A Hybrid Agent Simulation System of Rescue Simulation and USARSim Simulations from Going to Fire-Escape Doors to Evacuation to Shelters

Masaru Okaya, Shigeru Yotsukura, and Tomoichi Takahashi

Meijo University, Tenpaku, Nagoya, 468-8501, Japan
{e0427566,e0527080}@ccalumni.meijo-u.ac.jp, ttaka@ccmfs.meijo-u.ac.jp

Abstract. Disaster & rescue simulations handle complex social issues, the macro level modeling of which is difficult. Agent-based social simulation provides a platform to simulate such social issues. It is ideal that the simulations cover various evacuation patterns and the results are used to make effective plans against disasters. This requires that the behaviors of a numbers of heterogeneous agents are simulated at urban size areas in hostile environments. Representing all buildings of the area by 3D model requires a large amount of computer resources and computing the behaviors of a number of agents takes a lot of computation time. These make it difficult to simulate rescue behaviors at disasters in real scale.

We propose a hybrid agent simulation system that switches systems that is suitable for situations during simulations. A hybrid system of two simulations with different time and space resolution makes it possible to simulate urban size human behaviors and indoor movements with less computational resources than doing by one system. This paper presents protocols that connect two systems that are used in RoboCup Rescue Simulation League, Rescue Agent Simulation and USARSim. The prototype system provides a simulation of people's evacuation from going to fire-escape doors to moving to shelters.

1 Introduction

The Great Hanshin-Awaji earthquake of 1995 led researchers to apply their technologies for decreasing damages from disasters. Subsequent disasters including the 9/11 on the World Trade Center of 2001, 2004 Indian Ocean earthquake, and 2008 Sichuan earthquake China have driven to start disaster & rescue related projects around the world. Several systems that support decision of rescue operations or prompt planning for disaster mitigation have been presented. Their functions are to ensure prompt planning for disaster mitigation, risk management, and support of IT infrastructures at disasters [1][4].

In RoboCup, the rescue agent competition league has started since 2001 using RoboCup Rescue Simulation (RCRS). RCRS was designed to simulate the rescue operations and disasters simultaneously at the Hanshin-Awaji earthquake disaster. In competitions, rescue agents contest their performances at various disasters

J. Baltes et al. (Eds.): RoboCup 2009, LNAI 5949, pp. 414–424, 2010.

situations on various cities. At 2005, virtual robot competition has started [5]. USARSim is a high fidelity simulator based on the Unreal Tournament game engine and has provided environments to develop models of new robotic platforms, sensors and test environments and to develop control algorithms that are seamlessly migrated to systems in fields [6]. Commander training systems or other simulation systems have been presented using the RoboCup system [7][3][8].

The last disasters have made the purposes of disaster & rescue simulations more clear. It includes the simulations are used as emergency management system of local governments and the disaster & rescue simulations need more functions to that end. For example, when disasters occur, an urban size simulation is used to deploy rescue agents at the first stage of rescue operations. After the agents arrive at sites, simulations of inside devastated houses are useful to search victims.

It requires a huge amount of computation power and resources to simulate the behavior of agents at wide area with fine resolutions. We propose a hybrid agent simulation system of RCRS and USARSim. USARSim simulates people's evacuation from going to fire-escape doors with fine resolution and RCRS simulates the behaviors of moving to shelters. Section 2 describes rescue scenarios using multi agent simulation systems (MAS). A framework of hybrid system that executes evacuation systems is described in section 3. Protocols to connect two MAS and to support communication among agents at different MASs are described in section 4. Section 5 shows the simulation results of our prototype system. The summary of our proposal and discussions are described in Section 6.

2 Rescue Scenarios Simulated by MAS

Disaster & rescue simulations handle complex social issues, the macro level modeling of which is difficult. MAS is good to simulate such issues and it is ideal that disaster & rescue simulations can simulate various evacuation patterns and rich human interactions to make effective plans against disasters. It requires followings, (1) simulation of behavior of a numbers of heterogeneous agents, (2) at building inside and urban size areas, (3) under hostile environments caused by disasters, (4) with interactions of others including rescue operations.

