

Society-Centered Design for Socially Embedded Multiagent Systems

Toru Ishida

Department of Social Informatics, Kyoto University
ishida@i.kyoto-u.ac.jp

Abstract. To realize large-scale socially embedded multiagent systems, this paper proposes a new system design methodology towards *society-centered design*. We have already developed the *scenario description language Q*, which describes interaction protocols that link agents to society. Using the virtual space called *FreeWalk*, wherein agents behave under given *Q* scenarios, we explain each step of *society-centered design*. The process consists of *participatory simulation*, where agents and human-controlled avatars coexist in virtual space to jointly perform simulations, and *augmented experiment*, where an experiment is performed in real space by human subjects, scenario-controlled agents, and human extras. As an application of society-centered design, we are working on *mega navigation*, where millions of humans and socially embedded agents collaborate for developing services for traffic control, crisis management and large-scale event navigation.

1 Introduction

The waterfall model has been used as a software development model for a long time. Given the increase in human-computer interaction, however, it has become essential to use the user-centered design approach when creating usable and accessible interactive systems. It is natural to ask whether or not we need a different model for mobile/ubiquitous/pervasive computing, where thousands or millions of computers or electronic devices are connected in an ad hoc manner. Agent-based software engineering has been intensively studied [12]. Large scale multiagent systems may be a natural solution [14,20]. The problem is how to ensure that they yield adequate behavior: “adequate behavior” does not merely mean computationally correct behavior, but appropriate social behavior when embedded in human societies.

It is known that the behavior of socially embedded systems is hard to predict, not only because the system is highly distributed, but also because the system is exposed to the impact of human interaction. In this paper, in contrast to user-centered design, we pursue *society-centered design*, where participatory technologies are applied to confirm the adequateness of socially embedded systems; the following steps provide a description of this approach.

1. Describe interaction protocols linking service agents and users so as to define the expected behaviors of socially embedded agents. We have been developing the scenario description language *Q* [10] for this purpose.

2. Perform multiagent simulation by modeling the autonomous behaviors of service agents and users under the given protocols. The simulation takes place in virtual space by service agents and simulated users (we call both *agents*). Results of the simulation can estimate how the entire system would work in society.
3. Replace some of the simulated users by human-controlled avatars to perform a *participatory simulation*; the human subjects directly control their avatars. The simulation is performed in virtual space, and the avatars are controlled by the humans sitting in front of their desktop computers. From the results of the participatory simulation, we can improve the models of service agents and users, as well as the interaction protocols.
4. Perform experiments in a real space to try out the entire system with human subjects. Since the number of human subjects is often limited, however, the experiment should be augmented by scenario-controlled agents and human extras. We called this the *augmented experiment*; note that most of participants are virtual constructs.

Drogoul *et al.* proposed a methodological process for developing multiagent-based simulations, and introduced the idea of the participatory design of simulations [5]. Bousquet *et al.* applied role games to modeling multiagent systems [3]. In this paper, we propose participatory technologies to conduct simulations and experiments for large-scale socially embedded systems.

In the following sections, we explain the society-centered design approach in detail, through visualizing it by using *FreeWalk* [16], a virtual space platform developed by the digital city project in Kyoto [11]. At the end of this paper, we apply the above steps to design *mega navigation*; the goal is to create services for traffic control, crisis management and large-scale event navigation where millions of people may be involved.

2 Scenario Engineering

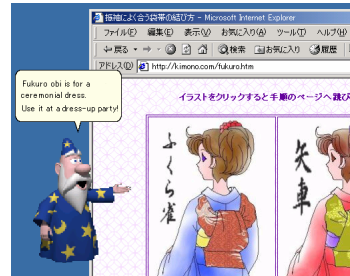
In this section, we focus on interaction protocols to link agents, not on the internal mechanisms of the agents. We view interaction protocols as behavioral guidelines of socially embedded agents, while agents keep certain degree of autonomy under the given constraints. We call the description of interaction protocols as *scenarios*. For realistic applications, since scenario writers are often not computing professionals, the correctness of scenarios for multiple agents is not guaranteed. Problems in scenarios should be adjusted not by proving their correctness but by conducting rehearsals. To monitor rehearsals, visualizing the scenario execution processes becomes essential.

