WaGuMi for RoboCup2002 : RoboCupRescue Simulation League

Kousuke Shinoda[1,2], Itsuki Noda[1,2,3]

[1]Cyber Assist Research Center, National Institution of Advanced Industrial Science and Technology, 135-0064 Aomi, Koto-ku Tokyo, Japan
kshinoda@carc.aist.go.jp, I.Noda@aist.go.jp
[2]Graduate School of Knowledge Science, Japan Advanced Institute of Science and Technology, 923-1292 asahidai, Tatunokuti Nomi Ishikawa, Japan
[3]PRESTO, Japan Science and Technology Corporation.

Abstract. Our team implemented with a framework to describe behaviors of Posit and a kind of problem solver of PS^2. Posit is a new framework of behavior rule discription for parallel scenario and PS^2 is an engine in order to infer the manipulate *Posit* and determine behaviors according to own situation with *Posit*. Conventional agent description languages are designed for modeling intellectual human behaviors that solves a task to achieve a single goal. However, it is hard to model ordinary people behaviors, for social simulation system, as such a goal-oriented problem solving. Our approaches is to model such behaviors of a human as progress of multiple scenarios, each of which is a state-transitions to achieve consists of two pars, a description language of multiple scenarios, and solver engine to make described scenarios according to changes of environment.

1 Introduction

RoboCupRescue is designed to maximize contribution to the society and attain high-throughput in research itself. RoboCupRescue Simulator[1, 2] is the system consists of a kernel and plug-in modules (ex. fire, traffic and several kinds of agent etc.). The kernel provides functions to synchronize distributed expert modules and to manage shard data among modules. The plug-in modules are classified into three types: expert simulation modules, human-interface modules, and agent simulation modules.

In the disaster simulation of RoboCupRescue, it is important to simulate agents suitably for dynamical environment. Therefore, the behavior design of an each agent is key issue from the viewpoint of realistic simulation.

Agents may have been changed behaviors and purpose according to situation. Therefor, they don't have always clear target and does not aim at efficient problem solution when they make a decision own behavior. It is unrealistic that every people always behave for a single purpose at the usual life. Even if they are under the emergencies, people have different purposes and goals. In other words, we choose a respectively different purpose according to each situation, though it is the persons that were under the same environment.

In this team development, we used that, are a framework of behavior rule description "Part of SITuation (Posit)" and designed a problem solver for Posit called "Parallel Scenario Problem Solver (PS^2)" in which makes it possible to evaluate two or more action target rules, for our team.

2 Behavior Discription for Parallel Scenario

In order to design agents as ordinary people in social simulations, we need to take the following issues into consideration:

- Situated-ness

  Unlike expert agents in a specific simulation, behaviors of agents as ordinary people vary widely. Therefore, the design of such agents should be able to cover a wide variety of such behaviors. In order to control a large number of rules for variable behaviors in real time, the agents need to minimize active rules according to situations dynamically.

- Parallelism and Multiplicity

  As discussed in the previous section, ordinary behaviors consists of collections of actions of reactive response to the environment and of routine works for multiple short term goals rather than sequence of actions extracted from deep-thinking plans for a single goal. In order to handle multiple goals and reactive behaviors, the agents needs a mechanism to process them in parallel. Moreover, the agents can have multiple purposes under a specific situation, since it is unrealistic that every people always behave for a single purpose at the usual life.

In order to attack these issues, we take an approach called "parallel scenario" description, in which agents' behaviors are devided into multiple scenarios each of which consists of a state-transition. Each state is called "Part of SITuation (Posit)", which consists of a set of rule that should be tested under the same situation. We also design a problem solver called "Parallel Scenario Problem Solver (PS^2)" that is able to find solution with multi-scenario in parallel.

2.1 Posit: Part Of SITuation

Posit is a framework to describe agent's behaviors for parallel scenario. Fig.1 shows its syntax.

```
   Scenario ::= (defbehavior ⟨Posit⟩)
      Posit ::= (defposit PositName ⟨Rule⟩)
       Rule ::= (defrule RuleName  :condition CondiForm
                                   :activity   ActivForm
                                   :action     ActioForm )
  CondiForm ::= ([LogicOP] CondiForm*) | (InSensor LocalData)
  ActivForm ::= CalcForm
  ActioForm ::= ([LogicOP] ActionForm*) | (OutSensor ⟨SensorAttr⟩)
    LogicOP ::= and | or | not | do | progn
```

Fig. 1. Syntax of Part of Situation(Posit)

Posit consists of rules that required for achievement of the goal under a specific situation, and has a small goal, so that it means a part of mental situation of an agent. We show some features of Posit, as following:

– Posit is described base on S-expression.
– In "Activity", a rule has the activity value, which is used in case of the conflict resolution, as the numerical value or the formula .
– In "Action", a rule descriptor can define not only the agent's behavior effects outside environment, but also the behavior to which internal environment(Situated RuleSet) is able to be changed.

