Embodiment of Evolutionary Computation in Network Agents

Robert E. Smith
The Intelligent Computing Systems Centre
The University of The West of England
Bristol, UK
email: robert.smith@uwe.ac.uk

1. INTRODUCTION

There is ongoing shift in the landscape of personal computation and telecommunication. Individuals are communicating with each other in increasingly complex ways. In the very near future, large number of software agents may be acting on behalf of any given individual. In the not-so-distant future, even relatively simple household appliances may interact in a massive, churning telecommunication network, filled with interacting agents.

The identifying qualities of agent-based components are that they exhibit autonomy, reactivity, proactivity, and social behavior. Moreover, agents have the possibility of mobility in complex network environments, putting software functions near the computational resources they require. Agents can also explicitly exploit the availability of distributed, parallel computation facilities (Wooldridge & Jennings, 1996).

Each of the agent features mentioned above suggests agents that adapt. However, in agent-based systems, one must consider not only adaptation at the individual level, but at the system level. When each agent is responding and adapting to local forces, the overall goal of adaptation is an *emergent* effect of enterprise adaptation at the highest level.

This begs the question: How does one obtain desirable, emergent, adaptive behaviour at a system level, in a world where large numbers of distributed, self-interested agents are interacting in complex, massively parallel, and only partially regulated ways? Fortunately, there are several extant paradigms that are directed towards the goal of understanding and encouraging desirable, emergent behavior from complex systems of self-interested individuals. One such paradigm is *Evolutionary Computation* (EC).

2. EC THEORY AND AGENTS

One of the foundations of EC theory is John Holland's seminal book, *Adaptation in Natural and Artificial Systems* (Holland, 1975). The book is directed at general issues of adaptation in systems of distributed, self-interested agents. Holland mathematically examines how populations of self-interested, autonomous agents interacting in systems can, and should adapt, for system-level, desirable, emergent effect.

In Holland's framework, the "agents" are (natural or artificial) genetic actors, agent features are genes, and the agent messages are analogous to gametes. Seen in this way, Holland's "genetics-like" models extend to more general (non-genetic) agents, and agent interactions. If agents can be considered to be gleaning any sort of feature information from messages, and if the information is used to create, destroy, or simply change agents, Holland's models continue to apply. This is the case even if there is no explicit (centralized or decentralized) "GA" involved in the system of agents.

Clearly, there is a need to investigate the implications of EC theory on agent-based systems. However, one should ask whether typical EC systems are appropriate for this investigation. To implement an EC system, a programmer would typically write computer code that creates a population of EC entities, and manipulates those entities in a pre-specified way. The pre-specifications of behavior in this code are a sort of "bubble" around the EC entities.

Any emergent behavior takes place within the constraints of that bubble. Even in advanced, network based EC systems, Tierra (Ray, 199), there are server programs that control and pre-specify the nature of agent interactions, in a way that is entirely specific to the simulation. The "bubble" around typical EC may limit the transfer of extant EC theory and practice to agent-based systems.

This observation can be related to the argument for *embodiment* in intelligent robotics (Brooks, Breazeal, Irie, Kemp, Marjanovic, Scassellat & Williamson, 1998). This argument would state that one cannot remove the intelligence of an agent from the sensors, actuators, and environment of that agent. This paper presents an agent-based EC system, that follows a philosophy of *software embodiment*. This manifests itself in two ways:

1. The system presented is designed to be composed of distinct, autonomous agents, and their behaviors, rather than around controlling systems or servers.

2. The system attempts to conform to emerging agent-based software standards.

Effectively, this "bursts the bubble" around EC technology, so that we can examine the embodied issues of emergence in independent, self-interested, real-world agents (Figure 1).