Q is a scenario description language for multiagent systems that allows us to define how agents are expected to interact with each other. Q 's language functionality is summarized as follows.

```

(defscenario card14 ()
  (opening
    (#t
      (!speak "Hm-hum, you are so enthusiastic.")
      (!speak "Then, how about this page?")
      (!display :url "http://kimono.com/index.htm")
      (go reactions-to-user)))
    (reactions-to-user
      ((?watch_web :url "http://kimono.com/type.htm")
        (!speak "There are many types of obi.")
        (!speak "Can you tell the difference?")
        (go reactions-to-user))
      ((?watch_web :url "http://kimono.com/fukuro.htm")
        (!gesture :animation "GestureLeft")
        (!speak "Fukuro obi is for a ceremonial dress.")
        (!speak "Use it at a dress-up party!")
        (go reactions-to-user))
      ((?watch_web :url "http://kimono.com/maru.htm")
        (card42 self)
        (go reactions-to-user))
      ((?timeout :time 20)
        (go closing))))
  (closing
    (#t
      (!speak "Did you enjoy Japanese Kimono?")
      (!speak "OK, let's go to the next subject."))))

```



Designed by Tomasan Kimono School

a) A sample scenario in Q language

Card ID	14	Card Name	Visiting Kimono Web site	Card Type	User Initiative
Opening	Action				
	Hm-hum, you are so enthusiastic. Then, how about this page? http://www.kimono.com/index.htm				
Reactions to Users' Mouse Click (Repeat)	Mouse Click	Cue	http://kimono.com/type.htm	Action	
		http://kimono.com/fukuro.htm	There are many types of obi. Can you tell the difference? (GestureLeft)		
		http://kimono.com/maru.htm	Fukuro obi is for a ceremonial dress. Use it at a dress-up party! (Evaluate Card42)		
	No Reaction	Seconds	20	Action (End of Repeat)	
Closing	Action				
	Did you enjoy Japanese Kimono? OK, let's move on to the next subject.				

b) The same scenario in IPC representation

Fig. 1. Q Scenario and Interaction Pattern Card (with Y. Murakami and A. Yamamoto).

Cues and Actions

An event that triggers interaction is called a *cue*. Cues are used to request agents to observe their environment. No cue is permitted to have any side effect. Cues keep on waiting for the event specified until the observation is completed successfully. Compared to cues, *actions* are used to request agents to change their environment.

Scenarios

Guarded commands are introduced for the situation wherein we need to observe multiple cues simultaneously. A guarded command combines cues and actions;

after one of the cues becomes true, its corresponding actions are performed. A scenario is used for describing protocols in the form of an extended finite state machine, where each state is defined as a guarded command. Scenarios can be called from other scenarios.

Agents and Avatars

Agents, avatars and a group of agents can be defined. An agent is defined by a scenario that specifies what the agent is to do. Even if a group of agents executes the same scenario, the agents exhibit different actions as they interact with their local environment (including other agents and avatars). Avatars are controlled by humans, and do not usually require any scenario.

Figure 1(a) shows an example scenario for Microsoft agents. Agents can be autonomous or dependent. If autonomous, scenarios cannot be complex; if not, scenario writers can specify all details. Note that the granularity of cues and actions depends on two independent factors: the level of agent autonomy, and the degree of preciseness required by the scenario writer.

Figure 2 shows the Q architecture for handling scenarios. When a scenario is given to a particular agent, a Q processor is instantiated and coupled to the corresponding agent system. The agent system is usually capable of hosting multiple agents. For example, the three dimensional virtual space FreeWalk can support hundreds of agents simultaneously. Consequently, the Q interpreter is designed to execute hundreds of scenarios simultaneously. Though the Q processor is implemented at the top of the Scheme, the program interface supports C++ and JAVA software, so it is easy to add Q to legacy agent systems. Q has been already combined with Microsoft, FreeWalk, CORMAS [2,22] and Caribbean [23,24] agents.

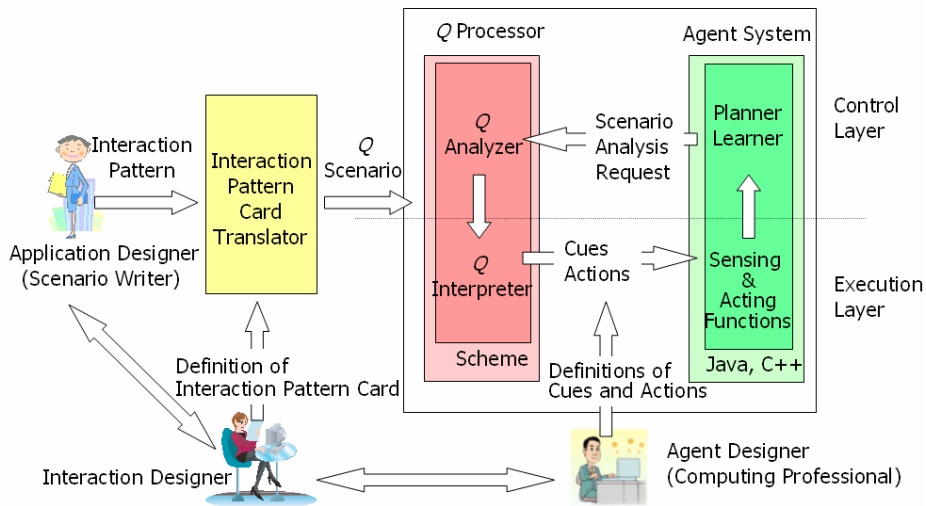


Fig. 2. Q Architecture.

