A GENOME BASED VISION OF MULTI-AGENT SYSTEMS

Monica Vitali
Università degli studi di Palermo, viale delle Scienze, Palermo, Italy

Massimo Cossentino, Riccardo Rizzo
CNR-ICAR Palermo, viale delle Scienze, Palermo, Italy

Salvatore Gaglio
Università degli studi di Palermo and CNR-ICAR Palermo, viale delle Scienze, Palermo, Italy

Keywords: Agent models and architectures, Multi-agent systems, adaptation.

Abstract: A set of software agents can be programmed to provide a large but finite set of services, often defined during design phase. After an evolution of the external environment, the pre-defined services could be unable to satisfy the requested quality. In this work an agent framework is proposed capable to adapt the agents in order to improve the quality of services provided by an agent society in correspondence with a modification of the external environment. These agents are based on a biologically inspired structure (genome), that defines all their behaviors and knowledges.

1 INTRODUCTION

Agent Oriented Systems should be able to autonomously adapt and deliver new services in response to unforeseeable problems, like many living things do. In living things this characteristic is mainly based on genome and natural selection. We propose to replicate this mechanism in Agent Oriented Software and Agent Systems.

In the proposed system agents capabilities are described in their genome and their improvement is possible by means of a Darwinian evolution. When a solution to a problem is not achievable (the corresponding service is not available or it does not provide the required quality of service), several agents can reproduce themselves thus creating a new generation of agents that have new capabilities and can satisfy the requirements. The consequence of this behavior is the creation of an adaptive system (Gleizes et al., 1999) realized through an extensive adoption of Genetic Programming techniques (Koza, 1992) and automatic code generation, compilation and execution.

The focus of our attention is not on the agent itself, but on its genetic makeup: the genome. This genetic makeup can be decomposed in different layers. In the first layer of the genome there are two kinds of chromosomes: a Knowledge Chromosome which describes knowledge about the environment and a set of Ability Chromosomes which describe agent's abilities to interact with the world. At a deeper layer we have genes. Each chromosome is made of genes. In the Knowledge Chromosome, each gene describes an element of the knowledge (predicates, concepts and actions), while, genes within Ability Chromosomes describe plans components.

During the designing process it is essential to define and manipulate the genome. In the definition of the genome an initial set of genes is given. This set does not have to contain the solution but only the elements which allow the creation of the first generation. From this initial set, through manipulation, genes are combined and activated originating new genomes. These genomes have to be evaluated through an objective function for measuring their level of adaptation to the required skill.

The agent adaptation procedure is represented in Fig.1. This process is started by an agent called CrosserAgent after a request for an unavailable service is received. The CrosserAgent creates new generations which will contain some agents inherited

Vitali M., Cossentino M., Rizzo R. and Gaglio S..
A GENOME BASED VISION OF MULTI-AGENT SYSTEMS.
DOI: 10.5220/0003165204060409
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 406-409
ISBN: 978-989-8425-41-6

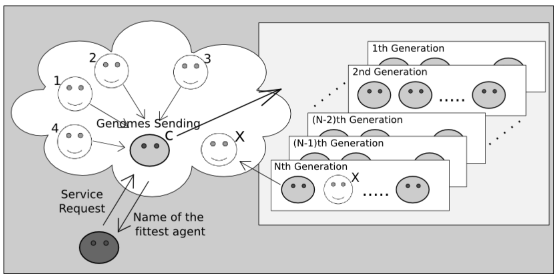

Figure 1: A representation of the adaptation procedure. GenomeAgents are represented in white. The agent selected at the end of the procedure is marked with an "X", while the CrosserAgent is marked with a "C".

from the previous generation and some new agents obtained crossing the agents' genomes. The number of individuals per generation is a parameter of the adaptation process. During the adaptation process two agents merge their chromosomes according to the rules of genetic programming that will be discussed later. The parents transmit their genes to the child which evolves and gains the capability of reaching different results. The adaptation process ends when an agent in the current generation provides a satisfying service or if an a priori defined number of generations has been reached. The fittest agent is selected and becomes a member of the society. The CrosserAgent notifies its name to the requesting agent in order to fulfill the service request. This process will be discussed more in details in section 3.

2 THE GENOME STRUCTURE

This section introduces the Genome structure shown in form of an UML class diagram in Fig.2. This figure highlights the two main parts in which the genome can be decomposed at a logical level: knowledge and abilities. The division is pointed out through the inclusion of chromosomes in two different packages.

Starting from the higher level, the genome (at the top of Fig.2) contains all the information needed to describe the agent; from this information a new agent can be created.

The genome enables a set of *Service*s which makes explicit the functions offered by the agent to the external environment. The genome is composed of a *KnowledgeChromosome* and a set of *AbilityChromosome*s.

The *KnowledgeChromosome* aggregates genes which refer to ontological concepts (*OntologyGene*) and that are specialized in three categories: *(i) ConceptGene*: describes an instance of a concept of the ontology; *(ii) ActionGene*: describes an instance of an action of the ontology; *(iii) PredicateGene*: describes an instance of a predicate of the ontology.

The *AbilityChromosome* is composed of a set of node genes which describe the plan structure (*NodeGene*) and by the contents of these nodes which describe the action associated to them. Node contents can be of three different kinds: predicate or action genes (indicated as *PredicateGene* and *ActionGene* in Fig.2) or other Ability Chromosomes. Plugging in an Ability Chromosome with a node allows us to associate a behavior, described by another plan, to a node, thus creating a sort of recursive structure. There are four kinds of nodes: *StartNodeGene*, *EndNodeGene*, *ActionNodeGene* and *IfNodeGene*. The node classification reported here is inspired by (van Der Aalst et al., 2003).

3 THE AGENT ADAPTATION PROCESS

The agent adaptation process, led by the CrosserAgent, is composed of the following iterative steps:

- definition of the parents' sub-society: this sub-society includes all the agents that will be used in the adaptation process and that contribute with their ability genes and knowledge genes to the definition of the resulting agent;
- creation of the new generation by using adaptation techniques;
- evaluation of the results provided by the new agents;
- stop of the process if one (or more) agent(s) successfully provide the required service.

During the agent adaptation process we create new generations by using mutation, elitism and crossing techniques. While the first two techniques are reused from litterature (Banzhaf, 1998)(Mitchell, 1998), crossing is described below.

The agent adaptation process can be divided into two steps: knowledge crossing and ability crossing.

Knowledge crossing allows to modify the set of knowledge genes about the environment. Knowledge Chromosomes crossing is inspired by (Noy and Musen, 1999) and it is performed over each agent's knowledge gene by using four techniques:

- **Fusion:** the two parents' knowledge genes are melted in a single gene;
- **Selection:** one of the parents' knowledges is chosen while the other one is discarded;
- **Union:** both of the parents' knowledges are copied in the new individual;

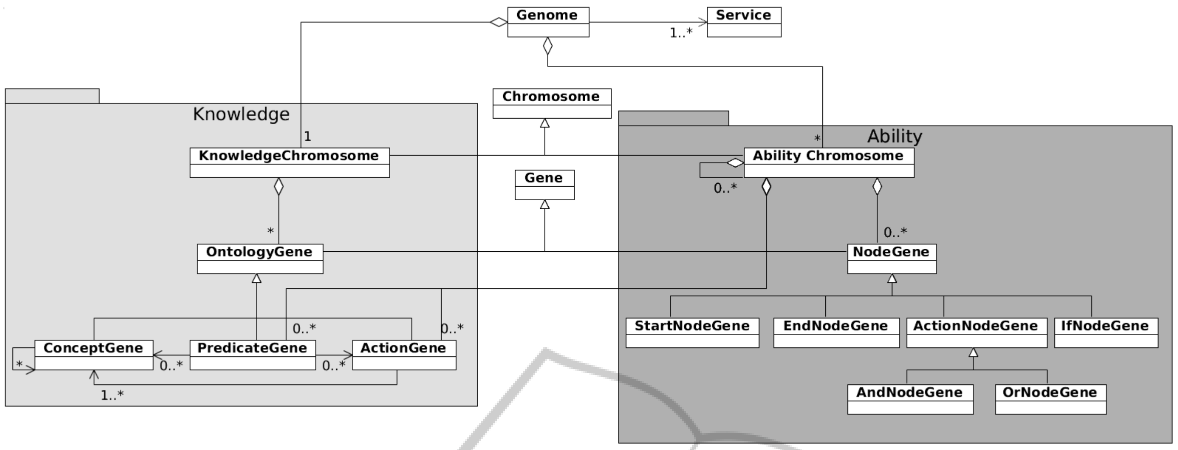

Figure 2: The genome structure. Genome is composed of two kinds of chromosome: Knowledge Chromosome and Ability Chromosome; both are composed of genes.

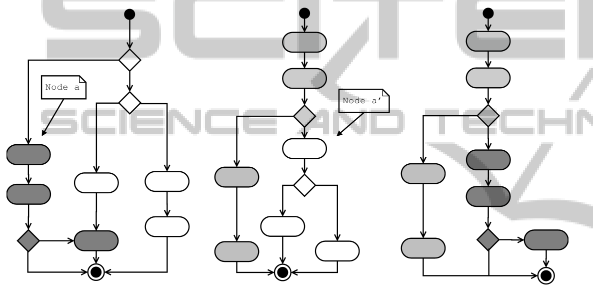

Figure 3: On the left the two plans of the parent agents and on the right the resulting plan. The parts of the plan selected for the crossover procedure are filled.

- **Copy:** if a particular portion of knowledge is present only in one of the parent, it is copied to the generated agent.

The result is a new Knowledge Chromosome. Once the knowledge crossing is completed, the ability crossing can be executed.

Tha ability crossing is performed on the abilities of an agent. Abilities are represented through plans, composed of nodes, and are labeled with a goal, which indicates the ability purpose. Agents in the platform are provided with a higher-level plan which handles the agent's life-cycle and allows each agent to interact with the external environment. This plan is always crossed by a fusion operation. All the other plans can be crossed also by using the selection, union or copy techniques already described in the previous paragraph. Since selection, union and copy simply transfers a plan from a parent to its child, the sole operation worth of a discussion is plan fusion and therefore it will be discussed in what follows. Fusion can be performed only if two plans are similar (if they have the same goal). Plan crossing is shown in Fig.3.

