach enables domain experts to evaluate the learning result in objective and intuitive way. In our experiment, this approach was successful in gaining more comments to the learning result than the way to directly show it without this method. Moreover, this method was effective to persuade the domain experts to accept that the learning result is really explaining what was occurred in the RPG because the learning result and the RPG data showed the same kind of result.

Five modeling steps combined with these methods was established: "Survey and Interview to Stakeholders", "Role Playing Game", "Output of a reliable result from classification learning", "Refinement of the learning result through visualization on a computer", "Identification of subjects for further investigation". These steps were actually applied to the real example. Finally, the domain experts' hypothesis was validated and several subjects for further investigation were actually found.

ii. Modeling interactions with participatory simulation

There are the following two contributions for modeling interactions.

- *Development of a platform for designing interactions with a legacy simulator*

  There exists many platforms to support domain researchers developing and analyzing multiagent simulations, but they do not always give an environment to design and evaluate interactions. Therefore, an effective method was proposed; first, to provide a computational model suitable for interactions, an interaction layer is constructed and connected from the outside of a legacy simulator. Next, to configure the agents interacting ability and refine the interaction models in more realistic situation, a method for controlling the flow of information in the connection area is provided.

  As a concrete example, we realized an interaction layer by *Q* which is a scenario description language and connected it to CORMAS, which is often used in researches of the participatory method. Finally, we discussed the capability of our method and showed a process to refine an interaction model through a concrete implementation in the Fire-Fighter domain.

  This method is useful because a legacy simulator familiar with a community is re-used and tailored for design of interaction models. The effectiveness is also discussed in the real example in the northeast of Thailand.

- *Modeling process using participatory simulation*

  Four steps for modeling interactions were established: “Survey and Interview to Stakeholders”, “Role Playing Game”, “Interaction Design”, and “Participatory Simulation”. The point is that

RPG is situated before a participatory simulation. This is because RPG is suitable for communication between domain experts and stakeholders who are not always familiar with computer systems and research processes. Moreover, to create an environment that supports the domain expert in designing agent-avatar interaction so as to elucidate the behavior of the stakeholder, three kinds of interaction descriptions (interaction protocol, interaction scenario and avatar control scenario) were defined and the designing process where domain experts and system developer cooperate were combined with the modeling process.

Interaction scenario is for describing how agents interact with the other agents and avatars (e.g. process of cooperation and negotiation), while avatar control scenario is for describing how users (stakeholders) take part in a participatory simulation, which consists of four elements: 1) control of UI (display etc.), 2) observation of input by a user through UI, 3) sensing from other agents, and 4) acting to other agents. Interaction scenario and avatar control scenario have to be described in the framework of the interaction protocol.

Finally, a participatory simulation was developed on CORMAS/*Q*, and we succeeded in capturing a negotiation model among suppliers of rice seeds. In our experiment, the stakeholders could well understand the negotiation process and the influence of their decision in the simulation world. The domain expert could concentrate to observe one specific stakeholder and ask questions to the stakeholder in any time, which brought a deep understanding of the negotiation by the domain expert. This is because computer agents took roles of the other stakeholders in the simulation and the computer program automatically made progress of the simulation.

The proposed methodology has achieved the following two important points for multiagent simulations to build consensus among stakeholders. First, RPG and participatory simulation give shared experience of modeling with domain experts, which is important for stakeholders to find their solutions in the simulation results. Second, the modeling method with classification learning and participatory simulation enables domain experts to deeply understand and model stakeholders' actual behavior in the system. Indeed, the impact was confirmed in a real case study of agricultural economics in the northeast of Thailand.

7.2 Future Directions

In this thesis, the proposed methodology was applied to only a few case studies. In order to sophisticate our methodology, it is desirable to apply our method to as many domains as possible. The participatory method are being used in such many places as southeast Asia, Africa, and Europe for consensus building among stakeholders, and our method can be useful for these domains. The technical aspects to be developed in the future will be shown below.

- *Modeling method with other learning algorithms*

  This thesis could use a decision tree for expressing and learning the decision making. However, a decision tree does not always have ability to express all of decision making models. Therefore, it is necessary to bring or develop computational models and the learning methods which match domain experts' needs. Moreover, when participatory simulation is used for multi-players instead of RPG, we can use machine learning for extracting interaction models from the log data, in order to reduce the work of human. For example, a learning method for a state transition model used in modeling interactions in our case is one of our future works.

- *Creative thinking support for new features*

In our case, the elements for decision making (features) was given by the domain experts in advance. However, domain experts cannot always give all the features to explain the model. In the result of our experiment, the set of features for explaining the farmers' decision making were not perfect. Therefore, it is important to provide a method to find such necessary features. For this approach, both inductive learning which forms a model from data (e.g. a decision tree learning algorithm in our case) and deductive learning which infers a model from the background knowledge will be able to support creative thinking of domain experts.

- *New technologies for user participation*

  Participatory simulations can be used instead of the traditional RPG in order to reduce a large amount of labor for gathering stakeholders and staffs for RPG. A participatory simulation environment where users participate through the Internet can be one of the solutions. Gilbert *et al.* proposed such a environment [Gilbert 02], but a systematical method for constructing such participatory simulation has not been established and it is necessary to investigate such method in the future.

  We will also be able to use another kind of technology for user participation. For example, *Augmented experiment* [Ishida 04] can give an environment where stakeholders take part in the simulation world from their real work space, which enables domain experts to understand more actual behavior of stakeholders.

- *Helper agents for domain experts*

  In the modeling process of our thesis, computer scientists mediated the domain experts to use machine learning and construct a participatory simulation. Nowadays, many social scientists concern to multiagent simulations and therefore it is important to provide tools which facilitate them constructing a simulation. Many platforms for multiagent simulations have been proposed for this purpose, but in the future, social scientists will find capabilities and demands in machine learning and participatory simulation. For kindly supporting social scientists who are not familiar with such technologies, the style in which humans and the helper agents cooperate in developing simulations and modeling with machine learning [Drogoul 02] will be one of the solutions.

Bibliography

[Almuallim 94] H. Almuallim and T. G. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features," *Artificial Intelligence*, Vol. 69, pp. 279-305, 1994.

[Alvin 92] P. Alvin and D. Foley, "Decentralized, dispersed exchange without an auctioneer," *Journal of Economic Behaviour and Organization*, Vol. 18, pp. 27-51, 1992.

[Axelrod 97] R. Axelrod, "Advancing the art of simulation in the social sciences," In R. Conte, R. Hegselmann and P. Terna Eds., *Simulating Social Phenomena*, pp. 21-40, Springer-Verlag, 1997.

[Bah 03] A.Bah, I. Touré and Ch. Le Page, "An Agent-Based Model tool for multi-agent simulation for Understanding the Multiple Uses of space Land and Resources around a Drilling Sites in the Sahel," In *Proc. Int. Congress on Modeling and Simulation (ModSim-03)*, pp. 1060-1065, 2003.

[Balmer 04] M. Balmer, N. Cetin, K. Nagel and B. Raney, "Towards Truly Agent-Based Traffic and Mobility Simulations," In *Proc. of Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-04)*, pp. 60-67, 2004.

[Barbuceanu 95] M. Barbuceanu and M. S. Fox, "Cool: A language for describing coordination in multiagent systems," In *Proc. of the First Int. Conf. on Multi-Agent Systems (ICMAS-95)*, pp. 17-24, 1995.

[Bousquet 98] F. Bousquet, I. Bakam, H. Proton and C. Le Page, “Cormas: common-pool resources and multi-agent Systems,” Lecture Notes in Artificial Intelligence, Vol. 1416, pp. 826-838, Springer-Verlag, 1998.

[Bousquet 99] F. Bousquet, O. Barreteau, C. Le Page, C,C. Mullon and J. Weber, “An environmental modelling approach. The use of multi-agent simulations,” F. Blasco and A. Weill Eds., *Advances in environmental and ecological modeling*, Elsevier, pp. 113-122, 1999.

[Bousquet 01a] F. Bousquet, J. C. Castella, G. Trébuil, S. Boissau and S. P. Kam, “The Use of Multi-Agent Simulations in Companionable Modelling Approach for Agro-ecosystem Management,” *Int. Conf. on Integrated Natural Re-sources Management*, 2001.

[Bousquet 01b] F. Bousquet, G. Trébuil, S. Boissau, C. Baron, P. d’Aquino and J. C. Castella, “Knowledge integration for participatory land management: The use of multi-agent simulations and a companion modeling approach,” *Int. Conf. on Participatory technology development and local knowledge for sustainable land use in Southeast Asia*, 2001.