Table 1 shows one of rescue scenarios when people evacuate from buildings. The scenario consists of three stages.

A. *the initial stage of disasters:* People in buildings try to evacuate from houses. Rescue teams rush to devastated houses.
B. *rescue operations at devastated houses:* The rescuers execute their actions to save lives, fight fire and do related actions. They use robots to search and rescue victims from the houses.
C. *evacuation to shelters:* People who get out of the houses evacuate to shelters. Rescue headquarters allocate shelters and announce rescue teams.

The rescue scenario contains indoor and outdoor environments. Indoor or open space people behavior are simulated by free space model and the traffic of outside

Table 1. Rescue scenario where agents go in and out of buildings

simulation scenario		outside		inside
stage	behaviors of agents	RCRS		USARSim
A.	When disasters occur,	A, B		x, y, z
1	people evacuate from buildings,	X	⇐	x
2	call for help from inside house to rescue teams.	A	←	y
3	People inside buildings help and communicate each other.			y, z
4	Rescuers rush to the sites according to their headquarters.	A, B		
B. 1	The rescuers arrive at devastated houses.			
2a	Some rescue teams start fire-fighting.	A		
2b	Others start searching by robots,	B	→	r
3b	confirm conditions of rescue operations in the houses,	B	←	y, z
4b	execute search-and-rescue operations.	B	⇒	y, z
5	They enter the houses.	B	⇒	b
6	They communicate each other in houses,	(B	↔	b), y, z
7	or colleagues outside.	A	↔	b
C. 8	All move to outside and evacuate to refuges.	B, Y, Z	⇐	b, y, z

Capital letter and low case letter represent agents in RCRS and USARSim, respectively.
⇐,⇒ show agents' transfer to the other system,
←, → represents communication among agents.

movements are simulated on a road network. Representing all buildings in three dimensional (3D) models and computing the behaviors of a number of agents require a large amount of computer resources and take a lot of computation time.

Our hybrid system can simulate (1) behaviors of indoor environments by USARSim, (2) the evacuation behaviors after exiting the buildings or going to refuges by RCRS. Figure 1 shows our idea of combining RCRS and USARSim. The two systems are agent based systems (ABS) with different resolutions of space and time. RCRS handled two dimensional (2D) urban size simulations of disaster & rescue operations and USARSim handles rescue robot motions at 3D buildings. Table 2 shows properties of RCRS and USARSim.

3 A Hybrid System to Execute Evacuation Scenarios

3.1 Requirements for a Hybrid Agent System

Followings are the RCRS commands of rescue agents to move into a building or to do rescue operations in it.

`AK_MOVE`**:** An agent submits it to enter into a building.
`AK_EXTINGUSH`**:** A firefighter submits it to extinguish fires.
`AK_RESCUE`**:** An ambulance team submits it to rescue a buried humanoid.
`AK_LOAD`**:** An ambulance team submits it to load an injured humanoid.

These commands are executed in one cycle of RCRS, and they don't reflect the facts that rescue operations change according to inside disaster situations. It

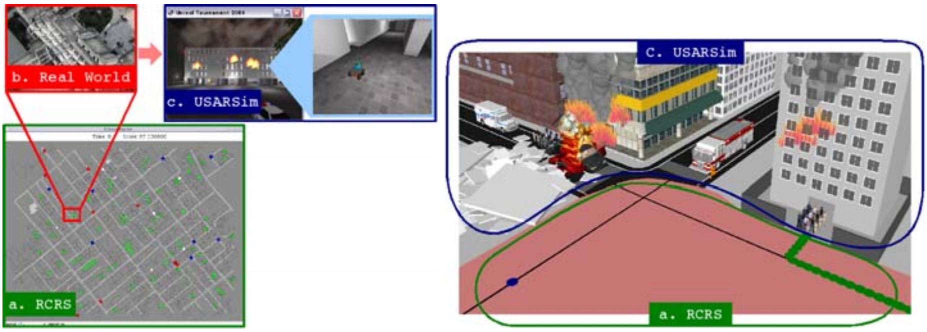

Fig. 1. Image of simulation systems that cover from wide rage disasters and local devastated houses. (The left shows the snapshots of RCRS simulation (a), a video picture of a building (b), and USARSim simulation of the building (c). The right shows an image of hybrid system. RCRS simulates 2D world, USARSim simulates 3D world, respectively.)

