Design of RoboCup-Rescue Viewers – Towards a Real World Emergency System –

Yoshitaka Kuwata[1] and Atsushi Shinjoh[2]

[1] NTT DATA CORPORATION , JAPAN.
E-mail: kuwatay@nttdata.co.jp
[2] International Academy of Media Arts and Sciences, JAPAN.
E-mail: kaminari@iamas.ac.jp

Abstract. In this paper the design and implementation of a prototype viewer system for the RoboCup-Rescue framework (Version 0) is described. We discuss the requirements for the visualization of disaster information and propose a workflow model for the disaster mitigation process. A prototype of disaster information system has been built, based on this model, and here we discuss the system from the viewpoint of information visualizations.

1 Introduction

The RoboCup-Rescue project [3] was started by researchers in fields such as robotics and artificial intelligence. It is a grand challenge for disaster mitigation; the final goal is to build teams of rescue robots for disaster mitigation, and autonomous robot rescue teams are expected to be cooperating with human rescue teams by the year 2050. The project is divided into four parts: a simulation project, a rescue robot project, a system integration project, and an operation project. The goal of the simulation project is to design a set of software agents which model the various actors in the disaster mitigation process (rescue teams, fire-fighters, police, supervisors in fire department, and so forth.) Many kinds of simulators should be integrated as the bases of a multi-agent framework. For example, a traffic simulator, a fire-spreading simulator, and a building-collapse simulator, for example, should be working together. A prototype RoboCup-Rescue framework, called version 0, is under development by the RoboCup-Rescue technical committee. It is currently public to the RoboCup-Rescue community. The first RoboCup-Rescue competition is scheduled in August, 2001.

The importance of collecting disaster information and using it effectively was widely recognized in the aftermath of the great Hanshin-Awaji earthquake. In the early stages of the disaster mitigation process, the local emergency office could not control the rescue teams efficiently because the office did not have adequate information. The Japanese government now assists the local governments in the preparation of disaster information systems. [2]

The technologies and strategies developed in the RoboCup-Rescue framework are expected to be used in practical applications of disaster information systems in local emergency offices. Visualization techniques for disaster information are one of the most important parts of the mitigation systems on which humans rely.

P. Stone, T. Balch, and G. Kraetzschmar (Eds.): RoboCup 2000, LNAI 2019, pp. 159-167, 2001.

In this paper we describe the design of RoboCup-Rescue (Version 0) and the architecture of the viewers as well as the requirements for the visualization of disaster information. Our analysis of the requirements is based on interviews of rescue teams and supervisors in emergency offices. This is because these requirements depend on the workflow of the mitigation process. A prototype system for the visualization of disaster information system has been based on this analysis.

2 Architecture of RoboCup-Rescue Framework

The architecture of RoboCup-Rescue framework (version 0) is described in detail in the simulator manual [5]. Figure 1 represents the architecture of RoboCup-Rescue framework in brief.

The architecture is based on the design of RoboCup-Soccer simulator. Thus, they are very similar except a few points; simulators and a Geographic Information System (GIS). As RoboCup-Rescue framework need to cooperate with many kinds of physical simulators which will be developed independently, protocols to exchange information between simulators and agents are defined. As geographic information is essential for both simulators and agents in this domain, a separate GIS is introduced.

The core of the framework is a kernel, that manages the status of the simulation world. It also takes charge of the communications between simulators and agents. A GIS component provides basic data such as the locations of buildings and road networks. It also provides the initial status of each agent. Several independent simulators, as well as several kinds of agents, are connected to the kernel. A traffic simulator, a fire-spreading simulator, a building-collapse simulator, a road-blockade simulator are currently available. One important goal for RoboCup-Rescue framework is to design a integrated disaster simulator by combining several independent simulators.

The other important goal is the modeling and design of agents. For example, action plans for rescue teams are developed and built into agents and tested in the framework. Good action plans are useful in practice. There are four kinds of agents exist in the RoboCup-Rescue framework. Fire brigade agents, police agents and rescue agents are introduced for the emergency response. Civilian agents represent victims.

The expected uses of the RoboCup-Rescue framework are the following:

1. Development of mitigation plans
   The mitigation plans should be made before disasters occur. With RoboCup-Rescue simulators, specialists can check how their mitigation plan works. Urban planners, for instance, can run the simulators with or without fireproof buildings in their new city.
2. Disaster mitigation process
   Viewers in emergency offices and support of rescue teams' communications are useful. We will discuss this issue in section 5 in details.

