An Aspect-Oriented Domain-Specific Language for Modeling Multi-Agent Systems in Social Simulations

Diego de S. Braga[1], Felipe Omena M. Alves[2], Fernando Buarque de L. Neto[1], and Luis Carlos de S. Menezes[1]

[1] University of Pernambuco (UPE), Polytechnic School of Pernambuco (POLI)
Street Benfica, 455, Madalena – 50750-410 – Recife, Pernambuco – Brazil
{dsb,fbln,lcsm}@ecomp.poli.br
[2] Estácio do Recife College
Av. Eng. Abdias de Carvalho, 1678, Madalena – 50720-635 – Recife, Pernambuco – Brasil
felipeomena@gmail.com

Abstract. Aspect-oriented programming (AOP) is a programming paradigm which aims to increase modularity by allowing the separation of cross-cutting concerns. This paper presents the definition and characteristics of the domain-specific language, aspect-oriented, AspectNetLogo and its compiler, the AspectNetLogoCompiler and show the use of this system in a multi-agent system in social simulation. This system allows the definition of the elements of the agents in the NetLogo environment in an isolated way and simplify the implementation of social simulations.

Keywords : AOP, MAS, NetLogo, Social Simulation.

1 Introduction

Social Simulation is an area of research that uses computational methods to solve problems in social sciences (politics economics anthropology etc.) using agents.

Agent Oriented Programming [1] defines the computing from social interactions of entities known as agents. An agent is an entity that perceives changes in the environment and act in response to these changes in order to achieve some predetermined goal. Agents in a Multi-Agent System (MAS) [2] have interesting characteristics such as: autonomy, proactivity, learning, communication and coordination of distributed tasks, among others that enable them to deal with complex problems.

Aiming to simplify the execution of social simulations using agents, several tools such as PAX [3], Repast [4] and NetLogo [5] provide to the programmer a basic infrastructure with the main elements of a multi-agent system. These structures have been successfully used to perform social simulations [6][7][8]. However, the tools available today to perform social simulations of agent-based models require considerable effort for modeling as they still require computer science knowledge from the user to define new types of simulations, because the characteristics of a simulation are mixed in the code. These abilities are not commonly found in social researchers who demand this type of tool.

H. Yin et al. (Eds.): IDEAL 2012, LNCS 7435, pp. 578–585, 2012.

The Aspect-Oriented Programming (AOP) [9] is a programming model that is designed to increase the modularity of implementation of cross-cutting concerns in computer systems. A cross-cutting concerns is a property of an application to be implemented, requires modification of a large number of modules, it cannot be isolated using conventional techniques of modularization. The principle of separation of concerns is based on construction systems that have different modules for solving different responsibilities. Thus, it will not be found cross-cutting concerns in the code, possible to change old services offered without other parts of the systems being impacted.

This paper proposes the establishment of a domain specific language (aspect-oriented) to the description of multi-agent systems in Social Simulations.

As example of using aspects in multi-agent simulations of this paper implements an aspect-oriented extension of NetLogo. The NetLogo is a programming language and integrated environment for multi-agent modeling. It is particularly well suited for modeling complex systems developing over time. Modelers can give instructions to hundreds or thousands of "agents" all operating independently. This makes it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from their interaction. However, the definition and characteristics of agents in complex problems are frequently produced in different parts of the application, which complicates the understanding and maintenance of the models developed.

The use of aspects in an agent oriented system, in particular NetLogo, will allow the elements that compose an agent to be defined and studied in isolation from the rest of the simulation. This feature can facilitate the definition of new simulations and reuse elements of existing simulations in new contexts.

2 AspectNetLogo

The Aspect-Oriented Programming defines structures known as aspects, permitting the declaration of characteristics in isolation from the rest of the system. At some point of compilation, these settings will affect the code of the multi-agent model, in a process called Aspect Weaving (view item 3.4). That way, a new file is generated containing all specifications performed, not changing the original model, as illustrated in Figure 1. The artifact obtained by the implementation of the basic functionality of an multi-agent system, for example, is joined to the characteristics defined in the AspectNetLogo language (e.g. autonomy, adaptation, interaction, etc.). The result of this combination is the generation of a new model containing these characteristics incorporated into the basic functionalities of the application.

The basic features of multi-agent model for example, declarations and procedures are implemented using the language component NetLogo. For the definition of aspects of the model are used the aspect-oriented language, presented in this work, AspectNetLogo [10]. It supports the implementation of aspects, defining their behaviors and situations that will occur in a clear and concise way.

AspectNetLogo is conceptually very similar to AspectJ [11] and AspectC++ [12]. The reason for this similarity is trying to allow people who learned the aspect-oriented paradigm by using AspectJ or AspectC++ implementation to easily switch over to AspectNetLogo, if they are also already familiar with NetLogo. Some important concepts and definition are: