In: H. Kleine Büning (Hrsg.): Proceedings des Workshops "Simulation in Knowledge-based Systems", Paderborn, April 1998 (= Report tr-ri-98-194, Reihe Informatik, Universität Paderborn), 1998

The Multi-Agent Simulation Environment SeSAm

Franziska Klügl and Frank Puppe
Dep. for Artificial Intelligence, University of Würzburg
Am Hubland, 97074 Würzburg
{kluegl, puppe}@informatik.uni-wuerzburg.de

1. Motivation

The SeSAm (**Sh**e**ll** for **S**imulated **A**gent Sys-te**ms**) provides a generic environment for modeling and experimenting with agent-based systems. We specially focused on providing a framework for the easy construction of complex models.

Although the idea of a domain-independent multi-agent simulation environment is not new, none of the existing environments fulfills the claim of usage without direct programming. Despite of providing a powerful general architecture (e.g. SWARM [4]) or a rather focused one (e.g. EthnoModellingFrame [2]) in many simulation environments the user has to program using a language that provides some additional specialized concepts or classes, but still is based on the syntax of for example C++. SeSAm on the other side keeps the balance between generality and easy usage.

2. Development Tools and System Architecture

As a graphical interface for implementing a model, built-in animation capabilities, tools for collecting and analyzing protocol data, etc. are provided, even scientists without traditional programming experience are able to build and experiment with multi-agent models. In Figure 1 the structure of the environment is presented. In principle all of the provided tools are domain-independent, but in order to facilitate modeling the system can be adapted to a given domain.

3. Simulator and simulated environment

In SeSAm the simulation of the environment is based on two-dimensional gridmaps. A third dimension can be represented as a combination of several two-dimensional maps. Every grid can carry a special amount of information (e.g. chemical trails) or objects. The latter may be resources or agents. Agents execute actions whereas resources are not allowed to actively influence the world. To enrich the environment, it is possible to define events as changes in the environment without a specific source.

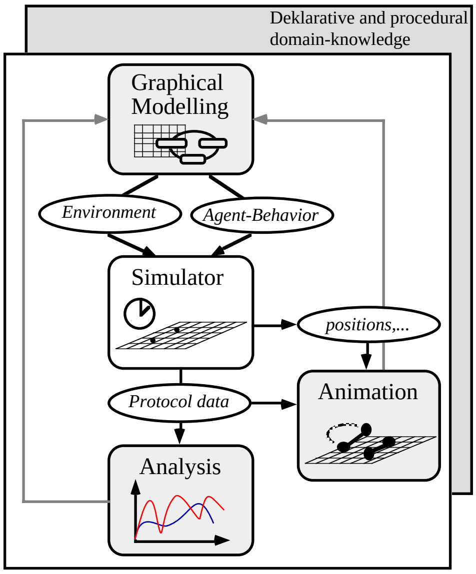

Figure 1: The SeSAm Simulation environment

4. Agent architecture

An agent is specified by fixing three categories: 1. Sensoric abilities and internal parameters 2. the action selection procedure including all internal representation used for it and 3. the effectoric abilities. The SeSAm system provides default solutions for each of these categories. The standard primitive sensoric and effectoric abilities can be adapted domain specifically using graphical editors. Every agent has a current activity which determines the actions it executes on itself and on its environment once every time step. This activity can be selected in two ways: Either because it is contained in a skeletal plan that the agent has chosen to pursue or because it selected as a reflex by firing a rule with a higher priority. The termination of the current activity may be driven by specified goal situations or simple time-outs. The transitions between different activities and the activities themselves can be formulated like every other part of the behavior model using tools as depicted in figures 2 and 3. Reflex rules specifying emergency situations can interrupt the normal execution at any time. In order to support modeling we introduced behavioral roles to structure the set of activities, skeletal plans and emergency rules. These roles correspond to stereotypes. In contrast to the agent class that e.g. represents the modeled species, the roles can change.

Figure 2: Graphical representation of activities and transitions can be used as a powerful tool for editing and navigating

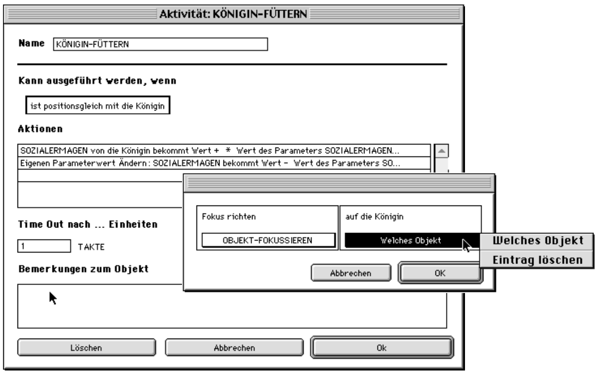

Figure 3: Activities and actions can be specified using more or less sophisticated forms.

When this architecture turns out to be not appropriate for the simulation task, other agent forms can be inserted. There is even a prototypical version with an expert system built with the expert system tool box D3 [5] as the control mechanism for agents.

5. Applications and existing simulations

We used SeSAm not only for toy problems, but for applications of realistic scales. The key to success was, that the experts (biologists, psychologists...) were able to built up the models largely by themselves:

Biologists at Würzburg University developed a model of an ant community that unifies different activities during a complete life cycle of an colony [3]. The simulated ant colony exhibits concurrently foraging, energy storing and its distributing inside of the anthill, production of brood and brood care, enlargement of the nest, interaction between different colonies with emerging „tournaments“ (see figure 4 for a screenshot during a simulation experiment) The success of the simulated ant colony can be measured in the production of new queens for the next generation of colonies. Experiments were made, beginning with the foundation by one queen up to colonies with about 2000 ants. Every agent belongs to one of six roles having a behavioral repertoire of maximal 60 possible activities.

