

Participatory technologies for designing ambient intelligence systems

Toru Ishida* and Hiromitsu Hattori

Department of Social Informatics, Kyoto University, Kyoto, 606-8501, Japan

Abstract. To realize large-scale socially embedded ambient intelligence systems, this paper proposes a design methodology towards *society-centered design*. Participatory technologies and multiagent systems are essential in the new system design perspective. Multiagent systems make it possible to test and predict the behavior of socially embedded systems. We have already developed the *scenario description language*, which describes interaction protocols that link agents to society. We use the virtual space, wherein agents behave under given scenarios, in explaining 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. For realizing realistic interactions between agents and humans during participatory simulations, an agent model that can reproduce human-like agent behaviors is needed. We show a direction for agent modeling based on learning from humans in actual application environments.

Keywords: Participatory simulation, participatory modeling, society-centered design, scenario engineering

1. Introduction

The waterfall model has long been used as a software development model. Given the increase in human-computer interaction, however, it has become essential to employ 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 ambient intelligence systems [1], where thousands or millions of computers or electronic devices are connected in an ad hoc manner. Though large scale multiagent systems may be a natural solution [9,23], the problem is how to ensure that they yield adequate behavior: “adequate behavior” does not merely mean computationally correct behavior, but includes 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, we propose to apply multiagent technologies

to design systems that are to be embedded in society. In contrast to user-centered design, we pursue *society-centered design*, where participatory technologies are applied to confirm the adequateness of socially embedded systems.

We propose multiagent-based participatory design methodologies (participatory design hereafter) to test socially embedded systems. For designing socially embedded systems, simulations in virtual space are not enough to confirm that they can reproduce the reality of an actual application environment. Therefore, we introduce real world experiments to the process of participatory design, so as to bridge the gap between participatory simulations and services in operation. The process of participatory design is as follows.

1. Describe interactions between human users and the socially embedded system so as to define the user behaviors expected when interacting with the system (interaction model hereafter).
2. Perform a multiagent-based simulation by modeling users under the given interaction scenarios

* Corresponding author. E-mail: ishida@i.kyoto-u.ac.jp.

(agent model hereafter). The simulation takes place in virtual space and involves a large number of simulated users. Results of the simulation can predict how the entire system would work in society.

3. Replace some of the simulated users by human-controlled avatars to perform a multiagent-based participatory simulation (participatory simulation hereafter). The simulation is performed in virtual space, and the avatars are controlled by human subjects.
4. Perform experiments in a real space to try out the entire system with human subjects. Since the number of subjects is often limited, the experiment should be augmented by a large number of simulated users in virtual space. We called this the multiagent-based augmented experiment (augmented experiment hereafter).

2. Scenario engineering

To realize the participatory design process, we first separate agent models from interaction models: the former covers the beliefs, desires, intentions, and emotions of human users, and the latter covers protocols, methods, rules, or laws that guide users when interacting with the socially embedded systems. We use extended finite state automata for describing interaction models, while various models including production systems and Bayesian networks can be used to describe the agent models that approximate users.

Many of the languages proposed for describing agent behavior are based on agent internal mechanisms. For social agents, however, we should also consider protocols among agents and humans. We need a language that can describe interaction scenarios between agents and humans based on agent external roles: scenarios do not depend on agent internal mechanisms; their goal is to describe how scenario writers should be able to *request* agents to behave. We call the descriptions of interaction protocols *scenarios*.

Q is a scenario description language that can describe interaction scenarios [8,11]. Since Q cannot control the internal mechanism of agents, they do not have executable functions. Consequently, Q has been connected to various agent systems including Free-Walk [14], Caribbean [22] and Cormas [3]. The major components of Q are explained below.

An event that triggers interaction is called a *cue*. Cues are used to request agents to observe their envi-

ronment. No cue is permitted to have any side effect. Cues remain waiting for the event specified until the observation is completed successfully. Compared to cues, *actions* are used to request agents to change their environment. 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. *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. Scenarios can be called from other scenarios. *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.

In order to run a successful multiagent simulation, scenario writers must assign appropriate scenarios to agents. A scenario, however, differs from a program in that no explicit specification is given in advance. Hence, the process of describing software cannot be applied to scenario description. It is necessary to create a process that models agents at an appropriate level of abstraction by observing the real world. We propose the following process for creating scenarios. First, 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 but are defined for each application domain. Second, the scenario writer describes scenarios, while the agent system developer implements cues and actions. The scenario writer then conducts experiments in both real and virtual environments. The scenario writer utilizes the knowledge obtained from the experiments in the real world to refine the original scenarios.

3. Participatory simulation

There are two types of *multiagent-based simulations* and they have different 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 to as synthetic simulation). Analytic simulations have been used to analyze complex social systems. Here, the KISS principle (Keep It Simple, Stu-

pid) is often applied [2]. 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 functionality. This approach is mainly used to analyze the relationship between the macro properties of the entire system and the micro properties of the agents constituting the system. On the other hand, synthetic simulations are used to reproduce 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 [16], 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 [5,6], and then replace some of the agents by human-controlled avatars. We call the simulation that includes human-controlled avatars the participatory simulation [12,21].

Below we illustrate how a multiagent-based simulation is realized. The scenario processor interprets interaction models and requests agents in virtual space to perform sensing and acting functions. Note that, since agents are autonomous and have their own agent models, though agents receive instructions according to the scenarios, there is no guarantee that they will behave as requested.

We can easily extend multiagent-based simulations to yield participatory simulations by replacing some of the scenario-guided agents with human-controlled avatars. Below we illustrate how human subjects and agents can cooperatively perform a simulation. Just as with video games, human subjects can join the simulation by controlling avatars via joy sticks, mice, or other input devices. To analyze the simulation results, we monitor the entire process of the simulation by visualizing the virtual space. In addition to videotaping the virtual space, we record how the human subjects control their avatars. Recording human behavior is useful for analyzing the simulation results and for improving the agent and interaction models.

Participatory simulations are particularly useful when conducting controlled experiments: they make it easy to prepare the application environments for testing, and user behavior can be recorded for later analysis. However, it sometimes fails to provide enough reality to allow the testing of ambient intelligence environments.

In summary, a participatory simulation consists of 1) *agents* for modeling users, 2) *avatars* to represent human subjects, 3) *scenarios* for modeling interactions, 4) *human subjects* to control avatars, 5) *virtual space* to represent real space, and 6) a *monitor* to visualize simulations ongoing in the virtual space.

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 the large number of electronic devices embedded in human society, it is costly or often impossible to conduct experiments in real space. The concept of augmented experiments is to perform experiments with a small number of human subjects in real space with augmentation by one or more multiagent systems [10,13].

Below we illustrate how augmented experiments are realized. In a real world experiment, human subjects communicate and participate in an experiment in real space. We then introduce a virtual space into the real world experiment. The sensors in the real space capture the behavior of human subjects for reproduction in the virtual space. The sensors can be cameras, RFIDs, or GPS depending on the environment. Conducting an augmented experiment is possible if the real space is equipped with enough sensors. By using the virtual space, we can monitor the entire experiment from various viewpoints. Furthermore, we can communicate with the human subjects in the real space through their avatars in the virtual space. Transcendent communication is a new monitoring interface, where a visually simulated public space provides a more flexible and consistent view than regular surveillance systems [15].

The process of an augmented experiment is as follows. In parallel with a real world experiment, a large scale multiagent-based simulation is conducted in virtual space: the experiment is augmented by the simulation. To provide social reality to the human subjects, scenario-guided extras are placed around the subjects. In contrast to the participatory simulations, the human extras do not control avatars: the human extras in the real space are controlled by the scenario-guided agents. For example, we can use

human extras acting as evacuation leaders to conduct disaster evacuation drills. In contrast to participatory simulations where human subjects sense and act in virtual space, augmented experiments allow subjects to sense and act in real space.

In summary, an augmented experiment consists of 1) *agents* for modeling users, 2) *avatars* to represent human subjects, 3) *scenarios* for modeling interactions, 4) *human subjects* to act out the experiment, 5) *virtual space* to represent real space, 6) *a monitor* to visualize the experiment in real space enhanced by simulations in virtual space, 7) *sensors* to reproduce human subjects in virtual space as avatars, 8) *communication channels* between real and virtual spaces, and 9) *human extras* to represent agents in virtual space for interacting with human subjects.

5. Case study

As a first step in addressing *society-centered design*, we simulated the controlled experiments conducted by Sugiman [20]. 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 numbers of leaders.

The experiment was held in a basement that was roughly ten meters wide and nine meters long; there were three exits, one of which was a safe exit that was not obvious to the evacuees. The two methods were tested with two and four leaders each; there were sixteen evacuees in every run. All sixteen evacuees were successfully guided to the safe exit in all runs except for the “Follow-me method” with two leaders. After carefully studying Sugiman's experiment, we developed a vocabulary, cues and actions, suitable for evacuation simulations.

