Salzburg Interest Group on Integrated Systems

siGis-00x-2000 March 23, 2001

Elc goes Java

Draft

Roland Schwaiger

rschwaig@cosy.sbg.ac.at

Department of Computer Science
University of Salzburg

siGis
Salzburg Interest Group on Integrated Systems

Correspondence to:

Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Elc goes Java

Roland Schwaiger
Department of Computer Science
University of Salzburg
A–5020 Salzburg, Austria
rschwaig@cosy.sbg.ac.at

Abstract

Contents

List of Figures

1 Introduction

2 Related Work

3 Basic Entities and Responsibilities

This section gives a first idea of the entities (classes) do develop their responsibilities and possible communication with other entities (classes).

3.1 Managers

The Managers are responisble for setting up the system in case of starting ELC the first time (Program Manager). Diverse parameters are initialized from initialization files (XML). These data are just relevant for program setup and will be used during simulation execution in addition to simulation parameters (Simulation Manager). The user has the possibility to save the parameters to files (XML). Furthermore the user may manipulate the parameters in a GUI. All other entities (classes) may request parameter values from the managers. The managers decide if the simulation has to be started with GUi or without GUI (batch mode).

The simulation has to store the different states of the simulation in order to re-run the simulation from a certain step on.

3.2 GUI/Visualization

The VGUI/Visualization is responsible for giving the user the possibility to interact with ELC through a GUI, i.e. setting parameter values, starting simulations, stopping simulations, step-by-step tracing of a simulation, visualizing of the current simulation graphically, visualizing diverse statistics (e.g. number of places visited), showing encoding of the ANNs in bitstrings (GA), showing ANNs (transfer functions, weights, outputs, activations, etc.), details of the simulation with respect to single agents, etc.

3.3 World

The World has the responsibility to manage the internal representation of the environment where the agents may move. The user has the possibility to select from a predefined list of environments (e.g. torus, sphere, cylinder, rectangle, etc) and define the word by points and neighbors (XML). The agents will move on the discretized environments. Thus the World has to deliver possible neighboring places where an agent may move to.

3.4 Agents

The Agents has the responisbility to enable software agents to move in the world, to sense other agents and possibly to communicate with other agents and exchange data. Agents are steered by ANNs, but agents should have the possibility to be set up with an arbitrary steering mechanism (e.g. rule based, script based, etc.) Agents deliver information about their internal state. Agents have sensors to collect information from their environment. Their may be an unlimited number of sensors associated with an agent.

3.5 Sensors

The sensors have the responsibility to collect information from the word (environment) of the agents. Sensors have internal behavior, e.g. functionality, thus they may be simple to complex. Agents request sensors for collected data, information from the environment.

3.6 Artificial Neural Networks (ANN)

The ANN are responsible for steering the agent. Based on sensory input they calculate the appropriate output and deliver the next step to perform. ANNs may be defined in a file (XML) or specified by the user in the GUI.

3.7 Genetic Algorithms/Evolution

The Genetic Algorithms/Evolution have the responsibility to change various parameters of the ANNs of the agents. The ANNs are encoded in bitstrings which are the objects to bemanipulated by the evolutionary process by mutation/crossover. Diverse selection methods for mating have to be implemented (see jevolution, Mayer). The new population for the next run in the evolutionary cycle is the result of the evolutionary process. The user has the possibility to select Generational based evolution and steady-state evolution. Diverse fitness functions have to be implemented.

3.8 Statistics

The Statistics have the responsibility to calculate during evolution an simulation run a predefined number of statistics, s.a. number of places visited, nuber of collisions, etc. These calculated statistics have to be displayed and presented to the user during simulation in the GUI or after batch running.

4 Conclusion

5 Acknowledgements