Simulator platform that enables social interaction simulation –SIGVerse: SocioIntelliGenesis simulator–

Tetsunari Inamura, Tomohiro Shibata, Hideaki Sena, Takashi Hashimoto, Nobuyuki Kawai, Takahiro Miyashita, Yoshiki Sakurai, Masahiro Shimizu, Mihoko Otake, Koh Hosoda, Satoshi Umeda, Kentaro Inui and Yuichiro Yoshikawa

***Abstract*— Understanding mechanisms of intelligence of human beings and animals is one of the most important approaches to develop intelligent robot systems. Since the mechanisms of such real-life intelligent systems are so complex, physical interactions between agents and their environment and the social interactions between agents should be considered. Comprehension and knowledge in many peripheral fields such as cognitive science, developmental psychology, brain science, evolutionary biology, and robotics is also required. Discussions from an interdisciplinary aspect are very important for implementing this approach, but such collaborative research is time-consuming and labor-intensive, and it is difficult to obtain fruitful results from such research because the basis of experiments is very different in each research field. In the social science field, for example, several multi-agent simulation systems have been proposed for modeling factors such as social interactions and language evolution, whereas robotics researchers often use dynamics and sensor simulators. However, there is no integrated system that uses both physical simulations and social communication simulations. Therefore, we developed a simulator environment called SIGVerse that combines dynamics, perception, and communication simulations for synthetic approaches to research into the genesis of social intelligence. In this paper, we introduce SIGVerse, its example application and perspectives.**

I. INTRODUCTION

Understanding the mechanism of intelligence in human beings and animals is one of the most important approaches to developing intelligent robot systems. Since the mechanisms of such real-life intelligent systems are so complex, such as the physical interactions between agents and their environment and the social interactions between agents, comprehension and knowledge in many peripheral fields are required. We combined the words Social, Intelligence, and Genesis to coin the word SocioIntelliGenesis (SIG), to express this concept. To acquire a better understanding of human and robotic intelligence, we focus on a synthetic approach to research into the elucidation of the genesis of social intelligence, to cover aspects such as physical interactions between bodies and their environments, social interactions between agents, and the role of evolution.

To further that approach, we have set up interdisciplinary discussions with a wide viewpoint covering various research fields, such as cognitive science, developmental psychology, brain science, evolutionary biology, and robotics. Discussions from an interdisciplinary aspect are very important for implementing this approach, but such collaborative research is time-consuming and labor-intensive, and it is difficult to obtain fruitful results from such research because the basis of experiments is so very different in each research field.

In the social sciences field, for example, several multi-agent simulation systems have been proposed for modeling factors such as social interactions and language evolution, whereas robotics researchers often use dynamics and sensor simulators. However, there is no integrated system that uses both physical simulations and social communication simulations.

In this paper, we propose a simulator environment called SIGVerse, which combines dynamics, perception, and communication simulations.

II. INTEGRATED SIMULATION BETWEEN DYNAMICS, PERCEPTION AND COMMUNICATION

Many robot simulation systems are being developed around the world, to simulate the dynamics of robot systems. One well-known example is the Open Architecture Humanoid Robotics Platform (OpenHRP)[1], which is a humanoid robot simulator developed by the Advanced Institute of Science and Technology (AIST), the University of Tokyo, and the Manufacturing Science and Technology Center (MSTC). This simulator has become popular not only in Japan but also abroad, to promote research into humanoid robot control. The latest version of the simulation, called OpenHRP3[2], is currently being developed by AIST, the University of Tokyo, and General Robotix Inc. Another example is Webots[3], which is a commercial product. This system enables users to simulate multi-robot environments with dynamics calculations. The Player/Stage/Gazebo suite[4] is freeware and is also well-known. Microsoft also has released *Robotics Studio*[5] to develop a software of autonomous agents from the similar background. These systems provide

T.Inamura is with National Institute of Informatics/The Graduate Univ. for Adv. Studies (SOKENDAI), Japan
T.Shibata is with Nara Institute of Science and Technology, Japan
H.Sena is Writter
T.Hashimoto is with Japan Advanced Institute of Science and Technology
N.Kawai is with Nagoya Univ., Japan
T.Miyashita is with Advanced Telecommunications Research Institute International, Japan
Y.Sakurai is with Production I.G, Japan
M.Shimizu is with Osaka Univ., Japan
M.Otake is with the Univ. of Tokyo, Japan
K.Hosoda is with Osaka Univ., Japan
S.Umde is with Keio Univ., Japan
K.Inui is with Tohoku Univ., Japan
Y.Yoshikawa is with JST/Osaka Univ., Japan

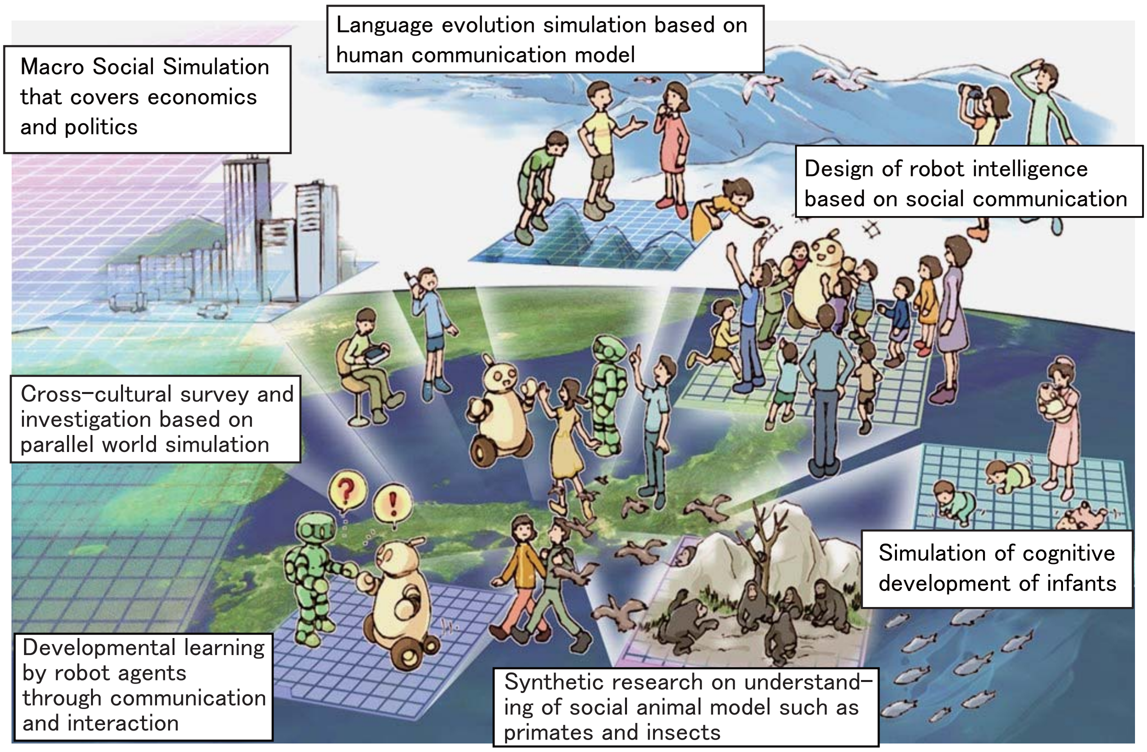

Fig. 1. Concept sketch of SIGVerse

multi-agent environments with dynamics simulations, but there hasn't been much consideration of the simulation of sensor perceptions. Since the communication simulations between agents provided in these packages are just simple signal transfers, it is difficult to use them to simulate the effects of the physical conditions of the agents for dialogue-based communication abilities and qualities.

Meanwhile, large-scale multi-agent systems are gaining attention from the social sciences field. One example of these is a project called "Construction of Agent-Based Social Systems Science" by the COE program of the Tokyo Institute of Technology, Japan; another project is called "Research into Social Order Fluctuations by Multi-Agent Simulator", which is supported by the Japanese government's MEXT; and a simulation system called GPGSiM[6] has also been proposed. In the field of language evolution, a system that has been proposed simulates language transmission between agents which is based on a repeatable learning model[7], However, such simulators do not consider the physical perception layer such as visual and auditory sensors, nor the physical communication layer such as limitations of communications based on the condition of each agent. The integration of dynamics, perception, and communication in the simulation world will play a great role in this social sciences field.

In this paper, we propose a novel concept for developing a simulation system that combines dynamics, physical perceptions, and communications for a multi-agent system; and discuss the potential of a prototype system based on that concept. The concept of this simulation system is shown in Figure 1.

This method is considered useful for setting evaluation experiments relating to human-machine cooperation, by not just simply building social agents that have physical bodies, but by also building an interface in which the user can act as an operator to manipulate an avatar within a virtual space. With a virtual space system such as SecondLife in which users manipulate avatars, it is possible to evaluate and analyze social systems in which large numbers of users participate and interact. However, introducing agents such as intelligent robots that can act autonomously into such an environment, analyzing their interactions with users, and evaluating the capabilities of the agents would necessitate changes to the core software of SecondLife, which would be extremely difficult. In addition, the time-scale within the SecondLife system is equal to real time, which imposes constraints in that it is necessary to continue the interactions over long periods of time when evaluating interactive strategies between agents or observing repercussions. For example, when performing an experiment into cooperation in which an expert robot that has already acquired a skill transmits that skill to a rookie robot in a virtual environment, something on the order of thousands or tens of thousands of trials are required. This means that it is necessary to make the experiment proceed smoothly by performing it at a time-scale that is faster than real time.

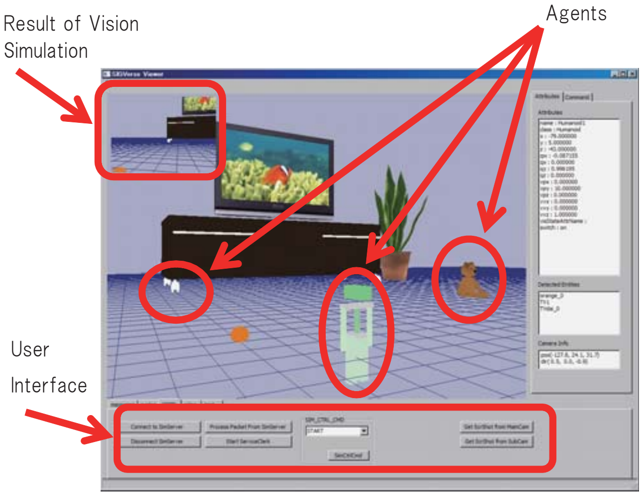

Fig. 2. A screenshot of user interface

III. Integration Simulation Platform: SIGVerse

A. Dynamics simulation

We use the Open Dynamics Engine (ODE)[8] for dynamic simulations of interactions between agents and objects. Basically, the motions of each agent and object are calculated by the dynamics engine, but the user can control the calculations to reduce simulation costs. A switch flag can be set for each object and agent to turn off the dynamics calculations if required.

B. Perception simulation

This system can provide the senses of vision, sound, force, and touch. We use OpenGL for visual simulations, to provide each agent with a pixel map that is a visual image derived from the viewpoint and field of view of that agent. In this case, the perception simulation has several levels that control the abstract level of perception.

One concept of this simulation system is that, since the system can be shared by researchers from several academic fields, each researcher can use any desired sensing information to suit the experimental conditions of his or her domain. For example, if one researcher wants raw visual information to study image processing and understanding, this system returns a raw pixel map as the result of the perception simulation. If another researcher wants the results of image comprehension instead of raw visual information, this system provides abstracted information instead of a pixel map. At a highly abstract perception level, the user is sent symbolized visual information, which comprises data such as the color, shape, size, and position of each object within sight, together with characteristic information on the name and ID number of each object. The visual perception processing also considers occlusions so that if an object is behind another object, the perception processing omits the hidden object from the list.

In simulating the sense of hearing, we did not adopt any simulation of the acoustic field or echoes, from consideration of calculation costs, but performed communications with audio data. We also enabled the effect of the volume of sound attenuating in inverse relationship with the square of distance, based on a setting of a condition that the voice emitted by an agent becomes more difficult to hear with distance. It is also possible to set the system so that only voices within a certain threshold distance are acquired. The system also has a high-order information provision function which transmits not just raw information that is unedited audio data, but also the results of audio recognition, in a similar way to the sense of sight described above.

For the sense of touch, it is possible to acquire force and torque information between objects that has been calculated mainly by ODE. It is also possible to simulate on/off tactile sensors by decisions as to whether or not there is contact.

C. Communication simulation

Basically, agents communicate by a simulation of the sense of hearing, but the design of the system enables the handling of a variety of modes of communication that are not limited to the sense of hearing, such as communications by optical signals or through a wireless network. With optical communications, it is possible to check whether light has been received by the sense of sight simulation described in the previous section, by making a sensor that receives light act as a camera. With wireless communications, it is possible to set constraints such that signals are transmitted only within the range of a certain distance.

With this system, not only can agents within the virtual environment communicate with each other, it is also possible to provide a function that enables interactions between the virtual environment and users in the real world. An example of the display of the virtual environment is shown in Figure 2.

There are already research platforms based on systems such as SecondLife and online Role-Playing Games (RPGs) as related systems[9] [10], but it is basically impossible to implement dynamics and perception simulations with them. Note that there are also modalities through gestures and body language, but at the current stage of development, protocols relating to such forms of communication have not been set. Concerning this problem, we plan to enable support for switching between a mode that directly communicates internal states, such as the intention behind a gesture, and a mode that transmits gesture patterns without modification to enable behavior recognition.

D. Simulator configuration

This simulator has a server/client format, with dynamics calculations being mainly done on a central server system. Bodies that use perception and perform actions are called ”agents”, and robot and human avatars are available as agents. The previously described perception and communication functions can be enabled by using dedicated C++ APIs to define the actions of agents. Some of the APIs that can be used are listed in Table 1. In the future, we plan to extend the programming beyond just C++ to include interpreter

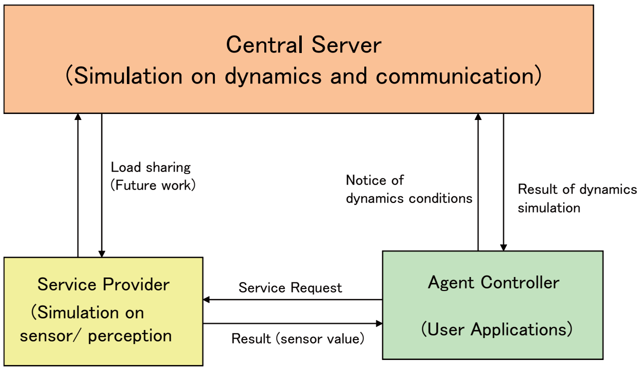

Fig. 3. Software configuration of SIGVerse

Fig. 4. A case in which a user operates avatar agent

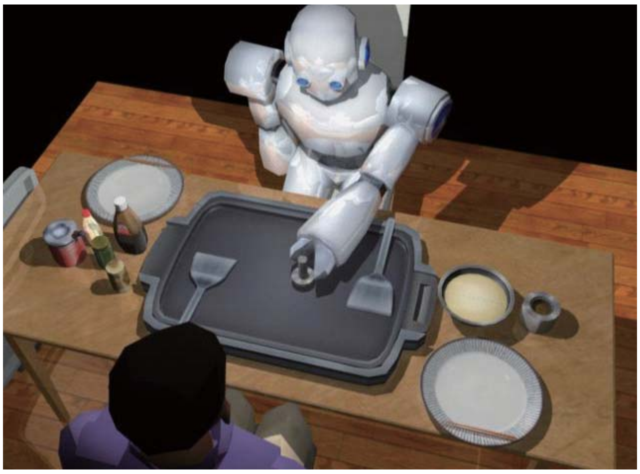

Fig. 5. A case in which a robot helps avatar's task

languages such as Python. The avatars do not just behave as programmed—they can also act on the basis of instructions given to them by operators in real time.

To simulate perception, it is necessary to spread the load so the system is configured to enable calculations not just by the server system but also by individually installed perception simulation servers. More specifically, the module that provides a pixel map of an image to simulate the sense of sight is operated by the perception simulation server, not the central server.

The configuration of the SIGVerse software is shown in Figure 3.

IV. EXAMPLE OF SIGVERSE USE

A feature of SIGVerse is the way in which dynamic calculations, perception simulations, and communication simulations can be done simultaneously. In this section, we describe an example of humans and robots working in partnership to execute a task, and another example of multi-agent system, as examples of applications that fully utilize all three of these functions.

A. Use as evaluation of human-machine cooperation

The objectives of the developers who use this simulation are to determine how to develop the intelligence of a robot that can execute a task in partnership with a human, and how to implement efficient cooperative behavior. The developers created decision and action modules for the robot while adopting various different models and hypotheses, and have confirmed their performance on the simulator. During the simulation, cooperation is required between a real-life human and a robot, which cannot be implemented otherwise without purchasing and developing a life-size humanoid robot. In this simulation, the operator who is in partnership with the robot manipulates an avatar in a virtual environment to reproduce cooperative actions between a user and a humanoid robot. An intelligence module created by the developers uses virtual equivalents of the senses of sight and hearing to comprehend the situation within that space and recognize the state of the user, performs dynamic calculations to control arms, and also simulates communications between the avatar and the robot. Expanding on this kind of usage example will not only further research into simple human-machine cooperation, it will also enable the construction of a research and teaching system with a competitive base for applications such as Robocup[11].

Taking the above application as an example, we implemented a situation in which a human being and a robot cooperate in the task of "cooking okonomiyaki" in SIGVerse ("okonomiyaki" is a popular cook-at-the-table food in Japan, like a thick pancake). Examples of the screens during the execution of this application are shown in Figures 4 and 5. The GUI that the operator can use has buttons such as "flip the pancake", "oil the hotplate", "apply sauce", and "adjust the heat" as shown in Fig.6.

Furthermore, providing immersive interface to the users is very important to conduct realistic phycophisical experiments throught the simulator. Fig.7 shows an example in which the user can operate the cooking devices with haptic interface *PHANToM Omni* to manipulate the "okonomiyaki".

The objective of the task is to cooperate in cooking the okonomiyaki as fast as possible, without burning it. The operator basically uses the GUI to propel the work forward,

TABLE I

LISTS OF AVAILABLE API FUNCTIONS

setJointAngle ($arg1$, $arg2$)	Set the angle of joint $arg1$ be $arg2$
setJointTorque ($arg1$, $arg2$)	Set the torque of joint $arg1$ be $arg2$
getPosition	Get 3D position of the target object
getRawSound	Get audio information cast by other agents
sendRawSound	Utter speech as sound information
sendText($text$, $distance$)	Cast a text message $text$ to agents who are existing within a distance $distance$
captureView	Get pixel map as a visual sensor from eyesight of an agent
detectEntities(arg)	List up all of the agents which is seen by an agent arg
getObj(arg)	Get ID number of an object arg
getObjAttribution($arg1$, $arg2$)	Get attribution value for attribute $arg2$ of an aobject $arg1$

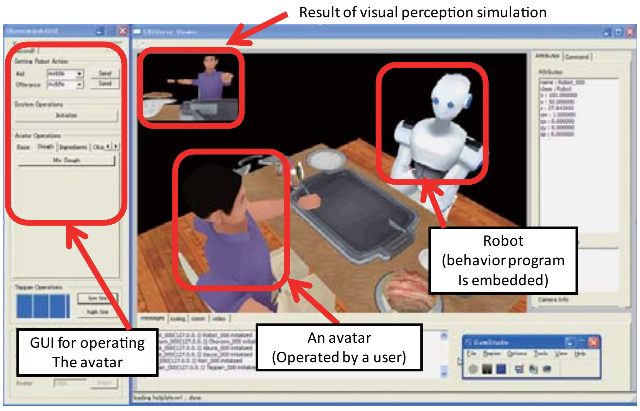

Fig. 6. GUI interface for a cooperation task on SIGVerse

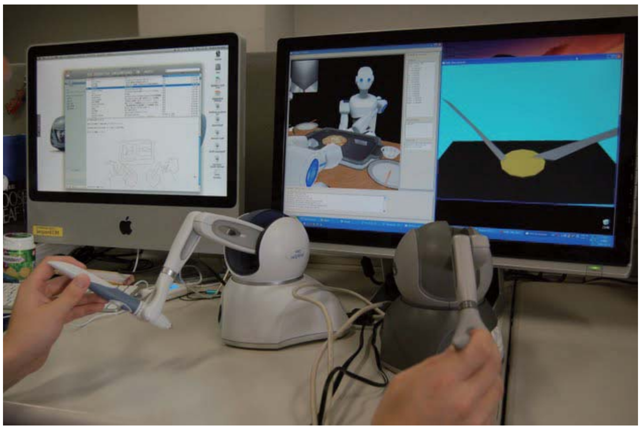

Fig. 7. A haptic interface to handle an "okonomiyaki"

but the robot continuously judges the current situation and, if it considers it can do something in parallel with the work that the operator is doing, asks the operator questions such as "Should I oil the hotplate now?" or "Should I turn the heat down?" It then executes those jobs while viewing the operator's responses. Figure 4 shows a scene in which the avatar in the virtual environment is about to flip the pancake based on the operator's instructions, and Figure 5 shows a scene in which the operator hasn't started any work in particular so the robot is performing the job of oiling the hotplate.