Cues and actions roughly fall into three groups: motion, conversation, and others. With regard to action, motion can be subdivided into movement, rotation, gesture, and appearance. By employing asynchronous actions, multiple actions can be concurrently executed. Actions within the same group cannot, however, be executed concurrently. Once the vocabulary is determined, the agent system developer implements cues and actions. Two simulators were used in our evacuation simulation; one for a two-



Fig. 1. Virtual Experiment.

dimensional space and the other for a three-dimensional space. These two different simulators can use the same scenario, since they share the same cues and actions.

We use FreeWalk as the platform for the three-dimensional simulation. It enables agents to interact with nonverbal cues; for example, gestures like pointing at something can be used. An example of a FreeWalk screen is shown in Fig. 1. One big difference between the two-dimensional and three dimensional simulators is that the latter allows humans to project their avatars into the virtual space. Namely, three-dimensional simulators provide people with a vicarious experiential learning environment wherein evacuation or other emergency drills can be experienced.

Every day, more than 300,000 passengers pass through Kyoto station, the main railway station in Kyoto. In this station, using FreeWalk/Q, we installed a disaster evacuation system that tracks passengers to help them navigate based on their current positions. Far beyond conventional navigation systems, which simply announce route information using public loudspeakers, our system sends instructions to individuals via their mobile phones. Augmented experiments are required for testing the evacuation system, because there is no other way to conduct experiments with enough reality.

The augmented experiment was designed as follows. As the *sensors*, we placed twenty eight cameras in Kyoto station, and captured the movements of passengers in real time. The cameras are specially designed to detect passenger behavior but not personal features. As the *virtual space*, we used FreeWalk, a three dimensional virtual city system, to reproduce the passengers' behavior. We implemented a *monitor* based on transcendent communication. Figure 2 includes a snapshot of the monitoring system; *human subjects* on the station platform are projected as *avatars* in virtual space. A bird's-eye view of the real space is reproduced on the screen of the control center so that evacuation leaders in the center can easily monitor the experiment. To establish *communication channels*, the leader merely selects particular passengers on the screen, and talk to them through



Fig. 2. Indoor Experiment.

their mobile phones. When the monitor detects a pointing operation, a wireless connection is immediately activated between the control center and the indicated passenger or passengers. A multiagent-based simulation with a large number of *agents* controlled by evacuation *scenarios* was performed in parallel with an experiment in real space.

From this indoor experiment, we learned that insufficient visual reality prevented the human subjects from recognizing the crowd of agents around the staircase. It appears that the usability of an augmented experiment depends significantly on the user interface employed for interacting with the agents.

One year after the indoor experiment, we implemented a large-scale outdoor evacuation system for emergency situations using Caribbean/Q. This system architecture is close to that of our indoor experiment, but the *sensors* were GPS devices instead of omnidirectional cameras, and as the *virtual space*, we used a two dimensional map instead of a three dimensional virtual city system. The virtual space was displayed on the *monitor* screen of the control center in a birds-eye view, so the leader could grasp how evacuees were behaving in the experiment. Evacuees on a screen consisted of a small number of *avatars* representing the *human subjects* in real space, and a large number of *agents* controlled by evacuation *scenarios* in virtual space. In the actual experiment, ten to thirteen humans and three thousand agents participated in each trial of the augmented experiment.

The locations of human subjects in real space were projected into virtual space based on their positions acquired by GPS. This map showed fires, blocked routes, and safe areas in real time. The human subjects could always get the latest map by sending their location. The evacuation leaders assigned evacuation destinations and directions to each evacuee through the monitor screen shown in Fig. 3. The leader issued high level instructions to the evacuees using the map, and precise navigation instructions were automatically generated for each evacuee. Dragging operations indicating a rectangular area enabled the leader to broadcast an announcement to a group of evacuees.

The human subjects could grasp the state of experiment from the screen of their mobile phones. The

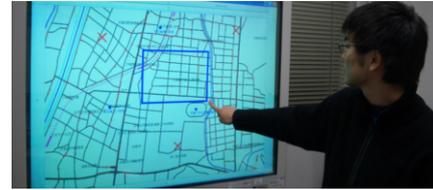


Fig. 3. Outdoor Experiment.

map around the place the subject was standing was displayed together with information of fires, blocked routes, and safe areas, just as on the monitor in the control center. The moves of other evacuees were also displayed on their mobile phones. From this experience, we learned that maps can be an excellent interface between human subjects and agents as well as evacuation leaders. Unlike the indoor experiment, since route selection was the main issue in the outdoor experiment, the human subjects did not have much difficulty in imagining the disaster situation.