In order to develop successful socially embedded systems, scenario writers must provide appropriate scenarios to agents. It is necessary to establish a procedure that models agents at an appropriate level of abstraction. We propose the following three-step procedure for creating scenarios [15].

STEP1: Define a Vocabulary

A scenario writer and an agent system developer agree upon cues and actions as the interface between them. Note that cues and actions are not provided a priori by Q language but are defined for each application domain.

STEP2: Write Scenarios

The scenario writer describes scenarios using Q language, while the agent system developer implements cues and actions.

STEP3: Extract Interaction Patterns

Patterns of interaction are extracted from already written scenarios, and are used to define *Interaction Pattern Cards (IPC)*. Scenario writers then describe scenarios by using the cards, and use their experience to further improve the definition of the cards.

Figure 1(b) indicates an IPC representation that is the exactly equivalent of the Q scenarios in Figure 1(a). Note that IPC is not merely an Excel representation of Q language; it is a language to express interaction patterns in a particular domain. The introduction of IPC not only provides a simple means of writing scenarios, but also facilitates dialogs between scenario writers and agent system developers.

3 Participatory Simulation

There are two types of *multiagent-based simulations* depending on their purposes; a) *analytic multiagent-based simulations* with a simple internal model of agents (hereafter referred to as analytic simulations) and b) *synthetic multiagent-based simulation* with a complex internal model of agents (hereafter referred as synthetic simulation). Analytic simulations have been used to analyze complex social systems. Here, the KISS principle (Keep It Simple, Stupid) is often applied [1]. 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 agent interaction. Hence, agents are expressed using a simple computational model that incorporates limited parameters. This approach is mainly used in the analysis of the relationship between the macro properties of the entire system and the micro properties of the agents constituting the system [7]. On the other hand, synthetic simulation is used for the reproduction of reality-based situations. Agent models reflecting the real world are created to make the simulation as realistic as possible. This approach is used in an early stage of system development [13], in the examination of strategies for decision making, and in education or training [19].

In our society-centered design approach, we first conduct synthetic multiagent-based simulations, and then replace some of the agents by human-controlled avatars: humans directly control agents in multiagent-based simulations. We call simulation

including human-controlled avatars as *multiagent-based participatory simulation* (hereafter referred to as *participatory simulation*).

Figure 3 illustrates how to realize participatory simulations. Since the scenario processor interprets interaction protocols and requests agents to perform sensing and acting functions, it is easy to realize participatory simulation by replacing scenario-controlled agents with human-controlled avatars. We can monitor the entire process of simulation by visualizing the virtual space as described in Figure 3(b). Recording videos of behaviors of human subjects in real space is useful for analyzing simulation results.

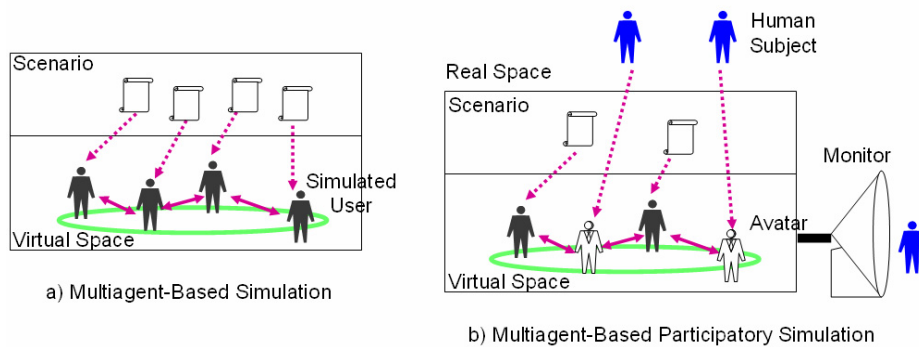


Fig. 3. Participatory Simulation.