We performed experiments on two cases: one in which the operator performed all of the steps through the GUI, and one in which the robot did suitable parts of the operator's work instead. In the first case in which the operator did all of the work, the task required three minutes 14 seconds before it was finished, but in the second case involving cooperation, the task took one minute 58 seconds to complete. In this manner, it is possible to make effective use of this system as a tool for quantitatively evaluating the human-machine cooperation systems.

B. Use as a multi-agents system that provides embodied perception and communication

Another example is multi agent simulator in which each agent can sense visual and audio perception and can make communication each other. The aim of this multi-agent simulation is to investigate performance of behavior dicision algorithm of hunter agent, which try to chase and hunt target agent. Three hunter agents and one target agent are used for this example as shown in Fig.8.

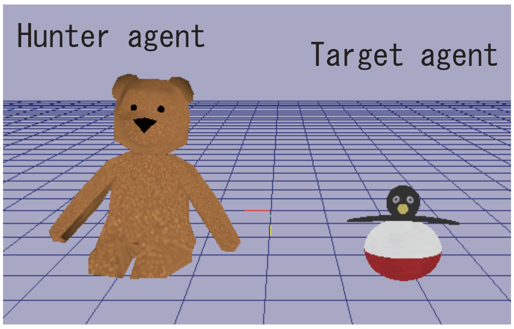

Fig. 8. Hunter and target agent

Only the way for the hunter agents to find the target is vision. The SIGVerse server always checks whether the hunter agents can catch sight of the target agent using `detectEntities` API. In this example, since pattern recognition based on computer vision technique is not the main issue but motion planning is the main issue, the hunter agents use abstracted information provided by the SIGVerse (see section III-B) to recognize the target agent. If the target is hidden by the obstacles such as wall, the hunter agents could not detect the target. If a hunter agent found the target agent even though other hunter agent couldn't find the target for example as shown in Fig.9, the hunter agent which found the target tries to make communication with collaborators using `sendRawSound` API. If the other hunter agents could hear the audio message, they restart to plan a suitable trajectory to chase up the target. Finally, the hunter agents could catch the target as shown in Fig.10.

As well as limitation of the eyesight, audio messages do not reach to other agents which stand faw away. In order to establish effective collaboration with audio communication, the agents should gather together; however, the probability that the hunters find the target would be decreased. For the future plan, we would implement more realistic functions such as 1) A hunter can find the target using sound of footstep even though the target is hidden, 2) Audio messages are not able to be reached to other agents if thick walls stand between the agents, as embodied perception simulation. Complex motion planning tasks which deal with such embodied perceptions can be discussed using the SIGVerse system.

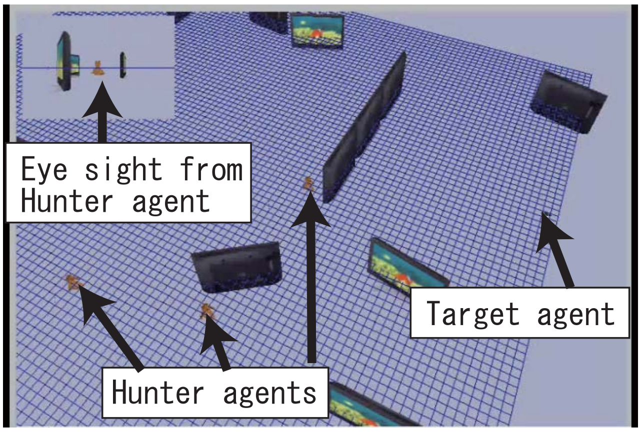

Fig. 9. Hunting scene 1: A hunting environment: the target agent try to escape from the hunters with hiding behind walls. One of the hunter agent found the target using detectEntities API, but other agents could not catch sight of the target. The eyesight shown in the upper left side indicates an image acquired by the hunter agent on the most left side.