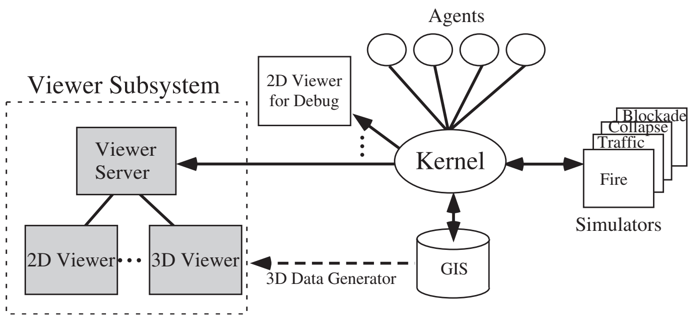

Fig. 1. An Architecture of RoboCup-Rescue Framework

3. Development of the simulation systems
   Viewers are used in the development of simulators and agents. The designers of agent programs use viewers to verify their strategies. Researchers of disaster try their simulators with other simulators in order to check the complex results.
   Detailed information is required for these purposes. For instance, the designer of rescue agents need to know exactly why their agents failed to rescue people in a burning house.
4. Demonstration
   As in RoboCup-Soccer, RoboCup-Rescue competitions are planned. The main viewer (a 3D viewer) will be the most important component. It will be used to show the progress of the rescue competitions to the audience. It is very hard for audience to understand the progress and result of simulations. Thus, visualization should be intuitive, easy to understand, and attractive.

Two kinds of viewers are currently under development. One is a plain 2D viewer designed for researchers and developers. The other is a 3D viewer suitable for supervisors at emergency offices. The 3D viewer, also called the main viewer, provides a perspective view of the simulation world.

3 Design of Viewers for the RoboCup-Rescue Framework

3.1 3D main viewer

Like the main viewer in the RoboCup-Soccer Simulator, the main viewer for the RoboCup-Rescue framework (version 0) is based on client-server architecture. The server manages the information displayed in the clients according to the capability of the clients and the capacity of communication channel. Clients can thus be located on remote machines in distant locations. The viewer has an automated viewpoint movement built in, so that view points jump to the places where interesting events occur.

Fig. 2. Snapshot of a 3D data image of part of Nagata-Ward, Kobe(left) and a snapshot of 3D viewer(right) : The area shown is about 1.5 km by 1.5 km. As there are about 10,000 building in the area, the 3D viewer need to handle more than 50,000 polygons. Buildings are displayed as transparent objects in 3D viewer.

As the GIS data is extremely large, we supply it to a 3D viewer client directly, not through the kernel; static information such as the shapes of buildings and roads are extracted from GIS and converted into 3D objects beforehand. On the other hand, dynamic information such as location of agents and the status of roads and buildings, are supplied by the kernel.

Figure 2 is a snapshot of a 3D data image of part of Nagata-ward, Kobe-area[3].

3.2 2D viewer

The 2D viewer, we designed for development support, can be used as a viewer which connects to the kernel via a TCP/IP socket. It is also a log viewer that can be used to read the kernel execution log file and redisplay the simulation. This is useful in analyzing the performance of simulators and agents. As the 2D viewer is written in Java, it can be executed on any machine which supports Java virtual machine(JVM).

A snapshot of 2D viewer is shown as figure 3. The main window of the 2D viewer is a map with objects such as buildings, roads, and agents. It has zoom in/out, and scroll functions. The status of buildings and roads are displayed as different colors; i.e. yellow for heating houses, red for burning houses, and so on. Agents are represented as small icons. The objects are mouse clickable; when users click objects of the map, the detailed information associated with the objects is displayed in the information area.

[3] Kobe is selected for the first trial mainly because we have a lot of real data from the great earthquake, which can be used to compare with the results of the simulation

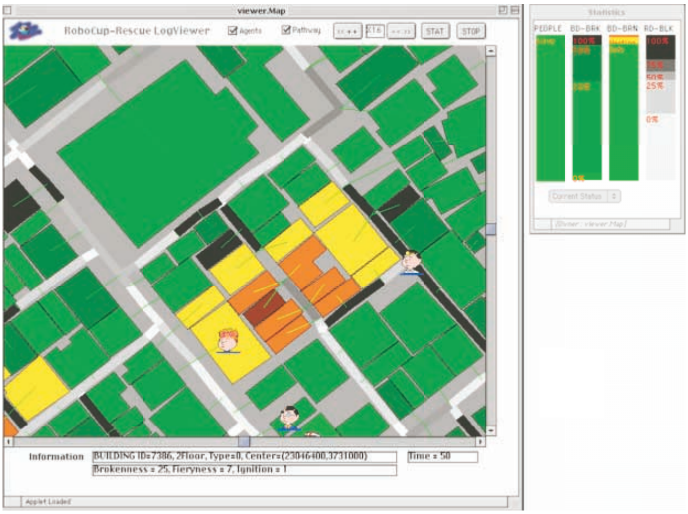

Fig. 3. Snapshot of the 2D Viewer

A small window on the top right corner is used to display statistic information. The percentage of life/death, building corruption, road blockade, and the status of fire spread are presented by small bar graphs with color.