Table 2. Comparison of two rescue simulations

items	RCRS	USARSim
purposes	planning of disaster prevention, verification of rescue plans	providing platforms to develop rescue robot
agent type	civilian, rescuer humans	rescue robot
agent number	$O(100)$	$O(10)$
area size	$O(km^2)$	$O(\text{house size})$
disaster simulations	fire, collapse building	–
simulation time	72 hours	real time
map model	2D network*	3D model

*: Dwelling environments vary from country to country. For example, our town has 135,000 and 120,000 inhabitants at daytime and night respectively, 70,000 households, and is $21.6km^2$. The 2D road network has 6,000 nodes and 4,000 edges.

is difficult to simulate the rescue operations inside houses at the resolution of RCRS. We employ USARSim in order to simulate the indoor rescue operations in more detailed way than RCRS. USARSim also simulates the inside evacuation behaviors of agents.

USARSim is a real-time simulator and RCRS was originally designed to simulate situations of 72 hours after earthquakes occur within a specified time. Followings are required to reflect the simulation results of USARSim and RCRS each other.

– *synchronization between simulation systems with different Scales:*
  The progress of simulation is paced by wallclock time [2]. The simulation step is mapped to wallclock time by the following formula.

$$T^s_{present} = T^s_{start} + Scale \times (T^w_{present} - T^w_{start}) \tag{1}$$

where T^s, T^w are simulation time and wallclock time, respectively. $Scale$ is a factor that shows how fast or slow the simulation advances the wallclock time, for example, $Scale = 2$ indicates that the simulator runs twice as fast as wallclock time.

- *communications among agents that are in different ABSs:*
  Rescue agents communicate each other to cooperate with properly. When some rescue teams enter to devastated buildings, they report the inside situations to their commanders outside or receive orders from them. Communications among agents are supported whether the agents are in RCRS or in USARSim.
- *management of agents when they move to a different ABS:*
  The kernel and USARSim server manage the status of which agents are connected to themselves. When the agents enter to or exit from a building, they switch connection from RCSR to USARSim or vice verse, respectively. The servers change the data of the connected agents.
- *reflection of a disaster to other ABSs:*
  RCRS simulates disasters such as aftershocks or fire, and rescue actions such as firefighting. These change the situations of houses where USARSim provides to its agents. Reflecting these changes to USARSim makes USARSim environments dynamic ones.

3.2 Protocols Systems for Hybrid System

New protocols are designed to connect RCRS and USARSim and to enable agents to switch servers. Agents consist of parts that connect RCRS and USARSim servers. They use following protocols in addition to the original ones. Table 3 shows the protocols newly added.

AK/KA. Prefix A stands for agent and K for kernel of RCRS. For example, when an agent is in front of building of RCRS, the agent can switch connection from RCRS to USARSim by `AK_USASIM_ENTER`. And they enter the corresponding 3D building of USARSim.

KU/UK. Prefix U stands for USARSim Controller. USARSim Controller spawns USARSim Client. The clients are connected to a USARSim server that supports the 3D environments. Commands with this prefix have a role to bridge two systems. The kernel submit `KU_connect` [1] command with the building ID to USARSim Controller that sets up USARSim for the building.

AU/UA. `AU_TELL` and `UA_HEAR` commands serve communications among clients in the environments created by one USARSim Server. `AU` commands are submitted and received at every USARSim time step.

Figure 2 shows architecture of agents and the hybrid system.

- USARSim is connected to RCRS as one of simulators. When agents move from RCRS to USARSim, they submit `AK` commands and the kernel passes it

[1] `KU` commands are implemented in the operand part of the RCRS `commands`.

