

An Evolutionary Nano-Agent Control Architecture for Intelligent Artificial Creatures

Jean-Claude Heudin

IIM Interactive Media Lab.

Pôle Universitaire Léonard de Vinci – 92916 Paris La Défense, France

(Jean-Claude.Heudin@devinci.fr)

Abstract: This paper describes a new multi-agent architecture for the Evolutionary Virtual Agent project. It introduces the nano-agent bio-inspired control architecture and its nanoScheme programming language. A typical application consists of one or more nano-agents, and possibly up to a large number if necessary as in natural swarms. As an example, it describes an online self-animated character that uses natural language and emotional expressions. This virtual character is based on a “schizophrenic” model composed of multiple distinct personalities, each with its own pattern of perceiving and interacting with the user.

Keywords: artificial creature, intelligence, nano-agent, schizophrenic architecture

I. INTRODUCTION

Recent advances in robotics and artificial life make now possible to develop a growing number of real-world applications in various fields. However, these applications require a new generation of open software architecture that combines such technologies with lightweight design and portability for onboard and online applications. This paper proposes a new evolutionary nano-agent control architecture designed for intelligent artificial creatures. This software environment takes advantages of our past experiences in distributed artificial intelligence with the Knowledge-based Operating System (KOS) [1], multi-expert applications such as the Electronic Copilote project for combat aircrafts [2], and the more recent Evolutionary Virtual Agent (EVA) applications [3].

In the first part of the paper, we introduce the nano-agent bio-inspired architecture and its programming language called nanoScheme. The second part describes an application developed using this software environment: an online self-animated character that can interact using natural language and emotional expressions. This virtual character is based on a “schizophrenic” model in which the character has multiple distinct personalities, each with its own pattern of perceiving and interacting with the user. The qualitative efficiency of this prototype is then compared with the ALICE conversational engine [4]. The paper concludes by outlining future developments and some possible applications.

II. ARCHITECTURE OVERVIEW

1. Nano-agent architecture

We propose a multi-agent bio-inspired architecture that does not try to *simulate* a specific organism but rather integrates several artificial life features in order to *implement* machine life and intelligence. A typical application consists one or more *nano-agents*, and possibly up to a large number if necessary as in natural swarms. Nano-agents can run on a standalone machine or can be distributed on a network of computers depending on the application requirements. We call them “nano” because of their small size and resource requirement compared to most existing software environments. In the current implementation, the core technology is implemented in Java and its weight is less than 25 Kilo-bytes. Most applications require a small set of knowledge-base and behavioral scripts text files, thus resulting in lightweight applications that are well-suited for web-based, mobile phone, robots and embedded environments.

2. The nanoScheme language

The behavior of each nano-agent is programmed using a user-friendly language, called nanoScheme, based on the Scheme programming language. It includes a reduced set of primitive functions which is a subset of the R4RS specification [5]. This subset is given in table 1. Most of the missing features of the Scheme specification could be added by programming them directly in nanoScheme. This provides the

application developer a high-level interactive language which is embedded in each nano-agent.

Table 1. The nanoScheme core primitives

| |
|--|
| <p>+ - * / = < > <= >=</p> <p>cos sin acos asin log expt round</p> <p>number? integer? even? string? symbol?</p> <p>string=?eqv? pair? null? procedure?</p> <p>string->number number->string</p> <p>string->symbol symbol->string</p> <p>substring string-length string-append</p> <p>cons car cdr set-car! set-cdr!</p> <p>quote eval apply load</p> <p>define lambda set! begin if</p> |
|--|

3. Artificial life primitives

The nanoScheme language includes also a reduced set of bio-inspired primitive functions. They have been designed in the same spirit of Tom Ray’s Tierran assembly language [6]. That is, the production of synthetic organisms based on a computer metaphor of organic life in which CPU time is the “energy” resource and memory is the “material” resource. These functions are the following in the current implementation :