We conducted a participatory simulation in the evacuation domain. We simulated the controlled experiments conducted by Sugiman in 1988 [21]. He established a simple environment with human subjects to determine the effectiveness of two evacuation methods: the *Follow-direction method* and the *Follow-me method*. In the former, the leader shouts out evacuation instructions and eventually moves toward the exit. In the latter, the leader tells a few of the nearest evacuees to follow him and actually proceeds to the exit without verbalizing the direction of the exit. Sugiman used university students as evacuees and monitored the progress of the evacuations with different number of leaders. The experiment was held in a basement that was roughly ten meters wide and nine meters long; there were two exits, one of which was not obvious to the evacuees. The ground plan of the basement is shown in Figure 4.

Our first evacuation simulation used scenarios with simulated leaders and evacuees in a two-dimensional virtual space as displayed in Figure 4(a). This simulation showed that the Follow-me method outperforms the Follow-direction method when there are enough leaders. Since the simulation results closely parallel those recorded in Sugiman's controlled experiment, we could conclude that the agents were reasonably well modeled [15]. We then examined the same scenario in the three-dimensional virtual space FreeWalk. A difference between the two-dimensional and three-dimensional simulations is that the latter allows humans to join the simulation: human subjects can sense their environment and control their avatars in the virtual space.

A participatory simulation was then performed with six human subjects: we replaced six out of twenty scenario-controlled agents by human-controlled avatars. The virtual space is designed so that human subjects cannot distinguish agents and

avatars. By analyzing the logs and videos of simulations, we learned that a) evacuees look for leaders not by their ears but by their eyes, b) when finding a leader, evacuees tend to approach the leader, even if the Follow-direction method is performed, and c) when losing their leaders, evacuees follow other evacuees moving in the same direction.

Once an accurate model is acquired by participatory simulations, it becomes possible to simulate an experiment that has not conducted in real space. For example, Sugiman's experiment was performed by two or four leaders, but we can vary the number to clarify the relation between the number of leaders and the time required for evacuation. Moreover, we can explore an effective combination of the Follow-me and Follow-direction methods.

Participatory simulations are useful to educate or train people once we obtain accurate agent models and interaction protocols. It provides people with a vicarious experiential learning environment wherein evacuation or other drills can be experienced.

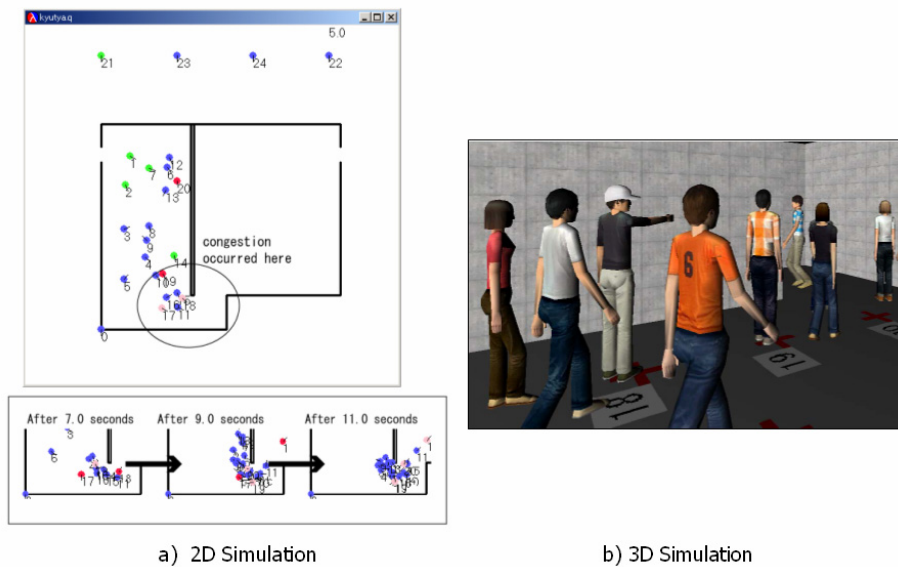


Fig. 4. Evacuation Simulation (with T. Kawasoe and K. Minami).

Many technical issues remain with participatory simulations. First, time management becomes difficult in participatory simulations. In our current design of multiagent-based simulations, the scenario interpreter has no responsibility for time management: time is controlled by the agent systems executing the sensing and acting functions. Therefore, in FreeWalk/ Q , just as any combination between Q and other agent systems, only FreeWalk manages time. To make this reasonable, however, we should assume that the scenario processor runs far more rapidly than the agent systems executing sensing and acting functions. This assumption becomes problematic if we use high-performance grid platforms as agent systems [8].

Furthermore, since human subjects play the simulation in real space, we need to keep time consistency among virtual and real spaces.

