Constructing Federations from Simple Agents and Environment Contexts

(Extended Abstract)

Dmitri O. Kondratiev, Vasily V. Suvorov

R&D, Luxoft, 9-b Dmitrovskoye shosse
Moscow 127434, Russia.
dkondratiev@ luxoft.com, vsuvorov@luxoft.com

Introduction

Multi-agent approach to simulation modeling suggests an environment where agents can communicate and evolve. Properties of agents and environments vary considerably depending on modeling domain. On the other hand, such properties of agents as emergent behavior and adaptation to environment through evolution are of general value in multiple domains. To realize these fundamental properties the core set of agent and envi ronment features needs to be defined.

In this paper we make an attempt to define this core set of agent / environment features and investigate the possibility to realize them with elementary building blocks that agents can be built from. We propose core agent architecture that should allow building agent federations according to different decomposition principles from **simple agents** (later in this paper referred to simply as agents). Federations of interest for this research include agent swarms and hierarchies of recursive and self-similar agents or "holons". As described in [GeSiVi99] holon is "a biological or sociological structure that is stable and coherent and that consists of further holons that function according to similar principles. No natural structure is either ``whole'' or ``part'' in an absolute sense, instead every holon is a composition of subordinate parts as well as a part of a larger whole." With well-defined building blocks of software architecture a group of several holonic agents may form a super-holon. This super-holon looks to the outside world like a single holon agent defined in terms of the same architectural blocks. On the other hand, core architecture proposed here does not impose any additional constraints on autonomy of sub-holons that holonic architecture [GeSiVi99] does in terms of goals sharing and resource management between super-holon and sub-holons.

The main goal of our work is to define minimal set of features that can be used across various domains to construct agents of arbitrary complexity. We plan to have a very wide range of creatures to be modeled with agents: from memory cells and neurons to swarms and simple robots.

With these building blocks defined, we further develop a set of general modeling primitives for specialized programming languages that will allow agents themselves express their behavior with and affect the environment. We develop very regular and fully reflective architecture where agents react to stimuli, modify their state and affect other agents through the environment (which is also an agent). The main idea is that agents themselves can now modify their own "behavior programs" in response to external stimuli and thus evolve and adapt to changing environment. With this architecture in place, agents get the ability to build their own models of the environment and as a result become learning agents.

Having general modeling primitives defined, we plan to build specialized agent programming languages to be used by agents from different ALife paradigms, such as various forms of evolutionary computations (GA, GP, co-evolution), neural nets and symbolic AI as well. This would allow creating environments where agents built with different ALife technologies and their combinations will be able to co-exist, thus using the strongest features of different methods as needed and to their full advantage.

Environment as Agent Context

In our framework we define a notion of "live context" as a property of every agent environment or every agent world. Thus any environment (world) is modeled as a set of "live contexts" managed by context controlling agents.

We define **live context** (or simply context) as a set of features of the environment that some agents may be interested in. One and the same environment can have many associated contexts. At any given time some set of agents may be interested in particular set of contexts from different environments.

Live context is a function of time. Context feature set may change in time – features may be added to or removed from context at any discrete moment during context lifetime. Agents always communicate with each other in "some context". In our system every agent controls zero or more live contexts of environment. In trivial case agent does not control any environment context, being a self-contained entity.

In non-trivial case agent mediates all communications of other agents participating in this particular agent context. Agent fully controls its life context defining mediation rules and mechanisms for information exchange for other participating agents interested in this agent context. In this regard we talk about "context controlling" agent and sub-agents controlled by this agent.

Contexts allow building different types of agent federations. For example, agent hierarchy may be modeled as a tree of agents with two contexts controlled by every node agent. One context in node agent represents environment for sub-node agents down the tree. Sub-agents will use this context as collaboration environment to realize complex behavior as one super-agent. Second context in the same node agent may then abstract its sub-tree as another environment for agents on the next, higher (closer to root) level of the same tree. In this case first context models internal environment of agent federation, while second context presents this federation to the outside world as one entity, abstracting its internal architecture. Note, that with this approach agent federations can be created of arbitrary structures, not necessarily as trees.

Building Blocks of Agent

The main building blocks of an agent defined in our framework are **sensors**, **visible traits** and **behaviors**. Agent uses sensors to get information about some environment (getting values of live context feature set). Agent may define separate sets of sensors for every context it is interested in. Agent may influence its environment (setting values of live context feature set) using its visible traits (or simply traits). Traits are agent attributes observable in some live context by other agents. Agent may define separate sets of traits for every context it participates in. Context controlling agent represents traits of all participating sub-agents within this context by means of feature sets. Any agent may observe any trait of another agent partic ipating in the same context. Thus traits and sensors are defined as properties of live context. To observe any trait or a group of traits agent creates appropriate sensor or a group of sensors. Agent can build sensors that may react either to specific trait (context feature) or a group of traits (context feature set) making generalization of sensor possible. Both sensors and traits may be added / removed by owning agent in run-time, which results in corresponding changes of context feature set.

Evolving Agent Behaviors

Agent behavior is defined as a function of sensor set and time with values represented by agent traits. Behaviors can be defined both for a single context (with sensors and traits belonging to the same context), as well as those spanning several contexts (with some sensors and traits in different contexts).

As a first step in our ongoing work with specialized modeling languages, we develop behavior description language allowing agent express its behavior in Scheme. In this approach behavior programs allow agent create sensors, traits, read sensor values, process these values and write output trait values. Using Scheme also allows agent in run-time (without recompilation) rewrite its behavior programs and thus adapt to changing environment contexts. To achieve this functionality we develop our framework in Java and use Jscheme (an implementation of Scheme in Java) to specify agent behavior programs.

Bibliography

[Fs97] Franklin, S.: Artificial Minds, MIT Press, 1997

[GeSiVi99] Gerber, C.; Siekmann, J.; Vierke,G.: Holonic Multi-Agent Systems, DFKI-RR-99-01

[BeCaPaRa00] Bedau, M.; McCaskill, J.; Packard, N.; Rasmussen,S.: Artificial Life VII, Proceedings of the Seventh International Conference on Artificial Life, The MIT Press, Cambridge, Massachusetts, London, England, 2000

[Wo2000] Wooldridge, M.: Reasoning about Rational Agents, The MIT Press, Cambridge, Massachusetts, London, England, 2000

[Ka88] Kanerva, P: Sparse Distributed Memory, The MIT Press, Cambridge, Massachusetts, London, England, 1988