V. Conclusion

In this paper, we proposed the concept of a simulation platform in which dynamic simulations of bodies, simulations of senses, and simulations of social communications are integrated into the same system, as an approach to the interdisciplinary research necessary for comprehending the mechanism of intelligence in human beings and robots, and reported on the implementation of a prototype system.

We described one example of cooperation between a human and a robot in the execution of a task, as a specific application of this system, but various other application examples could be considered other than evaluations of the design of intelligence modules for robots. For example, we could reproduce the world of insects by simulating perceptions, to clarify the mechanism of group intelligence in insects from the viewpoints of biology and neurophysiology,

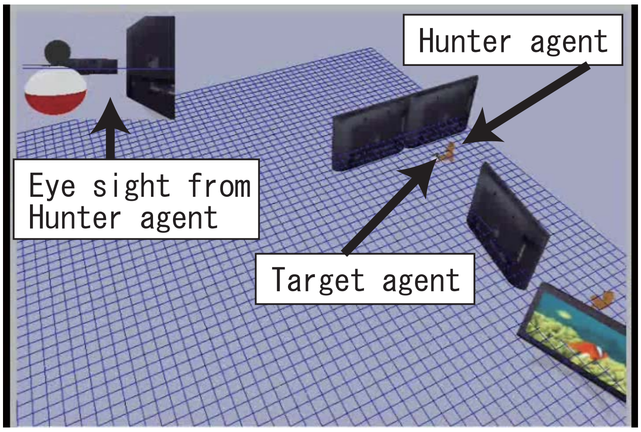

Fig. 10. Hunting scene 2: A hunter agent could catch the target agent with collaboration among other hunter agents using sendRawSound API

and thus enable the evaluation of a model of the codes of conduct of insects, linking together perception information, physical actions, and communications within the group. In the future, we plan to confirm the usefulness of SIGVerse when applied to specific research projects that consider physical problems and social problems simultaneously. Details of the developmental status of our simulator are put up on our webpage periodically, for reference by those with further interest in this field.

Acknowledgements

The authors gratefully acknowledge the research promotion program called *Grand Challenge* supported by National Institute of Information.

References

[1] F. Kanehiro, K. Fujiwara, S. Kajita, K. Yokoi, K. Kaneko, H. Hirukawa, Y. Nakamura, and K. Yamane. Open architecture humanoid robotics platform. In *Proc. of the 2002 IEEE Int'l Conf. on Robotics /& Automation*, pp. 24–30, 2002.

[2] S. Nakaoka, S. Hattori, F. Kanehiro, S. Kajita, and H. Hirukawa. Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms. In *Proceedings of the 2007 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pp. 3641–3647, 2007.

[3] O. Michel. Webots: Professional mobile robot simulation. *Internatoinal Journal of Advanced Robotic Systems*, Vol. 1, No. 1, pp. 39–42, 2004.

[4] B. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th Int'l Conf. on Advanced Robotics*, pp. 317–323, 2003.

[5] J. Jackson. Microsoft robotics studio: A technical introduction. *IEEE Robotics & Automation Magazine*, Vol. 14, No. 4, pp. 82–87, 2007.

[6] K. Ishiguro. Trade liberalization and bureau-pluralism in japan: Two-level game analysis. Technical Report 53, Kobe University Economic Review, 2007.

[7] S. Kirby and J. Hurford. The emergence of linguistic structure: An overview of the iterated learning model. In Angelo Cangelosi and Domenico Parisi, editors, *Simulating the Evolution of Language*, chapter 6, pp. 121–148. Springer Verlag, London, 2002.

[8] R. Smith. Open dynamics engine v0.5 user guide, 2004. http://www.ode.org/.

[9] A. von Kapri, S. Ullrich, B. Brandherm, and H. Prendinger. Global lab: an interaction, simulation, and experimentation platform based on "second life" and "opensimulator". In *Proc. Pacific-Rim Symp on Image and Video Technology (PSIVT'09)*, 2009.

[10] J. Magnus and H. Verhagen. Massively multiple online role playing games as normative multiagent systems. In Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, editors, *Normative Multi-Agent Systems*, No. 09121 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany.

[11] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawai, and H. Matsubara. Robocup: A challenge problem for ai and robotics. In *Lecture Notes in Computer Science*, Vol. 1395, pp. 1–19, 1998.