Table 3. Added Protocols to bridge RCRS and USARSim

Commands	specification	
Switch systems from RCRS to USARSim, vice versa		
AK_USARSIM_ENTER	A → K	An agent submits it to enter a building (to switch a server to USARSim) at its entrance node.
AK_USARSIM_EXIT	A → K	An agent submits it to get out of the building (to switch a server to RCRS) near its exits.
KA_USARSIM_ENTER_OK KA_USARSIM_ENTER_ERROR	K → A	The kernel notify whether USARSim connection succeeds or not.
KA_USARSIM_EXIT_OK KA_USARSIM_EXIT_ERROR	K → A	The kernel notify whether USARSim disconnection succeeds or not.
UK_ENTER_OK UK_ENTER_ERROR	U → K	USARSim Controller returns IP address and port number that USARSim Client uses when connection succeeds, errors why it fails,
UK_EXIT_OK UK_EXIT_ERROR	U → K	USARSim Controller returns `OK` when that the agent could disconnect to USARSim, `ERROR` and the reason that the agent cannot disconnect to USARSim.
Control the corresponding USARSim object		
AU_FORWARD, AU_BACKWARD	A → UC	The agent moves the object forward/backward.
AU_LEFT,AU_RIGHT	A → UC	The agent turns the object left/right.
AU_STOP	A → UC	The agent stops the object in USARSim.
AU_MOVE	A → UC	The agent moves the object to specified position in USARSim.
Communication between agents		
AK_TELL KA_HEAR	A → K K → A	They are basically the same as RCRS. They also support communications to USARSim or within via Communication Center.
AU_TELL	A → UC	The agent sends messages to other agents via Communication Center.
UA_HEAR	UC → A	Communication Center sends received messages to all agents that are in USARSim.
UK_HEAR	UC → A	Communication Center sends a list of received messages at the step of the kernel.

A, K, U, and UC represent agent, kernel, USARSim Controller, and USARSim Client, respectively. UC→ A shows command flow from USARSim Client to agents.

to USARSim Controller. After USARSim Controller receives it, the connection between the kernel and USARSim Controller and the communication are the same way as other sub simulators.

- USARSim Servers supply 3D simulation environments of buildings to the agents. The buildings in RCRS and USARSim are linked with the same ID number. USARSim Servers are set for every building and USARSim Controller supervises these servers.
- While agents connect to USARSim, the agents control the corresponding objects (avatar) in USARSim with `AU` commands. The commands express their wills how the avatars behave in the USARSim world. The behavior of avatars is simulated by a physical engine of USARSim.

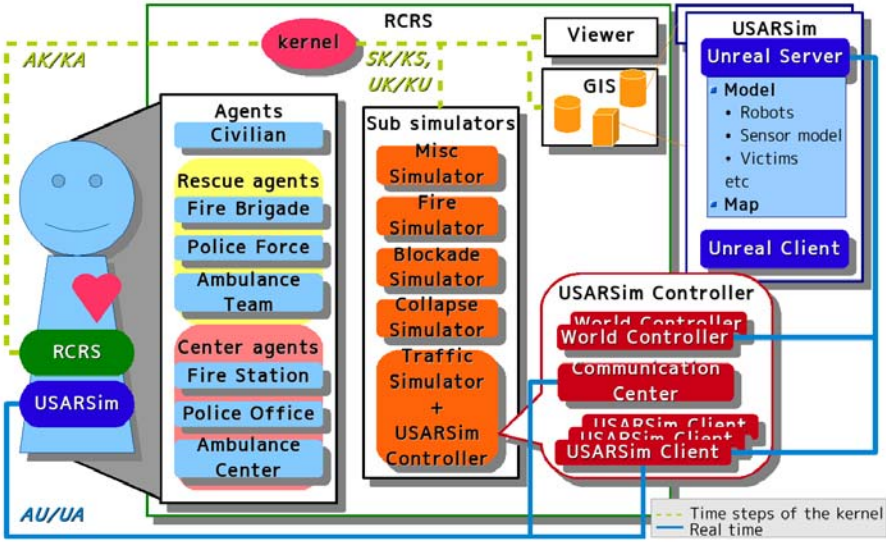

Fig. 2. System architecture of hybrid system connected by protocols. The left human figure shows that agent consists of parts to connect RCRS and USARSim.