Another issue is that learning from human subjects is not straightforward. By analyzing the logs of participatory simulation, we can improve the interaction protocols and agent models in several different ways. For example, in the evacuation domain, if the participatory simulation results differ from the expectation, there are several choices: to modify a given interaction protocol, such as Follow-me or Follow-direction methods, to modify simulated human models, or to regard the results just as an exception. To appropriately learn lessons from simulation results, we should carefully analyze the data. Machine learning can help this analysis, and so will become a central issue in participatory simulation.

4 Augmented Experiment

To understand how people accept/reject socially embedded systems, many real-world experiments have been conducted. A well-known example involves video phones. Since the value of video phones depends on the number of users, and user behavior in everyday life is not easy to simulate in virtual space, it is essential to observe how users accept and utilize the new technology. In mobile/ubiquitous/pervasive computing, however, because of a large number of electronic devices are embedded in human society, it is costly or often impossible to conduct experiments in real space. Augmented experiments have the concept of performing experiments with a small number of human subjects in real space with augmentation by multiagent systems. We call such experiments *multiagent-base augmented experiments* (hereafter referred to as *augmented experiments*).

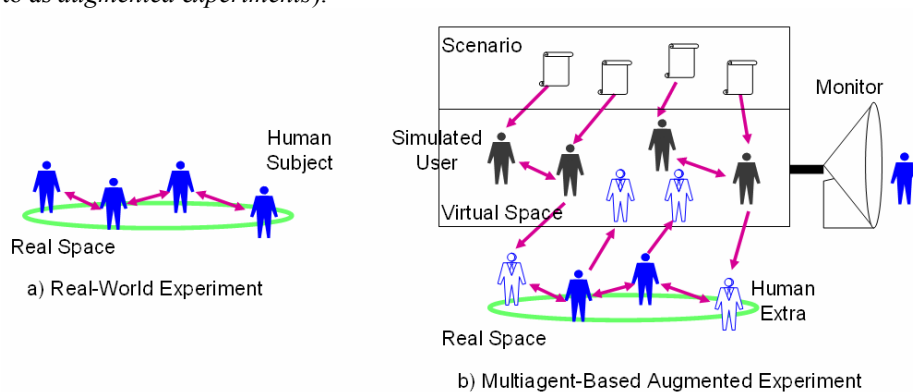


Fig. 5. Augmented Experiment.

Figure 5 illustrates how to realize augmented experiments. To provide enough reality to the human subjects, we need human extras around the subjects. In contrast to participatory simulations, the human extras do not control avatars, while agents in virtual space control the human extras. To monitor an entire experiment, the behavior of human subjects should be reproduced in virtual space. This can be realized if the

real space is equipped with enough sensors. To analyze the entire experiment, however, video capture of the real space is indispensable in augmented experiments.

The augmented experiment approach is not effective for controlled experiments with just a few subjects; instead, human subjects would be used rather than extras and complex virtual systems. For testing evacuation methods in open space like railway stations, however, we need augmented experiments, since there is no other way to conduct realistic experiments. To perform augmented experiments, we have placed more than twenty cameras in Kyoto subway stations, and successfully captured the movements of passengers in real time. We then reproduced their behavior in virtual space. We have implemented a monitoring system based on *transcendent communication* [17]. The bird's-eye view of real space is reproduced on the screen of the control center so as to monitor the experiment. Figure 6 is a snapshot of a monitoring system; evacuees on a subway station platform are projected in virtual space. A leader in the control center can easily monitor the experiment. Furthermore the leader can point at particular passengers on the screen, and talk to them through mobile phones. When the monitor detects pointing operations, the audio connection is immediately activated between the control center and indicated passengers. A dragging operation indicating a rectangular area enables the leader to broadcast an announcement to a group of passengers.

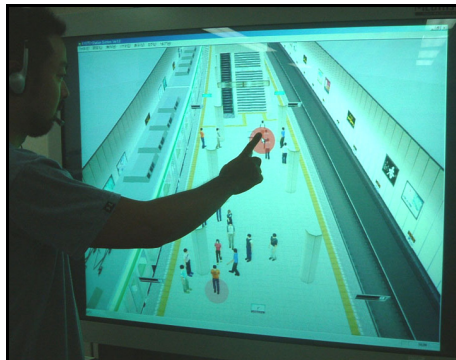


Fig. 6. Transcendent Communication at Kyoto Subway Station (with H. Nakanishi and H. Ito).

The difficulty with augmented experiments is providing the human subjects with a sufficient level of reality. This is not easy if the number of human extras is limited. If there are not enough extras, their roles should be reassigned during the experiment. To control human extras by scenarios in a timely fashion, agents need to receive sensing information from the corresponding human extras. This is possible by embedding sensors into the experiment environment. Another difficulty is how to evaluate the results of experiments. The evaluation of agent models and interaction scenarios can be done by comparing the results of augmented experiments to those of multiagent-based simulations. Learning issues are similar to participatory simulations, but the evaluation is more difficult, since the behaviors of human subjects in real space are hard to analyze.

