



This paper appears in the publication, *An Imitation-Based Approach to Modeling Homogeneous Agents Societies*
by Goran Trajkovski © 2007, Idea Group Inc.

Chapter XI

On a Software Platform for MASIVE Simulations

with Adam Conover, Towson University, USA

Abstract

Pattern-Aided Simulated Interaction Context Learning Experiment (POPSICLE) Agent Simulator began as a sample project in object-oriented agent programming, but quickly grew into a complete framework for the simulation of agent behavior based upon an associative memory model. The system began its implementation as a Java 2 (J2SE 1.4) application, but was later migrated to a Java 5 application (J2SE 1.5) to utilize the type-safe collections and enumerated types that became available in the latest Java version. Various design patterns were employed in the development; the most predominate being the Model/Controller/View (MCV) architecture. As we will see later, the system also relies heavily on delegation and observers.

Introduction

In this section, we overview the key notions in the agent model, as implemented in this project, followed by a description of the simulator.

The system simulates the behavior of autonomous agents in a two-dimensional world (grid) of cells, which may include cells the agent is free to move into or walls that block movement. The goal of an agent is to satisfy specific user-defined *drives*, such as hunger, thirst, and so forth. An agent may have any number of drives, but only one is active at any given point in time. Additionally, the user populates select world cells with *drive satisfiers* that are used to satisfy the active drive of any agent entering the cell. In other words, when an agent moves into a cell containing a drive satisfier, the drive is only satisfied if it is the agent's active drive.

In the beginning, the agents navigate around the world using a user-defined *inherent schema*, which is a short series of moves the agent will make by default. Gradually, an agent builds up an internal *associative memory* table as it explores the environment in search of drive satisfiers. As the agent moves in search of a drive satisfier, *observations* are made and recorded in the *emotional context* of the active drive. Once a drive has been satisfied, the recorded observations (leading to drive satisfaction) are recoded in the agent's associative memory table. As the agent continues to explore the world, other drives may become active, leading to new observations in new contexts. As the agent continues to build a model of the world in relationship to its drives, the agent will begin to use its associative memory to plan a route to the drive satisfier.

The agent uses current observations to derive *expectations* from the associative memory table. When a matching observation is found in the table, the agent temporarily abandons its inherent schema and uses the expectation to execute the next series of moves. If the observations made during this next series of moves match another observation in the table, the process continues until the drive satisfier is reached. If at any time, a subsequent observation does not match the expectation the agent records a surprise and returns to its inherent schema to continue exploration—all the while, continuing to make new observations. Additionally, if the agent cannot make a move because a path is blocked by a wall or world boundary, the agent registers this as *pain*, skips the move, and continues execution of schema or expectation.

Interface

Before delving too deeply into the internals of the system, let us look at the system from a users point of view. Figure 1 shows the four main interface areas in one

29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/software-platform-masive-simulations/5099

Related Content

Adaptive Congestion Controlled Multipath Routing in VANET: A Multiagent Based Approach

Anil D. Devangaviand Rajendra Gupta (2017). *International Journal of Agent Technologies and Systems* (pp. 43-68).

www.irma-international.org/article/adaptive-congestion-controlled-multipath-routing-in-vanet/201444

Penguins Huddling Optimisation

Mohammad Majid al-Rifaie (2014). *International Journal of Agent Technologies and Systems* (pp. 1-29).

www.irma-international.org/article/penguins-huddling-optimisation/117203

A Multiagent Approach to Teaching Complex Systems Development

Simon C. Lynchand Keerthi Rajendran (2011). *Multi-Agent Systems for Education and Interactive Entertainment: Design, Use and Experience* (pp. 70-87).

www.irma-international.org/chapter/multiagent-approach-teaching-complex-systems/50395

A Novel Approach for Business Process Model Matching Using Genetic Algorithms

Mostefai Abdelkaderand Ignacio García Rodríguez de Guzmán (2020). *International Journal of Distributed Artificial Intelligence* (pp. 1-19).

www.irma-international.org/article/a-novel-approach-for-business-process-model-matching-using-genetic-algorithms/264508

The Future of Companionable Agents

Roberta Catizoneand Yorick Wilks (2011). *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices* (pp. 379-394).

www.irma-international.org/chapter/future-companionable-agents/54647