[Bousquet 02] F. Bousquet, O. Barreteau, P. d’Aquino, M. Etienne, S. Boissau, S. Aubert, C. Le Page C, D. Babin and J. C. Castella, “Multi-agent systems and role games: an approach for ecosystem co-management,” M. Janssen Eds., *Complexity and Ecosystem Management: The Theory and Practice of Multi-agent Approaches*, Edward Elgar Publishers, pp. 248-285, 2002.

[Bousquet 05] F. Bousquet and G. Trébuil, “Synergies between multi-agent systems and role-playing games in companion modeling for integrated natural resorce management in southeast Asia,” In *Proc. Int. Conf. Simulation and Modeling (SIMMOD-05)*, C6-03, 2005.

[Castella 01] J. C. Castella, S. Boissau, H. Lan Anh and O. Husson, “Enhancing communities’ adaptability to a rapidly changing environment in Vietnam uplands: the SAMBA role-play,” In *Proc. Int. Conf. on*

Sustaining Upland Development in Southeast Asia: Issues, Tools and Institutions for Local Natural Resource Management, 2001.

[Cohen 89] P. R. Cohen, M. L. Greenberg, D. M. Hart and A. E. Howe, "Trial by fire: Understanding the design requirements for agents in complex environments," *AI Magazine*, Vol. 10, No. 3, pp. 32-48, Fall, 1989.

[Collier 02] N. Collier, "Repast: An extensible framework for agent simulation," 2002.

[Davidsson 02] P. Davidsson, "Agent Based Social Simulation: A Computer Science View," *Journal of Artificial Societies and Social Simulation*, Vol. 5, No. 1, 2002.

[Decker 98] K. Decker, "Task Environment Centered Simulation," M. Prietula, K. Carley, and L. Gasser Eds., *Simulating Organizations: Computational Models of Institutions and Groups*, AAAI Press/MIT Press, 1998.

[Downing 00] T. E. Downing, S. Moss and C. Pahl-Wostl, "Understanding climate policy using participatory agent-based social simulation," In *Proc. Int. Workshop on Multi-Agent Based Simulation (MABS-00)*, S. Moss and P. Davidsson Eds., pp. 198-213, Springer-Verlag, 2000.

[Doran 97a] J. E. Doran, "Foreknowledge in artificial societies," In R. Conte, R. Hegselmann and P. Terna Eds., *Simulating Social Phenomena*, pp. 457-470, Springer-Verlag, 1997.

[Drogoul 94] A. Drogoul and J. Ferber, "Multi-agent simulation as a tool for studying emergent processes in societies," In J. E. Doran and N. Gilbert Eds., *Simulating Societies: The Computer Simulation of Social Phenomena*, pp. 127-142, UCL Press, 1994.

[Drogoul 02] A. Drogoul, D. Vanbergue and T. Meurisse, "Multiagent-based Based Simulation: Where are the Agents," J. S. Sichman, F.

Bousquet, P. Davidsson Eds., *Multi-Agent-Based Simulation II*, Lecture Notes in Artificial Intelligence, Vol. 2581, pp. 1-15, Springer-Verlag, 2002.

[Epstein 96] R. J. Epstein and R. Axtell, "Growing Artificial Societies - Social Science from the Bottom Up," MIT Press, Cambridge, MA, 1996.

[Farrell 98] W. Farrell, "How Hit Happen: Forecasting Predictability in a Chaotic Marketplace," Harpercollins, 1998.

[Ferber 98] J. Ferber and O. Gutknecht, "A meta-model for the analysis and design of organizations in multi-agent systems," In *Proc. of the Third Int. Conf. on Multi-Agent Systems (ICMAS-98)*, pp. 128-135, 1998.

[Gasser 02] L. Gasser and K. Kakugawa, "MACE3J: fast flexible distributed simulation of large, large-grain multi-agent systems," In *Proc. Int. Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS-02)*, pp. 745-752, 2002.

[Gilbert 99] N. Gilbert and K. G. Troitzsch, "Simulation for the social scientist," Open University Press, 1999.

[Gilbert 02] N. Gilbert, S. Maltby and T. Asakawa, "Participatory simulations for developing scenarios in environmental resource management," In *Proc. Third Workshop on Agent-Based Simulation*, pp. 67-72, 2002.