4 Model of Workflow in the Mitigation Process

To apply the RoboCup-Rescue framework to the real world rescue operations, we started building a workflow model[4] of ideal mitigation process for the next-generation disaster response system. We based this model on the results obtained by several rescue teams and supervisors. The model is shown in Figure 4. The workflow model is divided into four steps: (1) Data Acquisition, (2) Data Analysis, (3) Decision Making, and (4) Command-and-Control. The process starts with the collection of disaster information. Perspectives views from cameras on helicopters and from surveillance cameras on the tops of tall buildings are used for collecting data.

To understand the situation quickly, we can use image analysis in step 2. In step 3, human supervisors make decisions based on the extracted information which includes the type of the disaster, the location and scale of the affected area, and the severity of the damage. In step 4, the rescue teams in the affected areas are informed of these decisions by way of the command-and-control support system. The rescue teams also collect information in the affected areas and send reports to the emergency office. Thus these steps makes a workflow process loop.

[4] By "workflow", we mean a series of operations which required in mitigation process.

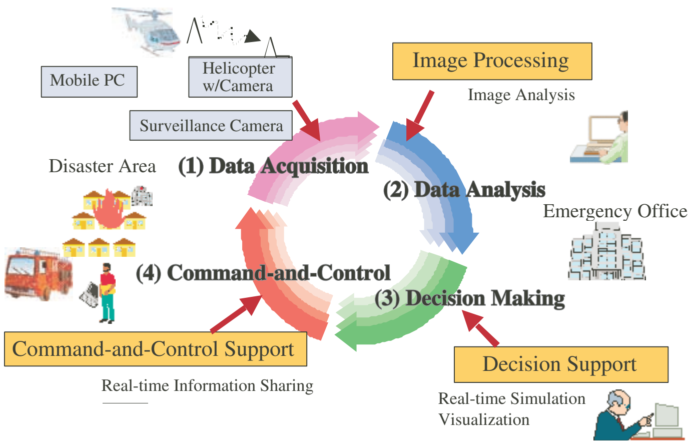

Fig. 4. Workflow model of the mitigation process.

The techniques and strategies developed in RoboCup-Rescue simulator will be used in both the decision-making process and the command-and-control process. At the emergency office, visualization is used to help supervisors make decisions. It is also used by rescue teams for sharing data within the teams and with emergency office.

The prototype systems consists of four components that corresponds to the steps in the workflow process. Figure 5 represents the decision-support system and command-and-control system which support step (3) and step (4).

The decision-support system is used by supervisors at an emergency office, and the command-and-control systems are for rescue teams. These two kinds of systems are designed to work together; all of the information is managed with location and time. A change of a piece of information in one system is transmitted to the other systems. If a member of a rescue team with a command-and-control terminal changes his location, for instance, other members of rescue team as well as the supervisors at the emergency office know it by using these systems.

Because the roles of the emergency office and rescue teams are different, their visualization requirements are different. In the following sections, we discuss two kinds of requirements.

5 Visualization Requirements

5.1 Visualization for Resource Management

The management of rescue resources is the most important of the emergency office's responsibilities. Supervisors in emergency offices need to assign rescue resources to each affected area. In the daily routine work, for example, they dispatch fire engines, ambulances, and rescue teams to areas in response to phone

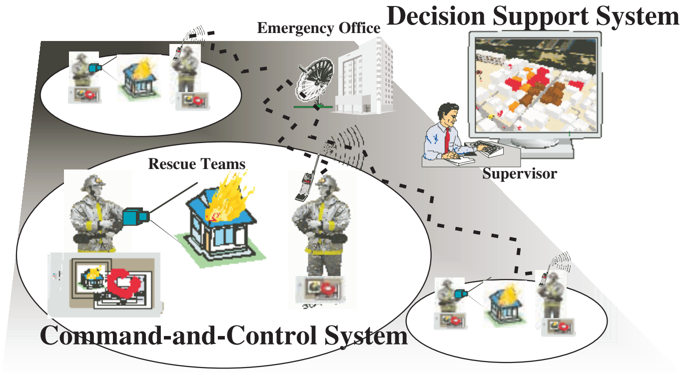

Fig. 5. Prototype system for decision support and for command-and-control: For decision support at an emergency office, 3D visualization is used to gain a quick understanding of the situation, for command-and-control support, wearable systems with touch-panel displays, CCD video cameras, wireless communication channels, and map-based shared white boards are integrated for communication support.

reports. The process becomes more difficult in the case of a big disaster, for which not enough rescue resources are available and the priority of each case need to be determined.