Table 2. Bio-inspired functions in nanoScheme

| |
|--|
| <p>reproduce: create a new nano-agent.</p> <p>terminate: kill the current nano-agent.</p> <p>diffuse: diffuse a message to other nano-agents in the local environment.</p> <p>rule: define a new behavior rule consisting of condition and action expressions, and a priority.</p> <p>engine: make an inference loop on the current behavior rule base.</p> <p>crossover: genetic programming crossover operator.</p> <p>mutate: genetic programming mutate operator and random code generator.</p> <p>random: return a random real number.</p> <p>time: return the current real time.</p> <p>stress: return a “stress” value based on the current available memory and computing resources.</p> <p>plugin: dynamically load a new package of dedicated primitive functions.</p> <p>message: hook invoked when the nano-agent receives a message.</p> <p>lifepulse: hook for implementing periodic behaviors.</p> |
|--|

Since all code, behavior rules, and messages are basically S-expressions (i.e. lisp expressions), the use of genetic programming seems natural in this environment [7]. Also, the remote execution of code on distant nano-agents is a natural feature by simply sending messages containing S-expressions. These expressions are then evaluated by all nano-agents. This approach enables an easy implementation of distributed algorithms on nano-agents.

4. Natural language interaction

Each nano-agent can be specialized to a given task by dynamically loading an appropriate dedicated package using the *plugin* primitive function. A typical example is the natural language package providing the developer natural language processing features [8] such as categories extraction, template expressions, etc. These functions allow the design of efficient behavior rules for implementing natural language interactions with the user. Table 3 gives a simple example of such a rule that handles the way to answer to most kinds of “bye” expressions.

Table 3. Example of a behavior rule

| |
|---|
| <pre>(rule "example" 2 ; condition part '(or (find? *categories* "BYE") (find? *user-input* "see you")) ; action part '(begin (show HAPPY 0.5) (random-template "BYE"))))</pre> |
|---|

III. THE EXPERIMENT

1. The experimental prototype

We describe an experiment that illustrates the use of the nano-agent architecture: an online self-animated character that interacts using natural language and emotional expressions. This experiment uses multiple nano-agents for the natural language processing part, plus an animated 3D character agent, a text-to-speech agent and a “chat” agent for interactions with the user. Figure 1 shows the graphical interface.

2. The “schizophrenic” character model

The virtual character is based on a “schizophrenic” model in which the character has multiple distinct identities or personalities, each with its own pattern of

perceiving and interacting with the user. Note that a more accurate psychological term is Dissociative Identity Disorder rather than schizophrenia [9]. Each personality is implemented as one or more nano-agents that reacts to the user's inputs by computing an answer using their behavior rules and diffusing messages containing answers. Then, a dedicated nano-agent "reconnects" the identities of the disparate alters into a single functioning identity by selecting the "thought" with the highest evaluation. In this prototype we used a straightforward priority-based scoring approach.

The different personalities are based on stereotypes used in story telling for creating believable characters [10]. There are four basic personalities:

- The *Protagonist* is essentially the principal driver of the effort to achieve the goal.
- The *Antagonist* is the personality which is opposed to the Protagonist's end goal and tries to undermine his success.
- The *Logic* personality is calm, perhaps even cold. He makes decisions and takes actions wholly on the basis of logic.
- The *Emotion* personality is reactive, seemingly uncontrolled, disorganized, mainly driven by feelings and moods.

As in story telling, additional secondary personalities could be added to complete the character such as an *Obstacle* personality which tries to block the ways or a *Sidekick* which is a faithful supporter of any of the other personalities. In our experiment, we have only implemented the four main personalities. The emotion one is based on both the previous emotional model of Eva [3] and the ALMA hierarchical model [11].

3. Memory and web mining

They are two additional groups of nano-agents that implement memory and web mining functionalities. The memory group is responsible for storing and retrieving information when needed. It stores all interactions with the user in log files and an indexer periodically parses these log files and extracts keywords. A dedicated plugin package adds to nanoScheme a small set of functions for retrieving information in the memory [12]. When the character has no information on a specific question, the web mining group tries to find a pertinent answer by searching information on the web [12].



Fig.1. The virtual character interface.

IV. RESULTS AND DISCUSSION

In this section, we discuss a typical interaction experiment with Alice [4] and Eva on the same simple interaction scenario. We obtained the following result with Alice:

- (1) User: hi
Alice: Hello there.
- (2) User: how are you today?
Alice: My logic and cognitive functions are normal.
- (3) User: what is your name?
Alice: My name is ALICE.
- (4) User: how old are you?
Alice: I am the product of over five years' research.
- (5) User: Can you help me?
Alice: What kind of help would you like?
- (6) User: What are clinical trials?
Alice: Have you tried a web search?
- (7) User: Thank you.
Alice: you are quite welcome.
- (8) User: bye
Alice: Sayonara.