[Guyot 05] P. Guyot, A. Drogoul, C. Lemaitre, "Using Emergence in Participatory Simulations to Design MultiAgent Systems," In *Proc. Int. Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS-05)*, pp. 199-203, 2005.

[Helbing 00] D. Helbing, I. J. Farkas and T. Vicsek, "Simulating Dynamical Features of Escape Panic," *Nature*, Vol. 407, No. 6803, pp. 487-490, 2000.

[Ishida 02a] T. Ishida, "Digital City Kyoto: Social Information Infrastructure for Everyday Life," *Communications of the ACM (CACM)*, Vol. 45, No. 7, pp. 76-81, 2002.

[Ishida 02b] T. Ishida, "*Q*: A Scenario Description Language for Interactive Agents," *IEEE Computer*, Vol. 35, No. 10, pp. 42-47, 2002.

[Ishida 04] T. Ishida, "Society-Centered Design for Socially Embedded Multiagent Systems," In *Proc. of Int. Workshop on Cooperative Information Agents (CIA-04)*, Lecture Notes in Computer Science, Vol. 3191, pp. 16-29, 2004.

[Ishino 95] Y. Ishino, K. Yoshinaga and T. Terano, "SIBILE: A Decision Aid to Extract Knowledge from Data Using Simulated Breeding and Inductive Learning Techniques," In *Proc. Pacific-Asian Conf. on Expert Systems (PACES-95)*, pp. 448-455, 1995.

[Kita 03] H. Kita, H. Sato, N. Mori and I. Ono, "U-Mart System, Software for Open Experiments of Artificial Market," In *Proc. of IEEE Int. Symposium on Computational Intelligence in Robotics and Automation*, pp. 1328-1333, 2003.

[Kitano 97] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda and E. Osawa "RoboCup: The Robot World Cup Initiative," In Proc. of Int. Conf. on Autonomous Agents (Agents-97), pp. 340-347, 1997.

[Kohavi 97] R. Kohavi and G. John, "Wrappers for feature selection," *Artificial Intelligence*, Vol. 97, pp. 273-324, 1997.

[Kuhl 00] F. Kuhl, R. Weatherly and J.Dahmann, "Creating Computer Simulation Systems, An Introduction to the High Level Architecture," *Prentice Hall*, 2000.

[Kolb 95] M. Kolb, "A Cooperation Language," In *Proc. of the First Int. Conf. on Multi-Agent Systems (ICMAS-95)*, pp. 233-238, 1995.

[Kuwabara 95] K. Kuwabara, T. Ishida, and N. Osato, "AgenTalk: Describing multiagent coordination protocols with inheritance," In *Proc. of the 7th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI-95)*, pp. 460-465, 1995.

[Minar 96] N. Minar, R. Burkhart, C. Langton and M. Askenazi, "The Swarm Simulation System, A Toolkit for Building Multi-Agent Simulations," 1996.

[Mizuta 02] H. Mizuta and Y. Yamagata, "Web-based simulation 1: transaction cycle of agents and web-based gaming simulation for international emissions trading," In *Proc. Conf. on Winter Simulation (WSC-02)*, pp. 801-806, 2002.

[Murakami 03] Y. Murakami, T. Ishida, T. Kawasoe and R. Hishiyama, "Scenario Description for Multi-Agent Simulation" In *Proc. Int. Joint Conf. Autonomous Agents and Multiagent Systems (AAMAS-03)*, pp. 369-376, 2003.

[Murakami 05] Y. Murakami, Y. Sugimoto and T. Ishida, "Modeling Human Behavior for Virtual Training Systems," In *Proc. National Conf. Artificial Intelligence (AAAI-05)*, pp. 127-132, 2005.

[Nakanishi 03] H. Nakanishi, S. Nakazawa, T. Ishida, K. Takanashi and K. Isbister, "Can Software Agents Influence Human Relations? Balance Theory in Agent-mediated Communities," In *Proc. Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-03)*, pp. 717-724, 2003.

[Nakanishi 04a] H. Nakanishi and T. Ishida, "FreeWalk/Q: Social Interaction Platform in Virtual Space," In *Proc. ACM Symposium on Virtual Reality Software and Technology (VRST-04)*, pp. 97-104, 2004.

[Nakanishi 04b] H. Nakanishi, "FreeWalk: A Social Interaction Platform for Group Behavior in a Virtual Space," *International Journal of Human Computer Studies (IJHCS)*, Vol. 60, No. 4, pp. 421-454, 2004.