The two parents' plans play a different role in this part of the adaptation process. The receiver's agent plan is used as the basis for implanting the contribution from the donor agent. An example of donor and receiver plan fragments are shown respectively in the right and the middle part of Fig.3. The two fragments are linked by replacing a randomly selected node (node a' in Fig.3) from the receiver's plan with the one selected from the donor's one (node a and its successors), as shown in the right part of Fig.3.

4 EXPERIMENTAL RESULTS

In the case study, the initial set of agents is composed of individuals drawing simple geometrical shapes. The aim of each agent is to reproduce a given picture, in the reported example a trapezium (Fig.4a).

The agent divides the picture in small chunks by using a grid and tries to fill each position of the grid according to the guidance provided by the target picture. The work is carried on through several iterations and each iteration is populated by a different generation of agents. The initial generation is composed of two simple agents. Each of them is able to fill in a grid cell with the shape of a triangle: the first agent (Fig.4b) draws a dark gray triangle oriented towards the up-left corner of the cell, the second agent (Fig.4c) draws a light gray triangle oriented towards the bottom-right corner of the cell.

Clearly none of two agents supply a fulfilling result; besides, neither the simple cooperation of the two agents could solve the problem. So that an adaptation process is started: the genome codes the color, dimension and shape of the grid cell together with the plan used to generate the figure. The evaluation pro-

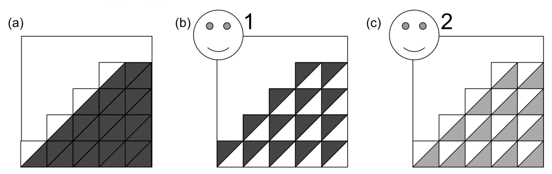

Figure 4: In (a) the target picture. In (b) and (c) the results achieved by two agents of the initial society.

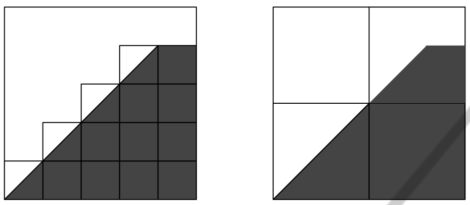

Figure 5: Two agents which provide the required service in different ways.

cess compares color and shape of the obtained figure to the target one.

In our experiments, after about nine generations, several individuals which perfectly reproduce the desired picture have been created (obviously because of the random characteristic of the new generations production different runs of the experiment may produce different results). Fig.5 reports two examples of such individuals; as it is possible to see, the two agents use a different grid to decompose the target picture.

The test case has been evaluated with different pictures and colors. It has been observed, as it was expected, that the number of generations needed to reach a perfectly fitting outcome grows up with the complexity of the target picture.

The adopted adaptation process proved to be successful but it is to be noted that the development framework is undoubtedly complex in its use and the setup of a new experiment requires a lot of programming.

5 CONCLUSIONS AND FUTURE WORKS

In this paper we proposed a service adaptation mechanism as an integral part of an agent-oriented adaptive and self-organizing society. As a first step towards our goal we tested an adaptive system inspired by the Darwinian evolution theory where all the agents' features are codified in a genome-like structure. In order to improve the quality of a given service, several agents can reproduce and generate new individuals which better fit the target. These individuals are provided with new capabilities derived by their parents. The approach has been tested through simple case studies. The application reported in this paper proves that it is possible to obtain a perfectly working agent from original agents which provides a service with a low quality service. Using the proposed Genome Framework the problem moves from the implementation of a solution to the definition of the problem domain.For sure different techniques may be explored (and they will be in the future) but the goal of the current study is evaluating the adoption of the proposed genome-based description of agent capabilities and knowledge. The obtained results encourage the development of further release of the proposed framework. The use of a formalization language to describe the genome structure might be the following step in order to lay the groundwork for an agent-oriented language.

REFERENCES

Banzhaf, W. (1998). *Genetic Programming: An Introduction on the Automatic Evolution of computer programs and its Applications*. Morgan Kaufmann.

Gleizes, M.-P., Camps, V., and Glize, P. (1999). A theory of emergent computation based on cooperative self-organization for adaptive artificial systems. In *In proc. of Fourth European Congress of Systems Science*.

Koza, R. (1992). *Genetic Programming: On the Programming of Computers by means of Natural Selection*. MIT Press.

Mitchell, M. (1998). *An introduction to genetic algorithms*. The MIT press.

Noy, N. and Musen, M. (1999). An algorithm for merging and aligning ontologies: Automation and tool support. In *Proceedings of the Workshop on Ontology Management at the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pages 1999–0799.

van Der Aalst, W., Ter Hofstede, A., Kiepuszewski, B., and Barros, A. (2003). Workflow patterns. *Distributed and parallel databases*, 14(1):5–51.