There are two basic operations, is add_posit and rmove_posit, for situation manipuration and transition. The operation (add_posit PositName) adds new Posit named *PositName* to the situated rule-set, and the operation (remove_posit PositName) removes Posit named *PositName* from. These control operators means that agent behavior descriptor can privide a mechanism, which agent can adjust own mind situation by itself.

2.2 PS²: Parallel Scenario Problem Solver

*PS*² is an engine in order to infer the manipulate *Posit* and determine behaviors according to own situation with *Posit*. The key concept of *PS*² is *situated-ness*, that is, not rules are evaluated in a cycle, but only rules in active Posits are evaluated, where the activeness of Posit are handled explicitly as manipulations of situation.

In *PS*², all Posits are stored into "Behavior Description Sets(Scenario)". Active Posits are selected and managed as "Situated RuleSets", which are candidates to be applied in the current situation, from scenario. In other words, Situated RuleSets means mental situation (for example, the current purposees and intentions) for the agent.

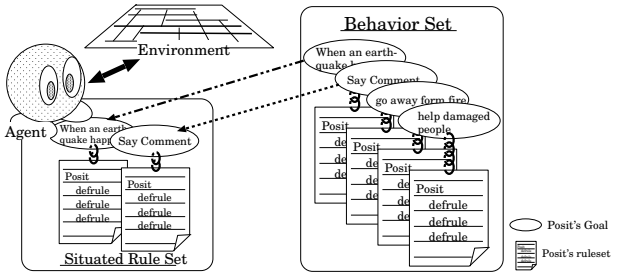

Fig. 2. Parallel Sinario Problem Solver: creats dynamically situated rule-set by selected form behavior description set, which consists of several Posit Rules

In a cycle of exection of an agent's decision making with *PS*², the system selects rule-set, which are suitable for current internal situation, from *agent-scenrio* dynamically as shown in Fig.2. Then, the system evaluates rules in Situated RuleSet and fired selected rules. Furthermore, this problem solver manages Situated RuleSet with PositName, and PositName of initial Posit defines beforehand "init".

3 Apprication: Rescue Agent for RoboCupRescue

3.1 Agent Airchitecture

Fig.3 expresses the flow of information and control, and shows image of Posit description format. Agent has two flows; one of these is information flow, which shows the decision process of own behavior with external information, and the other is control flow, shows control process of internal (mental) model.

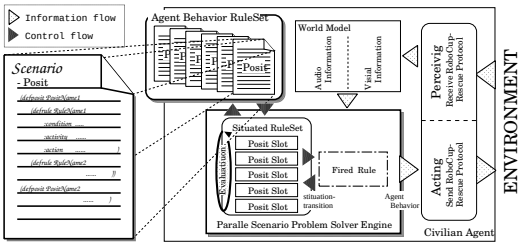

Fig. 3. Processing flow of Decesion Making of Agent's Behavior used by PS^2 and Posit

Moreover, we would like to providing explanation of an application with our proposed agent's architecture. Fig.4 shows construction of modules for rescue agent.

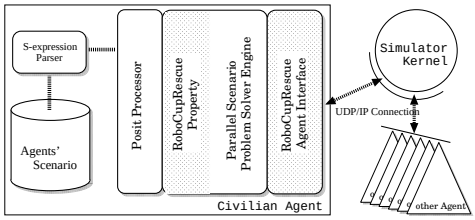

Fig. 4. Architecture of RoboCupRescue Simulator

Agent consists of several modules. First, basic level module is "RoboCupRescue Agent Interface", which is provided by RoboCupRescue Simulator as sample program, has library modules for connecting with simulator kernel and parser of communications protocol. Next level one is Parallel Scenario Problem Solver Engine (PS^2 Engine). This module needs the specific property's definition for RoboCupRescue Simulator in order to implement for this. The other modules, Posit Processor manages situated rule-set, S-expression Parser and Scenario Database. Posit Processor interprets agent's scenario through S-expression parser. If the descriptor of agent's behavior want to change a part of agent's scenarios, can only make a new Posit or modifie a existant Posit in a Scenario Database.

References

1. Kitano, H.: Robocup rescue : A grand challenge for multi-agent system. In: Proceedings of the Third International Conference on Multi-Agent Systems(ICMAS-2000). (2000) 5–11
2. RoboCupRescue HP: (http://www.r.cs.kobe-u.ac.jp/robocup-rescue/)