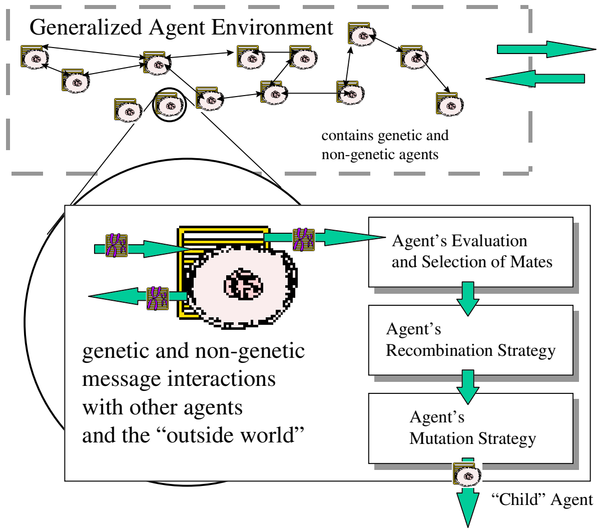

Figure 1: Embodied EC agents in a general agent world.

3. A FRAMEWORK FOR EMBODIED EC AGENTS

This paper introduces *Egglets*, a framework for embodiment of EC in standards-based agents. The `Egglet` framework is built in Java, for its machine independence, and on IBM's Aglets framework, for basic agent capabilities (e.g., agent messaging, agent storage and recall, mobility, etc.). The framework is transferable to other agent systems as well.

There are two basic elements of the framework. The first is a set of Java interfaces for general EC objects. Key interfaces are Randomizable, Mutatable, Gene, Genotype, Gamete, Sperm, and Egg. The mnemonic interpretations of these interfaces are obvious. However, note that these are general purpose interfaces, that can be used to construct a range of EC data structures (e.g., GA, GP, ES, etc.)

The second element of the framework is a set of mate-seeking and reproducing behaviors for individual agents.

When combined, these elements yield a set of autonomous agents that have the capability to pursue individual genetic adaptive plans, with *emergent* EC effects. For instance, Figure 2 shows the effect of 50 autonomous, asynchronous, interacting agents who have a mating goal equivalent to the "OneMax" problem

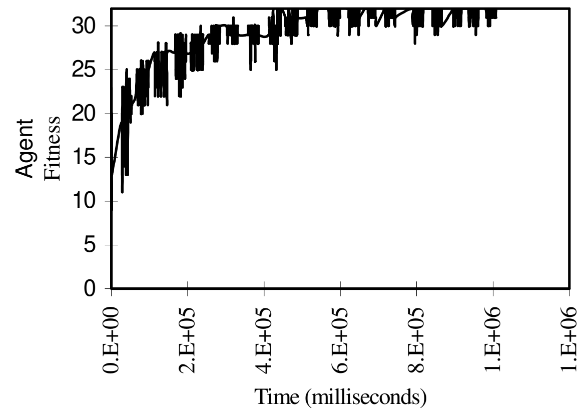

Figure 2: Emergent EC effects from autonomous agents.

4. FINAL COMMENTS

The agent based EC framework presented here suggests several areas for future investigation. These include; collaborative EC-adaptive work (search, problem solving, etc.) between distributed users and agents (without a priori knowledge of the collaborations), and a variety of EC-scientific experiments, including emergent speciation, self-adaptive GA operators, etc.

The Egglet framework design also allows for diverse agents that can use a variety of genetic representations and operators, while interacting in the same distributed computational environment. Finally, the framework presented establishes a way of using EC techniques within agents that are based on emerging system standards. This will hopefully increase appropriate and productive use of EC techniques in real world telecommunications systems.

5. REFERENCES

Aglets Workbench. http://www.trl.ibm.co.jp/aglets/

Brooks, R.A., C. Breazeal (Ferrell), R. Irie, C. Kemp, M. Marjanovic, B. Scassellat and M. Williamson (1998). Alternate essences of intelligence. To appear in *Proceedings of AAAI-98 Conference*. AAAI Press.

Holland, J. H. (1975). Adaptation in natural and artificial systems. The University of Michigan Press. Ann Arbor.

Ray, Thomas (1990). An approach to the synthesis of life. In Langton, C., Taylor, C. Farmer, J. D. and Rasmussen, S. (eds.) *Artificial Life II*. Addison-Wesley. p. 371-408.

Smith, R. E. and Taylor, N. (1998). A framework for evolutionary computation in agent-based systems. In C. Looney and J. Castaing (eds.) *Proceedings of the 1998 International Conference on Intelligent Systems*. ISCA Press. p. 221-224

Wooldridge and Jennings (1996). Software agents. IEE Review. January 1996.p. 17-20.