5 Example: Mega Navigation

To apply society-centered design to large scale multiagent systems, we started a project called *mega navigation*; it assumes that at least a million agents are embedded in society. One agent serves one human by guiding him/her for traffic control, crisis management and large-scale event navigation. To draw a clear image of mega navigation, examples of the Beijing Olympics in 2008 are listed below. We assume that third generation mobile phones with GPS will be in wide use in Beijing at that time.

Restaurant recommendation: Suppose one company is planning to develop a restaurant recommendation service for foreign tourists. Tourists can get his location by GPS and can take a photo of the restaurants visited and their recommendations. The company expects tourists to provide the recommendations voluntarily. The question is whether or not the tourists will participate and upload/download recommendations. Since the information available can be consumed at no cost (the abuse of the commons), the response seems quite personal.

Smart tag: This company is also developing a smart tag for tourists so that they can easily find new friends who have similar hobbies, preferences, and schedules. Can we expect tourists to store their personal information in the tag? The company believes that the smart tag will help tourists to create their own communities, which would make the restaurant recommendation service community-oriented.

Traffic control: Traffic jams will increase in Beijing during the Olympics. But nobody knows how crowded the roads and Olympic facilities will become. The company is planning to guide people to less crowded roads and venues by using data of the tourists' smart tags. Will tourists follow the instructions? To decrease traffic jams in Beijing, demand responsive bus services might be a good idea. However, it is hard to know how tourists will respond to this service.

Crisis management: Suppose a disaster happens during the Olympics. The company provides navigation service for people who have GPS mobile phones. It is expected that people with mobile phones send their locations and telephone numbers to the company. At the control center, the locations of such people are displayed on a large screen. Using transcendent communication, the control center talks to people to get local information, and asks them to guide surrounding people to safe areas. Will people voluntarily join this service? How many GPS mobile phone users are necessary to provide safe guidance to all citizens in Beijing?

Research on large-scale multiagent systems has just started [18]. To handle a million people, we have been developing a mega navigation platform by combining *Q* and Caribbean [21,22]. Caribbean is a large-scale agent server designed for handling a million agents. Agents are created in the Caribbean server and persist in the server. Caribbean agents communicate with other agents via asynchronous peer-to-peer messaging. When a message is delivered to a particular agent, the corresponding activity of the agent is invoked. Agents can invoke service objects, which provide common services such as database management. Figure 7 shows how we connect *Q*

and Caribbean. As in the case of FreeWalk/ Q , time management is performed by Caribbean. Since Caribbean is a mega-scale agent server, however, it appears that Q cannot guarantee to run scenarios faster than Caribbean. To make time management easier, we are currently developing a translator to turn Q scenarios into Caribbean activities.

We plan to use Caribbean/ Q to conduct participatory simulations and augmented experiments on the service examples listed above. Traffic control problems [6] and demand bus services [4] are common examples in the agent research community. Multiagent-based simulation for demand responsive bus services is straightforward: we first define an interaction protocol between passengers and bus companies, run simulation with agents that simulate passengers, bus drivers and bus companies.

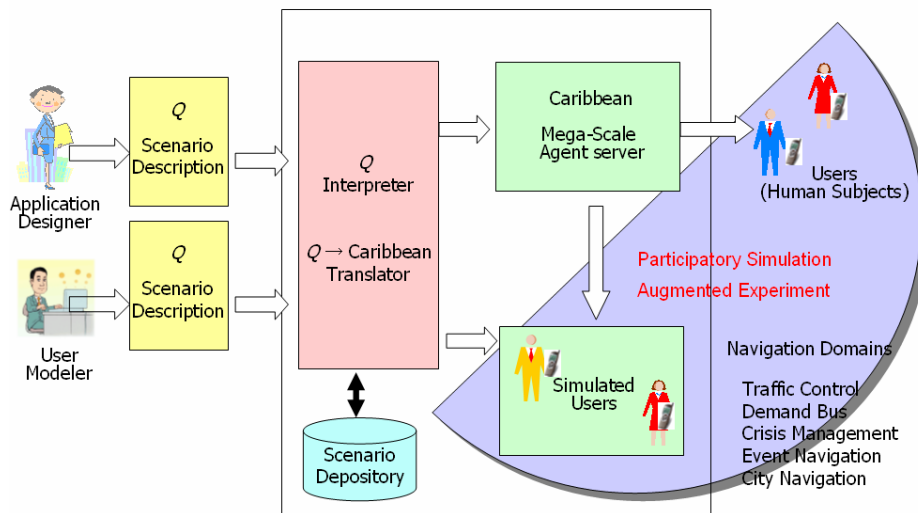


Fig. 7. Mega Navigation Platform.