4 Management of Changing ABS and Communication among Agents in Different ABSs

4.1 Protocols Systems for Changing ABS

Figure 3 shows a timing chart when an agent enters into a building and a sequence of protocols associated with it. The columns below Agents, kernel USARSim Controller show time steps. Time `a, c, e, f` correspond to RCRS time points and `b, d` correspond to USARSim time points.

a : When an agent arrives at an entrance node of a building, it submits `AK_USARSIM_ENTER` to enter the building. Receiving the command from the kernel, USARSim controller
  1. when this is the first entry to the building, spawn USARSim server that maintains the corresponding 3D USARSim world according to entries of its configuration file,
  2. place a corresponding avatar in the USARSim world.

c : The agent switches connection from RCRS to USARSim, when it receives `AK_USARSIM_ENTER_OK` commands with a port number to communicate with USARSim Client.

d : The agent submits `AU` commands. The commands control its avatar to move or rescue in the building of USARSim.

e : When the agent get out of the building, it submits `AK_USARSIM_EXIT` near the entrance of the building.

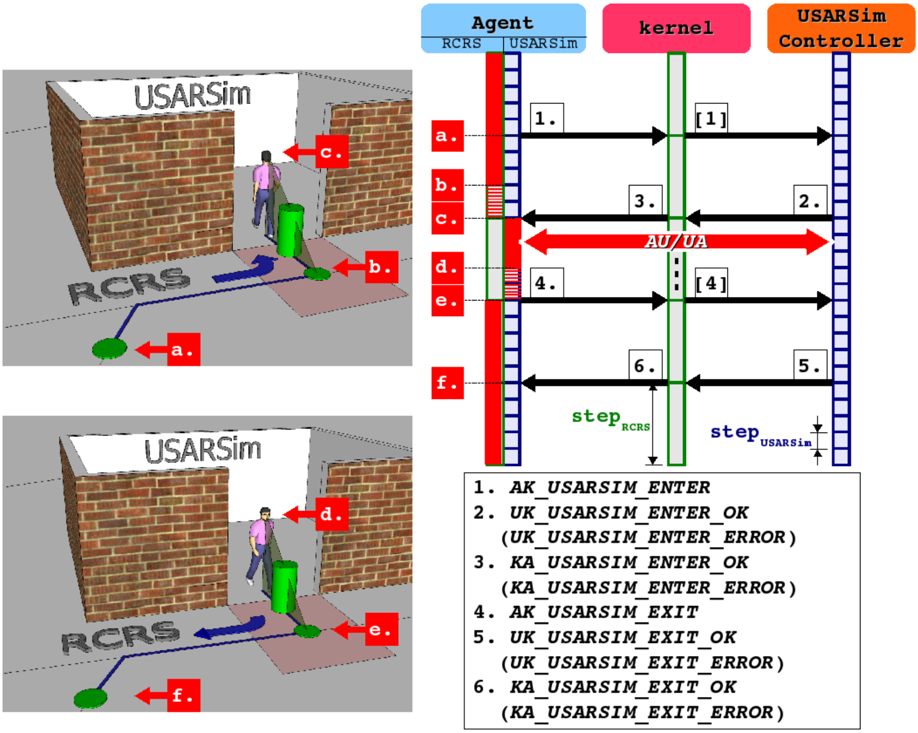

Fig. 3. Time chart for agent management change(The left shows the movements of agent. The middle is time sequence, and the slots show the time steps of RCRS and USARSim. The right one is the sequence of commands. Commands in parenthesis are ones that are embodied in operands of other protocols.)

f : Receiving `AK_USARSIM_EXIT`, USARSim deletes the avatar and returns `AK_USARSIM_EXIT_OK`. Receiving the `AK_USARSIM_EXIT_OK`, the agent returns to RCRS world and it position is the entrance node of the building.