The following requirement list is obtained from the interview of resource managers.

- Intuitive Understanding of Global Information
  Because supervisors need to manage a big area, such as a whole city, their decisions must be based on a global view of resources and damage.
- Estimation and Prediction of Damage
  In the early stages of mitigation efforts, almost no information is available. Emergency offices thus need to estimate the damage for the decision on their first actions. The prediction of damage is also important for the middle term (around 72 hours) planning in big disasters.
- Trial-and-error simulation
  There are two phases of decision support, and the first is to make mitigation plans beforehand. Supervisors can base a decision on these mitigation plans. Building mitigation plans require trial-and-error simulation. Because the visualization of disaster response systems will be used in both phases, it also needs to support trail-and-error.
- Real-time Visualization
  In the execution of mitigation plans, live information is collected and should be displayed. Intuitive and real-time visualization is required.

Based on the list, we built a prototype decision-support system with the following functions.

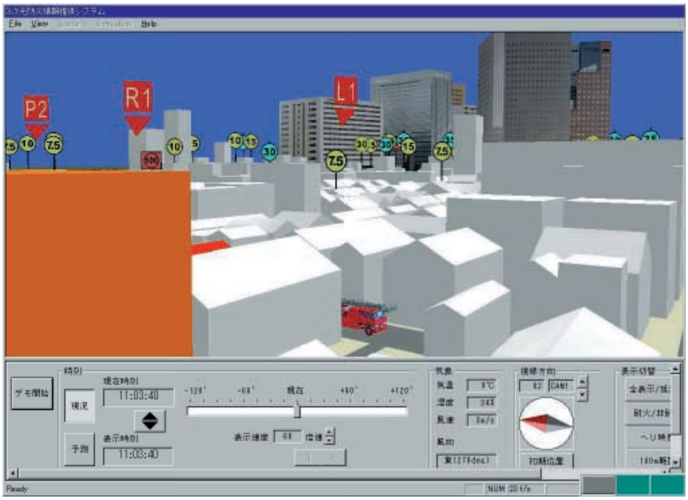

Fig. 6. A 3D perspective view in the prototype decision-support system

1. Real-time 3D visualization
   Real-time 3D graphics are used. Users can select any place and any viewpoint in the 3D space. The location, scale, and degree of affected area are shown, as well as rescue resource information such as the locations of rescue teams, fire engines, hydrants.
2. Fire-spreading simulation
   A real-time fire spreading simulator with physical model [6] is integrated in this system. The prediction of fire-spreading can also be displayed in the same 3D graphics. By adjusting the time-dial, the system can show the status of fire spreading in any time.
3. Integrated interface with GIS
   The system is based on a GIS: The 3D objects are made from geographic data and managed with structured GIS database. We can show the detailed information of a 3D object that is pointed out on a screen.

Figure 6 represents a snapshot of a 3D visualization in our decision-support system.

5.2 Visualization for Communication Support

We assume the rescue teams use communication systems, and we propose to supply the following functions for rescue teams.

1. Real-time map-sharing
   Because almost all disaster information is related to the locations, we can use a GIS for managing the data. For example, a photograph taken by a rescue

team member is stored with the location. A real-time map-sharing mechanism provides a shared electric map with dynamic object on it. In shared map, a picture icon appears at the location so that all of the other members can know the availability of the picture. Hand writings and locations of members are also dynamic objects.

2. Visual data acquisition
   The field commanders of rescue teams need to know the conditions of the disaster area in detail when they plan their actions. Moving pictures are useful in helping their quickly understand the status of the disaster area. Even in the same disaster area, it is impossible to know all of the status which is essential for decision making. For example, commanders need to know the situations on the other sides of burning houses.
3. Support of multi-modal communication
   Walkie-talkies are widely used by rescue teams for communication both within the disaster area and between the area and emergency office. In general, it is very hard to report the conditions of the disaster area and status of a rescue teams' actions via voice links. Using a combination of handwriting-gestures and voice helps. For example, a member of a rescue team can point to one area in the shared map and report vocally that there are burning houses in that area.
4. Safety and security monitoring
   The safety and security of rescue teams is also of great importance to field commanders. The commanders always need to pay close attention to the status of rescue team members. It is impossible, however, for most commanders to monitor all of rescue team members while the commanders are planning the next action. Life sensors enable the commanders to monitor the rescue team members on-line.