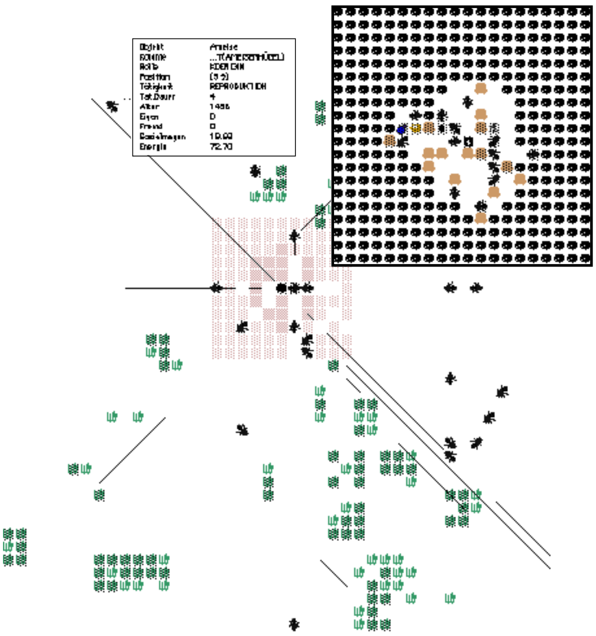

Figure 4: Part of an screenshot during a ant simulation experiment: On maps of the ground surface and the anthill (on the screen in different windows) agents and resource objects, e.g. soil-objects or food-objects, are depicted. Pheromone trails and other odors are also shown on the maps. Information about the internal state of an agent can be accessed using a status window.

SeSAm was also used for biological exercises with students modeling the behavior of individual honey bees inside of a bee hive. They incorporated about 20 different activities for five roles representing different developmental stages. These activities include feeding the queen, caring for larvae or cleaning cells. Another group of biology students evaluated the use of SeSAm for more analytic simulations by modeling the distribution of complete bee swarm.

Another application of SeSAm is providing simulation scenarios for psychological experiments. Psychologists from Bamberg university developed a forest fire scenario: On a map with forest and small villages fire breaks out at certain conditions, spreads and burns down houses, etc. In this environment fire fighting units controlled by humans try to save these inflammable objects. This artificial world has formerly been used to examine problem solving abilities of test persons [1]. This simulation scenario has been rebuilt in SeSAm for additionally testing delegation abilities in combination with problem solving. As SeSAm provides a possibility to directly control single fire fighting units (see figure 5 for a screenshot), the test persons can control the units per hand or additionally built automatons for the units automatically selecting the next action.

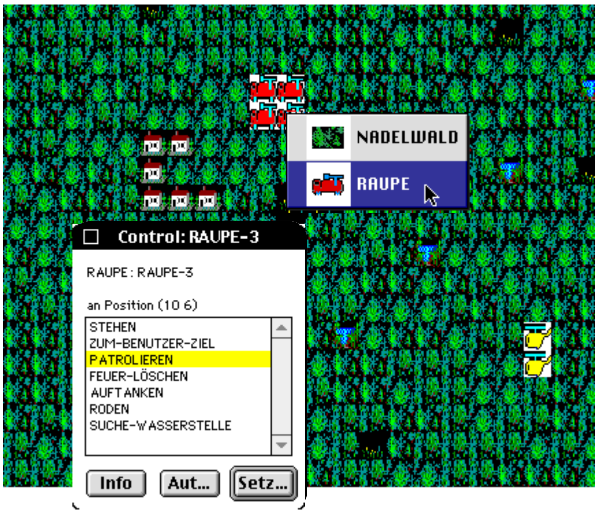

Figure 5: Part of a screenshot from the fire fighting simulation: The vehicles can be controlled using dialogs like the one in the front. Clicking on a grid all objects on that cell are shown, like in the upper part of the figure.

Also mixed procedure is possible. Besides the examination of the ability for delegate tasks to units the modeling in SeSAm during the experiments is useful for making the problem solving of the test persons explicit.

Besides SeSAm is also used for implementing testbeds for certain aspects of planning, learning or organizational design in complex and dynamic environments like a supermarket with thieves, soccer games and strategic games like „civilization“.

References

[1] Dörner, D. and H. Schaub (1992) Spiel und Wirklichkeit: Über die Verwendung und den Nutzen computersimulierter Planspiele. In. Kölner Zeitschrift für Wirtschaft und Pädagogik, 12, pp. 55-78

[2] Drogoul A. and Ferber J. (1992) Multi-Agent Simulation as a Tool for Modeling Societies: Application to Social Differentiation in Ant Colonies, C. Castelfranchi and E. Werner (eds.). Artificial Social Systems (MAAMAW'92), Springer, pp. 3-23

[3] Klügl, F., Puppe, F., Raub U. and Tautz, J. (1998) Simulating Multiple Emergent Phenomena - Exemplified in an ant colony. Accepted as a poster for the Sixth International Conference on Artificial Life. June 1998, Los Angeles.

[4] Minar, N., Burkhart, R., Langton C. and M. Askenazi (1996): The Swarm Simulation System: A Toolkit for Building Multi-agent Simulations. http://www.santafe.edu/projects/swarm/

[5] Puppe, F., Gappa, U., Poeck, K., Bamberger, S. (1996). Wissensbasierte Diagnose- und Informationssysteme, Springer