4.2 Protocols for Communication to Agent in ABSs

Changing the connecting server requires to support communications between agents that are in different servers. The scope of commutation expands from within one ABS to between different ABSs. The rescue simulation scenario of Table 1 contains three patterns of communications.

1. *communication within RCRS:* A.4 is the same pattern of communication as the original RCRS. Communications of C.8 are between agents in RCRS. However, Y and Z are USARSim agents initially at USARSim.
2. *communication within USARSim:* Conversely, A.3 is communication between agents, y and z, that are USARSim agents initially. B.6 is communication to b that is a RCRS agent initially.

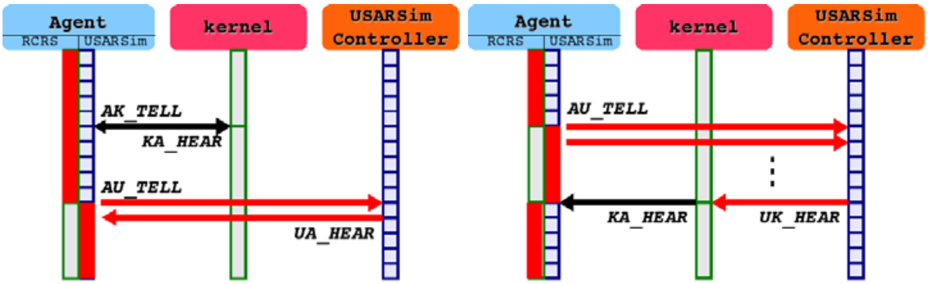

Fig. 4. Time chart for agent communication between two ABSs, RCRS and USARSim. The ratio of *Scales* is 8 in this figure.

3. *communication between RCRS and USARSim:* A.2 is communication between one agent in RCRS and the other in USARSim. They are initially in RCRS and USARSim, respectively. In a case of B.7, the agent, b, is initially in RCRS. The agent A communicates with B without knowing whether the agent B is in RCRS or in USARSim.

Protocols support the communications among agents that are different systems. The left of Figure 4 shows a sequence of protocols communication within one system. The upper diagram corresponds to communication within RCRS (pattern 1), and the lower diagram is within USARSim (pattern 2). The right of Figure 4 shows a sequence of protocols that agents in RCRS and USARSim communicate. Agents in USARSim tell at USARSim time step, and agents in RCRS hear at RCRS time step.

5 Evacuation Simulation from Indoor to Outdoor Shelters

A subset of the rescue scenario in Table 1 is simulated. Figure 5 are snapshots of simulations by our prototype system. The simulation conditions are followings:

situation: Fires occur at a university campus. Students in school buildings evacuate to an outdoor refuge.

map: The road network and buildings of the university campus are presented by 2D RCRS map. Two of the school buildings are linked to USARSim and represented by its 3D models.

agent: Six and four student agents[2] are at the two buildings respectively, and total 150 agents evacuate to one refuge by following the instructions of teacher agents. There are ten teacher agents instruct the students evacuation routes.

Two PCs are used in our experiment system. One PC is Core2 Duo of 2.2GHz with 2GB Memory and the other one is Pentium 4 of 3GHz with 1GB Memory.

[2] Student and teacher agents are the civilian and police agents of RCRS.