6. Learning from humans

So far, we have not discussed agent models. We view interaction models as behavioral guidelines of human users playing within socially embedded systems; users retain autonomy within the given guidelines. The question is whether or not users will follow the guidelines in an actual environment. In other words, the effectiveness of the interaction models depends on the agent models, which include user personalities such as deliberative and reactive. Therefore, agent models reproducing realistic behavior are critical for conducting participatory simulations.

In previous work, agent models were often constructed based on the knowledge of experts and modelers [4,17]. Humans, however, can take diverse behaviors depending on their surroundings. For realizing an agent model that can exhibit human-like and reasonable behavior in diverse situations, another obvious approach is to learn from human behavior: to observe human behaviors in an application domain, and to turn the observed behaviors into computational models. The challenge is how to deal with the inconsistency of human behavior. The behavior-selection of humans is generally non-deterministic. Even in an identical environment, a human may behave differently. A computational behavior model has to be able to represent such non-deterministic behavior. Additionally, the behavior model must be flexible enough to support unknown environments.

To construct behavior models, we need realistic human behavior data. However, in the real world, it

is often difficult to conduct controlled experiments for obtaining data due to drastic changes in the environment. For example, in urban traffic, the traffic status is ever changing so that it is quite hard to obtain useful driving behavior data. In addition, there are cases in which it is impossible to conduct experiments, such as a huge disaster. We can apply participatory simulations to offset this omission. They are useful in conducting controlled experiments and obtaining human behavior data in unknown environments since we can simulate any environment desired.

One approach to modeling the inconsistency of human behavior is to use a Bayesian network [18]. Bayesian networks can explicitly represent causal relationships between propositions. They also allow probabilities to be ascribed to the degree of belief. By using Bayesian networks, we can use graph structures to express dependencies between the surrounding environment, mental state, and behaviors. Uncertainty of behavior-selection is also expressed by assigning probabilities to each graph-node. To construct a behavior model, we first try to catch the decision-making factors from human subjects in participatory simulations by interviewing them. At the next step, each obtained factor is defined as a probabilistic node and we encode dependencies between factors and behaviors into a graph-structure. Probabilities assigned to each node can be calculated using accumulated human behavior data obtained in participatory simulations. One important advantage of behavior modeling based on Bayesian networks is that the resulting models can work in unknown environments; the Bayesian network can adapt to varying environments by revision of the probabilities.

Clearly, as participatory simulations are becoming more realistic, human behaviors during participatory simulations are becoming more realistic. This means that the reality of human behavior models depends on the quality of participatory simulations. We thus iteratively improve simulation quality by the following process: conduct a participatory simulation; acquire and analyze behavior data; construct behavior model; apply constructed models to agents. Initially, the simulation environment is really artificial. However, after several iterations, the behavior models of a number of agents are replaced with ones constructed in the previous stages. That is to say, the simulation environment becomes more realistic.

Consider urban traffic simulations; in the initial traffic simulations, the driving behaviors of vehicle agents may be naive. For example, vehicle agents

may simply drive at the same speed. However, the characteristics of acceleration and braking become more human-like and specific driving operations, such as lane changing, overtaking, can be realized as the driving behavior models are refined. As a result, simulated road traffic becomes complicated like that in the real world, and the obtained driving behavior data becomes more realistic.

7. Conclusion

In this paper, we proposed the participatory design methodology for ambient intelligence systems, which are to be embedded in human society. Though the concept of participatory design is open-ended, we include the following two components.

We call multiagent-based simulations *participatory simulations* when they include human-controlled avatars and scenario-guided agents. We can monitor the entire process of a participatory simulation by visualizing the virtual space. We call real-world experiments *augmented experiments* when they are associated with large-scale multiagent-based simulations. In ubiquitous/pervasive computing, because a large number of electronic devices will be embedded in public spaces and in continuous use, it is often impossible to conduct experiments with a large number of human subjects. Augmented experiments enable us to perform such trials with a small number of human subjects. We used augmented experiments to conduct indoor and outdoor experiments in the city of Kyoto [7].