[Noda 03] I. Noda and P. Stone, "The RoboCup Soccer Server and CMU-nited Clients: Implemented Infrastructure for MAS Research," *Autonomous Agents and Multi-Agent Systems*, Vol. 7, No. 1-2, pp. 101-120, 2003.

[Perez 02] P. Perez, N. Ardlie, P. Kuneepong, C. Dietrich, W. S. Merritt, "CATCHCROP: modeling crop yield and water demand for integrated catchment assessment in Northern Thailand," *Environmental Modelling and Software*, Vol. 17, No. 3, pp. 251-259, 2002.

[Quinlan 86] J. R. Quinlan, "Induction for Decision Trees," *Machine Learning*, Vol. 1, No. 1, pp. 81-106, 1986.

[Quinlan 93] J. R. Quinlan, "C4.5: Programs for Machine Learning," Morgan-Kaufmann, 1993.

[Rickel 02] J. Rickel, S. Marcella, J. Gratch, R. Hill, D. Traum and W. Swartout, "Toward a New Generation of Virtual Humans for Interactive Experiences," *IEEE Intelligent Systems*, Vol. 17, pp. 32-38, 2002.

[Russell 02] S. J. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach (2nd edition)," Prentice Hall, 2002.

[Said 02] L. B. Said, T. Bouron and A. Drogoul, "Agent-based interaction analysis of consumer behavior," In *Proc. Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-02)*, pp. 184-190, 2002.

[Takahashi 02] T. Takahashi, S. Tadokoro, M. Ohta, and N. Ito, "Agent Based Approach in Disaster Rescue Simulation," *RoboCup 2001: Robot Soccer World Cup V*, Lecture Notes in Artificial Intelligence, pp. 102-111, Springer-Verlag, 2002.

[Torii 05a] D. Torii, T. Ishida, S. Bonneaud and A. Drogoul, "Layering Social Interaction Scenarios on Environmental Simulation," In P. Davidsson, B. Logan and K. Takadama Eds., *Multi-Agent and Multi-Agent-*

Based Simulation, Lecture Notes in Computer Science, Vol. 3415, pp. 78-88, Springer-Verlag, 2005.

[Torii 05b] D. Torii, F. Bousquet, and T. Ishida, “Extension of Companion Modeling Using Classification Learning,” *Trans. of the Japanese Society for Artificial Intelligence*, Vol. 20, No. 6, pp. 379-386, 2005 (in Japanese).

[Torii 05c] D. Torii, F. Bousquet, T. Ishida, G. Trébuil and C. Vejpas, “Using Classification Learning in Companion Modeling,” In *Proc. of the 8th Pacific Rim International Workshop on Multi-Agents (PRIMA-05)*, pp. 327-341, 2005.

[Torii 06a] D. Torii and T. Ishida, “Construction of Interaction Layer on Socio-Environmental Simulation,” *Trans. of the Japanese Society for Artificial Intelligence*, Vol. 21, No. 1, pp. 28-35, 2006 (in Japanese).

[Torii 06b] D. Torii, T. Ishida and F. Bousquet, “Modeling Agents and Interactions in Agricultural Economics,” In *Proc. of the 5th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS-06)*, 2006 (to appear).

[Torii 06c] D. Torii, T. Ishida and F. Bousquet “Modeling Negotiation by a Participatory Approach,” *Trans. of the Japanese Society for Artificial Intelligence*, Vol. 21, No. 3, pp. 287-294, 2006 (in Japanese).

[Tébuil 02] G. Trébuil, B. Shinawatra-Ekasingh, F. Bousquet, C. Thong-Ngam, “Multi-agent systems companion modeling for integrated watershed management: a northern Thailand experience,”. In *Proc. of the 3rd Int. Conf. on Montane Mainland Southeast Asia (MMSEA-3)*, pp. 349-358, 2002.

[Vejpas 04] C. Vejpas, F. Bousquet, W. Naivinit, G. Trébuil and N. Srisombat, “Participatory modeling for managing rainfed lowland rice variety and seed systems in lower northeast Thailand,” *Mekong Rice Conference*, OS3.4, 2004.

[Vincent 01] R. Vincent, B. Horling, and V. Lesser, "An Agent Infrastructure to Build and Evaluate Multi-Agent Systems: The Java Agent Framework and Multi-Agent System Simulator," T. Wagner and 0. Rana Eds., *Infrastructure for Scalable Multi-agent Systems*, Lecture Notes in Computer Science, Vol. 1887, pp. 102-127, Springer-Verlag, 2001.