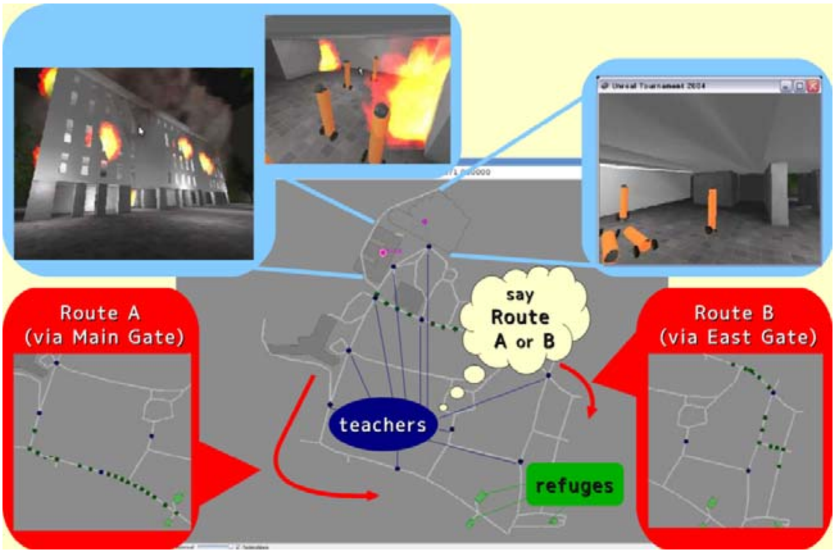

Fig. 5. Execution of evacuation scenario (Two buildings are linked to USARSim, cylinder figure avatars are student agents.)

RCRS and one USARSim server are run at the first PC and the other USARSim server is run on the second PC. The two USARSim servers take charge of the buildings that are linked to RCRS. The ratio of *Scale*s, USARSim to RCRS, is set to 60 (one simulation time of RCRS corresponds to one minute).

The upper three figures are snapshots of USARSim. The left one is an exterior view of one school building and right two figures display the behavior of students inside the buildings. The student avatars in USARSim are represented by a cylinder figure robots. The figures of left bottom and right bottom show the results of simulations which the teachers instructed different routes.

The results show that all ten students inside get out of the building and evacuate to the refuge. They show different patterns according to the teachers' instructions. These indicate communications among agents work well.

6 Discussion and Summary

It is ideal that disaster & rescue simulations handle rescue behaviors at disasters in real scale. This requires simulations of a huge number of agents that behave at wide areas with fine resolutions. The requirements lead a huge of computer resources and powers.

We propose an idea of switching systems during simulations to choose one that is suitable for situations. A hybrid system of two different simulations with different time and space resolution makes it possible to simulate urban size human behaviors and indoor movements by reasonable computational resources.

This paper presents protocols that connect two systems that are used in RoboCup Rescue Simulation League, Rescue Agent Simulation and USARSim. The prototype system can simulate people's evacuation from going to inside fire-escape doors to moving to outside shelters. The result shows an possibility that the hybrid system make the simulations feasible ones that takes a lot of computer recourses.

References

1. Van de Walle, B., Turoff, M. (eds.): Emergency response information systems: Emerging trends and technologies. Communications of the ACM 50(3), 28–65 (2007)
2. Fujimoto, R.M.: Parallel and Distributed Simulation Systems. John Wiley & Sons, Inc., Chichester (2000)
3. Kuwata, Y., Takahashi, T., Ito, N., Takeuchi, I.: Design of human-in-the-loop agent simulation for disaster simulation systems. In: Proc. SRMED 2006 (Third International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster), pp. 9–14 (2006)
4. Mehrotra, S., Znatri, T., Thompson, W.(eds.): Crisis management. IEEE Internet Computing 12(1), 14–54 (2008)
5. RoboCupRescue, `http://www.robocuprescue.org/`
6. Wang, J., Balakirsky, S., Carpin, S., Lewis, M., Scrapper, C.: Bridging the gap between simulation and reality in urban search and rescue. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X. LNCS (LNAI), vol. 4434, pp. 1–12. Springer, Heidelberg (2007)
7. Schurr, N., Marecki, J., Kasinadhuni, N., Tambe, M., Lewis, J.P., Scerri, P.: The defacto system for human omnipresence to coordinate agent teams: The future of disaster response. In: AAMAS 2005, pp. 1229–1230 (2005)
8. Takeuchi, I., Kakumoto, S., Goto, Y.: Towards an integrated earthquake disaster simulation system. In: First International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster (2003), `http://www.dis.uniroma1.it/~rescue/events/padova03/papers/index.html`