Because viewers are used as communication tools in disaster areas, they should be easy to use. Figure 7 is a snapshot of a prototype command-and-control system for rescue teams. The system has all of the function except safety and security monitoring.

Another design for a wearable system is described in Reference [8].

6 Conclusions

This paper has proposed a process workflow model for disaster mitigation and has based a discussion the requirements for the visualization of disaster information. A prototype system with visualization is built for the evaluation. The design and implementation of RoboCup-Rescue simulator viewers is also described.

We would like to apply knowledge from RoboCup-Rescue simulation system in practice, and we are interested in the evaluation of the visualization techniques we proposed. This evaluation will require the use of field trials.

We are planning to integrate the prototype system and the RoboCup-Rescue simulator.

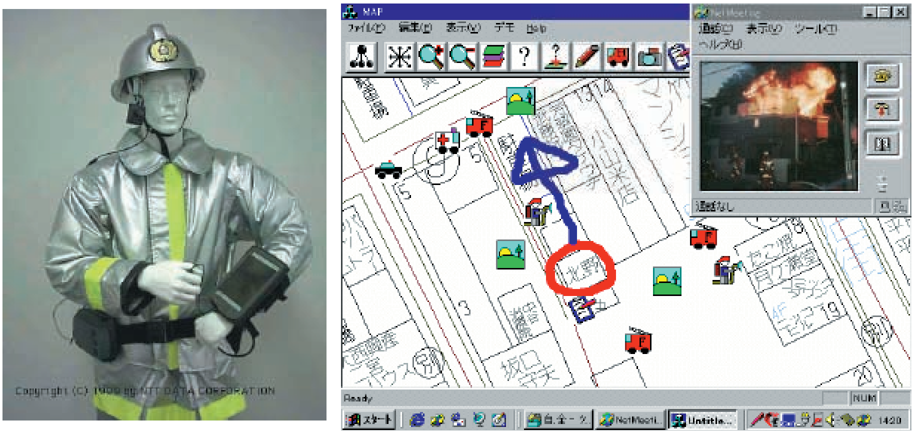

Fig. 7. An outfit of the command-and-control system (left), and a screen image of the system (right): A map-based shared white board is shown with city map of Kobe. Dynamic objects such as fire engines, ambulances, and rescue team members are represented as small icons on the map. A rough circle and a arrow in the center of screen are handwritten. The small window in the top right corner is part of a remote conference system used to share live images.

References

1. Paul R. Cohen, Michael L. Greenberg, David M. Hart and Adele E. Howe, *Trial by Fire: Understanding the Design Requirements for Agents in Complex Environments*, AI Magazine Vol. 10, No.3 Pp.32-48, Fall, 1989.
2. Edited by Administrative Inspection Bureau, at Management and Coordination Agency(MCA), *Towards the perfect anti-earthquake procedure – Based on a study of the Hanshin-Awaji earthquake – (in Japanese)*, Printing Bureau, Ministry of Finance, April 6, 1998.
3. Hiroaki Kitano et al., *RoboCup-Rescue: search and rescue in large-scale disasters as a domain for autonomous agents research*, Proc. IEEE SMC, 1999.
4. Hiroaki Kitano et al., *RoboCup-Rescue – Grand challenge for disaster mitigation (in Japanese)*, Kyoritsu-shuppann, May, 2000
5. The RoboCup-Rescue Technical Committee, *RoboCup-Rescue Simulator Manual (Version 0 Revision 2)*, The RoboCup Federation, 2000.
6. Yuji Ishikawa et al., *RoboCup-Rescue Project (7th report) Fire Spreading Simulation based on Physical Models (in Japanese)* , Proc. 60th IPSJ Conference, Mar., 2000.
7. Yoshitaka Kuwata et al., *RoboCup-Rescue Project (8th report) Collection and Visualization of Disaster Information (in Japanese)* , Proc. 60th IPSJ Conference, Mar., 2000.
8. Astushi Shinjoh et al., *Warable system for the Disaster Mitigation Problem: Mission Critical Man-Machine Interface for RoboCup-Rescue simulator*, Proc. 9th Int. Conf. on Artificial Reality and Telexistence (ICAT'99), Dec., 1999.