[Waldrop 92] M. Waldrop, "Complexity: The Emerging Science at the Edge of Choas," Simon & Schuster, New York. 1992.

[Wellman 01] M. P. Wellman, P. R. Wurman, K. O'Malley, R. Bangera, S. Lin, D. Reeves and W. E. Walsh, "Designing the Market Game for a Trading Agent Competition," *IEEE Internet Computing*, Vol. 5, No. 2, pp. 43-51, 2001.

[Yamamoto 01] G. Yamamoto and H. Tai, "Performance Evaluation of an Agent Server Capable of Hosting Large Numbers of Agents," In *Proc. Int. Conf. on Autonomous Agents (Agents-01)*, pp. 363-369, 2001.

[Yamashita 05] T. Yamashita, K. Izumi, K. Kurumatani and H, Nakashima, "Smooth traffic flow with a cooperative car navigation system," In *Proc. of Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-05)*, pp. 478-485, 2005.

Publications

Major Publications

Journals

1. Daisuke Torii, Francois Bousquet and Toru Ishida, "Extension of Companion Modeling Using Classification Learning," *Transaction of the Japanese Society for Artificial Intelligence*, Vol. 20, No. 6, pp. 379–386, 2005 (in Japanese).

2. Daisuke Torii and Toru Ishida, "Construction of Interaction Layer on Socio-Environmental Simulation," *Transaction of the Japanese Society for Artificial Intelligence*, Vol. 21, No. 1, pp. 28–35, 2006 (in Japanese).

3. Daisuke Torii, Toru Ishida and Francois Bousquet, "Modeling Negotiation by a Participatory Approach," *Transaction of the Japanese Society for Artificial Intelligence*, Vol. 21, No. 3, pp. 287–294, 2006 (in Japanese).

International Conference

1. Daisuke Torii, Toru Ishida and Francois Bousquet, "Modeling Agents and Interactions in Agricultural Economics," In *Proceedings of the 5th International Conference on Autonomous Agents and Multiagent Systems (AAMAS2006)*, 2006 (to appear).

Other Publications

Awarded Papers

1. Daisuke Torii and Toru Ishida, “Connecting Q and CORMAS: Toward Multi-Layer Multiagent Simulation,” *The 18th Annual Conference of the Japanese Society for Artificial Intelligence*, 2H1-01, 2004 (in Japanese). JSAI Annual Conference Award.

Workshops

1. Daisuke Torii, Akishige Yamamoto and Toru Ishida, “Execution of Multi-Agent Scenario on Distributed Environment,” In *Proceedings of 3rd Joint Agent Workshops & Symposium (JAWS2003)*, pp. 121–128, 2003 (in Japanese).

2. Daisuke Torii and Toru Ishida, “Extraction of Decision Making Rules Using Machine Learning in Participatory Modeling,” In *Proceedings of 4th Joint Agent Workshops & Symposium (JAWS2004)*, pp. 363–370, 2004 (in Japanese).

3. Daisuke Torii, Toru Ishida, Stephane Bonneaud and Alexis Drogoul, “Layering Social Interaction Scenarios on Environmental Simulation,” In P. Davidsson, B. Logan and K. Takadama Eds., *Multi-Agent and Multi-Agent-Based Simulation*, Lecture Notes in Computer Science, Vol. 3415, pp. 78–88, Springer-Verlag, 2005.

4. Daisuke Torii, Francois Bousquet, Toru Ishida, Guy Trébuil and Chirawat Vejpas, “Using Classification Learning in Companion Modeling,” In *Proceedings of the 8th Pacific Rim International Workshop on Multi-Agents (PRIMA2005)*, pp. 327–341, 2005.

5. Daisuke Torii, Francois Bousquet, and Toru Ishida, “of Interaction Scenarios in Companion Modeling,” In *Proceedings of 5th Joint*

Agent Workshops & Symposium (JAWS2005), pp. 353–360, 2005 (in Japanese).

Conventions

1. Daisuke Torii and Toru Ishida, “Participatory Modeling and Multiagent-Based Simulation,” *Technical Report of IEICE*, AI2004-2, Vol. 104, No. 133, pp. 7–12, 2004 (in Japanese).