In order to realize realistic agent behaviors, we need technologies that can learn from humans. For example, we can model the inconsistency of human behavior by using a Bayesian network. The problem is how to acquire realistic human behavior data to construct the network. Conducting participatory simulations resolves this issue. By iterating the modeling and participatory simulations, we can refine the quality of behavior models through probability updating and the alteration of network structures.

References

- [1] J.C. Augusto, and P. McCullagh. Ambient Intelligence: Concepts and Applications. International Journal on Computer Science and Information Systems, Vol. 4, No. 1, pp. 1-28, 2007
- [2] R. Axelrod. The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration. Princeton University Press, pp. 4-5, 1997.

- [3] F. Bousquet, O. Barreteau, P. Aquino, M. Etienne, S. Bois-sau, 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*, Edward Elgar Publishers, 2002.
- [4] P. Davidsson, J. Holmgren, J.A. Persson, and L. Ramstedt. Multi Agent Based Simulation of Transport Chains. *AAMAS-08*, pp. 1153-1160, 2008.
- [5] A. Drogoul, D. Vanbergue and T. Meurisse. Multiagent-Based Simulation: Where are the Agents? *LNAI* 2581, pp. 1-15, 2002.
- [6] P. Guyot, A. Drogoul and C. Lemaître. Using Emergence in Participatory Simulations to Design Multi-Agent Systems. *AAMAS-05*, pp. 199-203, 2005.
- [7] T. Ishida. Digital City Kyoto: Social Information Infrastructure for Everyday Life. *Communications of the ACM*, Vol. 45, No. 7, pp. 76-81, 2002.
- [8] T. Ishida. Q: A Scenario Description Language for Interactive Agents. *IEEE Computer*, Vol. 35, No. 11, pp. 54-59, 2002.
- [9] T. Ishida, L. Gasser and H. Nakashima Eds. *Massively Multi-Agent Systems I*. LNAI, 3446, 2005.
- [10] T. Ishida, Y. Nakajima, Y. Murakami and H. Nakanishi. Augmented Experiment: Participatory Design with Multi-agent Simulation. *IJCAI-07*, pp. 1341-1346, 2007.
- [11] Y. Murakami, T. Ishida, T. Kawasoe and R. Hishiyama. Scenario Description for Multi-Agent Simulation. *AAMAS-03*, pp. 369-376, 2003.
- [12] Y. Murakami, Y. Sugimoto and T. Ishida. Modeling Human Behavior for Virtual Training Systems. *AAAI-05*, pp. 127-132, 2005.
- [13] Y. Nakajima, H. Shiina, S. Yamane, H. Yamaki and T. Ishida. Disaster Evacuation Guide Using a Massively Multi-agent Server and GPS Mobile Phones. *IEEE/IPSJ Symposium on Applications and the Internet*, 2007.
- [14] H. Nakanishi and T. Ishida. FreeWalk/Q: Social Interaction Platform in Virtual Space. *VRST-04*, pp. 97-104, 2004.
- [15] H. Nakanishi, S. Koizumi, T. Ishida and H. Ito. Transcendent Communication: Location-Based Guidance for Large-Scale Public Spaces. *CHI-04*, pp. 655-662, 2004.
- [16] I. Noda and P. Stone. The RoboCup Soccer Server and CMUnited Clients: Implemented Infrastructure for MAS Research. *Autonomous Agents and Multi-Agent Systems*, Vol. 7, No. 1-2, pp. 101-120, 2003.
- [17] P. Paruchuri, A.R. Pullalarevu, and K. Karlapalem. Multi-agent Simulation of Unorganized Traffic. *AAMAS-02*, pp.176-183, 2002.
- [18] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufman Publishers, 1988.
- [19] J. Rickel and W.L. Johnson. Task-Oriented Collaboration with Embodied Agents in Virtual Worlds. J. Cassell, J. Sullivan, S. Prevost, and E. Churchill Eds. *Embodied Conversational Agents*, pp. 95-122, MIT Press, 2000.
- [20] D. Torii, T. Ishida and F. Bousquet. Modeling Agents and Interactions in Agricultural Economics. *AAMAS-06*, pp. 81-88, 2006.
- [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] G. Yamamoto and Y. Nakamura. Architecture and Performance Evaluation of a Massive Multi-Agent System. *Agents-99*, pp. 319-325, 1999.
- [23] S. Yamane and T. Ishida. Meta-level Control Architecture for Massively Multiagent Simulations. *Winter Simulation Conference*, pp. 889-896, 2006.