We can note the quality of answers in the classical introductory and concluding discussion phase. However, even if its knowledge base includes a lot of general culture information, Alice was not able to answer to the

very specific question about clinical trials and suggest the user to try a web search.

We obtained the following result with the multiple personalities of Eva :

- (1) User: hi
Eva: Hello.
- (2) User: how are you today?
Eva: Not too bad for a virtual creature.
- (3) User: what is your name?
Eva: You may call me Eva.
- (4) User: how old are you?
Eva : Just an advice: never ask such a question to a woman.
- (5) User: Can you help me?
Eva: I will do my best. What do you want?
- (6) User: What are clinical trials? [...]
Eva: Researchers use interventional clinical trials to test hypotheses about the effect of a particular intervention upon a pathological disease condition.
- (7) User: Thank you.
Eva: You're welcome.
- (8) User: bye
Eva: Goodbye.

As for Alice, the Eva introductory and concluding phases are good. In (1), (3), (5), (7) and (8) the Protagonist personality answers have been selected, while in (2) and (4), the answers were from the Emotion personality. The question (6) has been processed by the web mining nano-agents through a search using Wikipedia. The answer in this specific case is very pertinent. However, for a more ambiguous question, the answer is not so convincing [12]. Another problem is that, in most cases, the user must wait for few seconds between his questions and the answer (shown by [...] in the interaction). This delay is due to the time required to access Internet, make the search query and compute the answer. This could be solved by enabling the schizophrenic model to continue interactions with the user while searching on the web. Another evident critic is that the interaction case reported here is too short and simple to let all the personalities express themselves in the flow of conversation.

V. CONCLUSION

Eva is a long term open architecture project for designing artificial creatures [13]. Future works include

(1) a deeper study of the schizophrenic model for creating rich believable characters, (2) enabling interactions not only with the user but between the multiple personalities, (3) adding much more personalities to create a swarm. Another direction is to develop more robust web mining and memory functionalities using evolutionary programming and swarm algorithms. Our goal is to reduce the amount of programming required by learning information from the flow of conversation and from the web rather than coding a large amount of predefined knowledge.

The Eva bio-inspired architecture has obvious applications for designing intelligent agents for commercial web sites and marketing studies. However, its lightweight nano-agent architecture enables a wider spectrum of applications. We like to imagine virtual assistants on mobile phones, assistants for lone aged and/or sick people, for learning foreign languages, virtual characters in video games and, of course, for robotic and embedded applications.

REFERENCES

- [1] Heudin JC, Burg B, Zavidovique B (1986), A flexible operating system for distributed process control. 4th SPIE Conf. on Applications of AI
- [2] Gilles A, Lebiannic Y, Montet P, Heudin JC (1991), A parallel architecture for the Electronic Copilote. 11th EC2 Int. Conf. on Expert Systems Applications
- [3] Heudin JC (2004), Evolutionary Virtual Agent. ACM/WIC/IEEE Intelligent Agent Technology Conf. 93-98
- [4] <http://www.alicebot.org>
- [5] Clinger W, Rees J (Eds.) (1991), Revised(4) Report on the Algorithmic Language Scheme
- [6] Ray TS (1991), An approach to the synthesis of life. Artificial Life II, Addison-Wesley, 371-408
- [7] Koza JR (1992), Genetic programming. MIT Press.
- [8] Heudin JC (2007), Evolutionary Virtual Agent at an exhibition. Virtual Systems and Multimedia Conf.
- [9] <http://en.wikipedia.org/wiki/Schizophrenia>
- [10] Masterson L (2000), Casting your character. <http://www.fictionfactor.com>
- [11] Gebhard P (2005), ALMA: A layered model of affect. 4th Int. Joint Conf. on Autonomous Agents and Multi-agent Systems, 29-36
- [12] Millet P, Heudin JC (2007), Web mining in the EVA intelligent agent architecture. ACM/WIC/IEEE Intelligent Agent Technology Conf.
- [13] <http://www.artificial-creature.com>