The corresponding participatory simulations are not straightforward. We want to focus on passengers' decision making. It is not obvious, though, how human subjects in front of desktop computers can make realistic decisions given several options: wait for a bus, give up and walk, take a taxi, etc. Augmented experiments, on the other hand, would be enjoyable. A human subject with a mobile phone would actually make a phone call to the bus control center. One bus is allocated for him/her in virtual space, but the human subject cannot ride a virtual bus. To provide enough reality to the human subject, a real vehicle (maybe a chartered taxi if the budget is limited) is used to realize the virtual bus. The human subject then rides on the bus. Several simulated users may share the bus with the human subject. The bus stops at several places to let the simulated users get off the bus. Finally, the bus stops at the destination of the human subject, lets him/her get off, and the bus becomes virtualized afterwards. The real vehicle then becomes available to realize another virtual bus.

6 Conclusions

In this paper, we proposed society-centered design for making large-scale agent systems run in human societies. We proposed a paradigm shift from user-centered design, where individual usability is a major concern, to *society-centered design*, where usability in the community becomes essential. Using multiagent-based simulations, we invented a system design process including participatory simulations and augmented experiments so as to confirm the adequateness of socially embedded multiagent systems. FreeWalk/ Q is used to visualize those steps of society-centered design. Though the concept of society-centered design is open-ended, we proposed to include the following three components.

Scenario Engineering: We developed Q , a scenario description language, to define interaction protocols that link agents and humans. Scenarios also foster the emergence of dialogs between agent designers (computing professionals) and application designers (scenario writers). Q has been combined with several legacy agent systems: Microsoft, FreeWalk, CORMAS and Caribbean agents. The reason that totally different agent systems can be supported is because Q is designed to describe interactions among legacy agents.

Participatory simulation: We call multiagent-based simulations that include human-controlled avatars *participatory simulations*. In our framework, since agents are controlled by scenarios, it is easy to realize participatory simulations by replacing agents by human-controlled avatars. We can monitor the entire process of a participatory simulation by visualizing the virtual space.

Augmented experiment: We call real-world experiments augmented by simulated users *augmented experiments*. Given that mobile/ubiquitous/pervasive computing will see a large number of computers or electronic devices embedded in human society, it will be costly or impossible to conduct experiments in real space. Augmented experiments enable us to perform experiments with a small number of human subjects.

We then introduced *mega navigation* as an example of society-centered design. We have started working on guide services for a million people who are connected via wireless networks. We implemented Caribbean/ Q as a mega navigation platform. By using examples of services for the Beijing Olympics, we showed the potential of applications such as traffic control, crisis management and large scale event navigation.

Society-centered design focuses on the process to embed large-scale multiagent systems into human societies, and is suitable for designing community support systems [9]. This methodology differs from the Gaia methodology [25], which focuses on an early stage of designing multiagent systems mainly for enterprise applications. As more humans and agents coexist, the need for describing interaction scenarios becomes essential. If we allow agents to be completely autonomous, the whole system becomes hard to control. It is necessary to give social constraints as scenarios so that the behaviors of agents are reasonable. Remaining research issues include the topic of how agents learn to behave under given social constraints.

Acknowledgements

The author thanks to Stéphane Bonneaud, Francois Bousquet, Alexis Drogoul, Reiko Hishiyama, Hideaki Ito, Tomoyuki Kawasoe, Zhiqiang Liu, Kazuhisa Minami, Yohei Murakami, Hideyuki Nakanishi, Daisuke Torii, Hirofumi Yamaki, Akishige Yamamoto and Gaku Yamamoto for discussing and contributing to this work. The work is supported by Japan Science and Technology Agency.

References

1. R. Axelrod. The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration. Princeton University Press, pp. 4-5, 1997.
2. F. Bousquet, I. Bakam, H. Proton and C. Le Page. Cormas: Common-Pool Resources and Multiagent Systems. *Lecture Notes in Artificial Intelligence* 1416, pp. 826-838, 1998.
3. F. Bousquet, O. Barreteau, P. Aquino, M. Etienne, S. Boissau, S. Aubert, C. Le Page, D. Babin, and J.-C. Castella. Multi-Agent Systems and Role Games: Collective Learning Processes for Ecosystem Management. M. Janssen Ed. *Complexity and Ecosystem Management: The Theory and Practice of Multi-Agent Approaches*, Edward Elgar Publishers, 2002.
4. M. Diana and M. M. Dessouky. A New Regret Insertion Heuristic for Solving Large-Scale Dial-a-Ride Problems with Time Windows. *Transportation Research, Part B* 38, pp. 539-557, 2004.
5. A. Drogoul, D. Vanbergue, T. Meurisse. Multiagent-based Based Simulation: Where are the Agents? *International Workshop on Multi-Agent Based Simulation (MABS-02)*, LNAI 2581, pp. 1-15, 2002.
6. K. Dresner and P. Stone. Multiagent Traffic Management: A Reservation-Based Intersection Control Mechanism. *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-04)*, 2004.
7. J. M. Epstein and R. L. Axtell. *Growing Artificial Societies: Social Science from the Bottom Up*. MIT Press, 1996.
8. L. Gasser and K. Kakugawa. MACE3J: Fast Flexible Distributed Simulation of Large, Large-Grain Multi-Agent Systems. *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, pp. 745-752, 2002.
9. T. Ishida Ed. *Community Computing and Support Systems*. Lecture Notes in Computer Science, State-of-the-Art Survey, 1519, Springer-Verlag, 1998.
10. T. Ishida. Q: A Scenario Description Language for Interactive Agents. *IEEE Computer*, Vol. 35, No. 11, pp. 54-59, 2002.
11. T. Ishida. Digital City Kyoto: Social Information Infrastructure for Everyday Life. *Communications of the ACM (CACM)*, Vol. 45, No. 7, pp. 76-81, 2002.
12. N. R. Jennings. On Agent-Based Software Engineering. *Artificial Intelligence*, Vol. 177, No. 2. pp. 277-296, 2000.
13. H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda and E. Osawa. RoboCup: The Robot World Cup Initiative. *International Conference on Autonomous Agents (Agents-97)*, pp. 340-347 1997.
14. K. Kurumatani. Social Coordination with Architecture for Ubiquitous Agents - CONSORTS. *International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC-03)*, 2003.
15. Y. Murakami, T. Ishida, T. Kawasoe and R. Hishiyama. Scenario Description for Multi-Agent Simulation. *International Joint Conference on Autonomous Agents and Multiagent*

- Systems (AAMAS-03)*, pp. 369-376, 2003.
16. H. Nakanishi, C. Yoshida, T. Nishimura and T. Ishida. FreeWalk: A 3D Virtual Space for Casual Meetings. *IEEE Multimedia*, Vol. 6, No. 2, pp. 20-28, 1999.
 17. H. Nakanishi, S. Koizumi, T. Ishida and H. Ito. Transcendent Communication: Location-Based Guidance for Large-Scale Public Spaces. *International Conference on Human Factors in Computing Systems (CHI-04)*, pp. 655-662, 2004.
 18. E. Ogston, B. Overeinder, M. van Steen and F. Brazier. A Method for Decentralized Clustering in Large Multi-Agent Systems. *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, pp. 789-796, 2003.
 19. J. Rickel and W. L. Johnson. Task-Oriented Collaboration with Embodied Agents in Virtual Worlds. In: Cassell, J. Sullivan, S. Prevost, and E. Churchill Eds. *Embodied Conversational Agents*, pp. 95-122, MIT Press, 2000.
 20. T. Sugawara S. Kurihara, K. Fukuda, T. Hirotsu, S. Aoyagi and T. Takada. Reusing Coordination and Negotiation Strategies in Multi-Agent Systems for Ubiquitous Network Environment. *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-04)*, 2004.
 21. T. Sugiman and J. Misumi. Development of a New Evacuation Method for Emergencies: Control of Collective Behavior by Emergent Small Groups. *Journal of Applied Psychology*, Vol. 73, No. 1, pp. 3-10, 1988.
 22. D. Torii and T. Ishida, S. Bonneaud and A. Drogoul. Layering Social Interaction Scenarios on Environmental Simulation. *AAMAS Workshop on Multiagent and Multiagent-based Simulation (MAMABS)*, 2004.
 23. G. Yamamoto and Y. Nakamura. Architecture and Performance Evaluation of a Massive Multi-Agent System. *International Conference on Autonomous Agents (Agents-99)*, pp. 319-325, 1999.
 24. G. Yamamoto and H. Tai. Performance Evaluation of an Agent Server Capable of Hosting Large Numbers of Agents. *International Conference on Autonomous Agents (Agents-01)*, pp. 363-369, 2001.
 25. F. Zambonelli, N. R. Jennings and M. Wooldridge. Developing Multiagent Systems: The Gaia Methodology. *ACM Transactions on Software Engineering and Methodology*. Vol. 12, No. 3, pp. 317-370, 2003.