The Perplexus Programming Framework: Combining Bio-inspiration and Agent-Oriented Programming for the Simulation of Large Scale Complex Systems

Olivier Brousse, Gilles Sassatelli, Thierry Gil, Michel Robert, François Grize, Eduardo Sanchez, Andrés Upegui, Yann Thoma

The Perplexus Programming Framework: Combining Bio-inspiration and Agent-Oriented Programming for the Simulation of Large Scale Complex Systems

Olivier Brousse[1], Gilles Sassatelli[1], Thierry Gil[1], Michel Robert[1], François Grize[2], Eduardo Sanchez[3], Andrés Upegui[3], and Yann Thoma[3]

[1] LIRMM, Univ. Montpellier 2, CNRS. 161 rue Ada, 34392 Montpellier, France
firstname.lastname@lirmm.fr
[2] UNIL, Switzerland
Francois.Grize@unil.ch
[3] HEIG-VD, Switzerland
firstname.lastname@heig-vd.ch

Abstract. This paper presents a flexible agent-oriented programming framework that provides native support for bio-inspired mechanisms. This solution, developed within the Perplexus IST European project [IST-034632][1] aims at providing a means for the specification of applications running on a distributed and pervasive network of mobile nodes. In such applications, the deployed systems may face time-changing environments and bio-inspiration may prove useful bringing self-adaptability to the system. The presented framework features are demonstrated on a proof-of-concept application made of simple robots that autonomously improve their behaviour over time.

Keywords: Bio-inspired, Scalability, Distributed platform, Ad-hoc Network, Multi-Agent Framework.

1 Introduction

The Perplexus Project aims at developing a scalable and ubiquitous platform endowed with bio-inspired features to simulate complex phenomena.

The developed platform is composed of a cluster of communicating nodes called UBIDULES (UBIquitous moDULES). These nodes have a microprocessor and a specific reconfigurable integrated circuit; both providing support to distributed bio-inspired mechanisms. Furthermore, application-specific hardware sensors and actuators may be added to any Ubidule for application-specific purposes (environmental interactions). The Perplexus Ubidules integrate at least one bio-inspired chip and a XScale PXA270 microprocessor that is in charge of networking, sensors, actuators and more generally software aspects. Figure 1 gives an overview of

[1] [IST-2006-034632] STREP IST project web site: http://www.perplexus.org

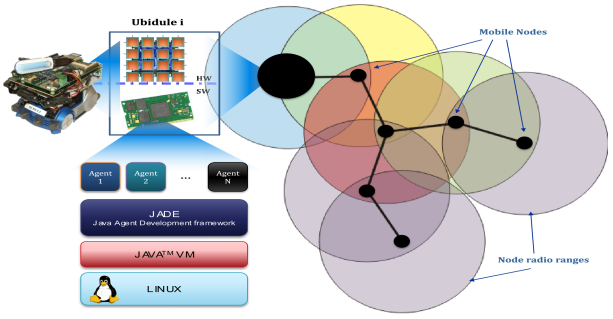

Fig. 1. Perplexus platform overview

the obtained platform which is represented in form of a network of Ubidules. Each Ubidule is in turn represented as a unit made of a hardware partition and a software partition. An embedded Linux operating system runs on the microprocessors and handles most high-level functionalities.

1.1 Bio-inspired Mechanisms

Bio-inspiration has found applications in many areas ranging from finance to integrated circuits designs. Many theories exist regarding the emergence of life and evolution of biologic lifeforms; however it is often considered that three main mechanisms exist:

1. *Phylogenesis* deals with the evolution of a set of species. Evolution gears species towards a better adaptation of individuals to their environment; genetic algorithms are inspired from this very principle of life.
2. *Ontogenesis* describes the origin and the development of an organism from the fertilized egg to its mature form. Biological processes like healing and fault tolerance are qualified of ontogenetic.
3. *Epigenesis* refers to features that are not related to the underlying DNA sequence of an organism. Learning as of performed by Artificial Neural Networks (ANN) is a process which scope remains limited to an individual lifetime and therefore is Epigenetic.

1.2 Application Case-Study

Among the many challenges that form this project, this paper puts focus on the work realized at the platform level for enabling the implementation of distributed applications that take advantage of the bio-inspired features of the platform. This work relies on an agent-based programming methodology and a programming framework made of resident agents which provide support for the three bio-inspired mechanisms presented previously. Three applications are proposed within the project. In this paper we focus on a so-called embodied application (i.e. robotic application) that takes advantage of learning and evolution capabilities of the platform. This case-study application is intended to be

a proof of concept of the proposed approach; it uses the pervasive capabilities of the Perplexus platform for it relies on a network of robots that communicate wirelessly. Each robot is equipped with a prototype Ubidule board that handles the bio-inspired mechanisms such as learning, thanks to a software artificial neural network, and evolution that helps improving robot behaviour over time.

2 The FIPA Multi-agent System

The distributiveness and capabilities of the Perplexus platform greatly rely on the Agent-oriented programming (AOP) style. AOP derives from the initial theory of agent orientation which was first proposed by Yoav Shoham [1]. Agent-orientation was initially defined for promoting a social view of computing and finds natural applications in areas such as Artificial Intelligence or social behaviours modeling. This programming format infers object-oriented programming by endowing objects with additional characteristics; they are viewed as entities which exhibit behaviours, capabilities and are entitled to take decisions.

In an AOP systems Agents exclusively communicate with each other through typed messages. The type of these messages vary very much depending on the nature of the exchanged information. Agents may be informing, requesting, offering, accepting, and rejecting requests, services or any other type of information. AOP furthermore sets constraints on the parameters defining the state of the Agent (beliefs, commitments and choices). These constraints actually define the Agent Oriented computational system which is then viewed as a set of communicating software modules that exhibit a certain degree of awareness. These characteristics naturally geared the PERPLEXUS modeling framework towards AOP which fits perfectly the objectives of the platform.

There exists various Multi-Agent Platforms that allow to develop agent based applications, such as JADE [2], JXTA, FIPA OS, JAX or MADKIT. JADE is a Java environment and is therefore portable, FIPA[2] compliant and exists in a lightweight version called LEAP for Light Extensible Agent Platform [3] that suits our hardware restrictions (PXA270) as it is running with Sun Microsystem J2ME (Java2 Micro Edition) JAVA Virtual Machine. Agents in a JADE Framework "live" in containers. Both containers and Agent Platforms (APs) may span several hardware hosts. Agents existing inside a given container may very well communicate with agents of other containers and even platforms, through a TCP/IP network using the Message Transport Protocols (MTP).

3 Perplexus Agent Framework

The here presented Bio-inspired Agent Framework is built on top of a JADE agent platform that has been ported to the Ubidule sotware environment. Figure 2 shows the proposed framework; shaded components correspond to platform specific features while others are part of the JADE environment. Hence, the Directory

[2] http://www.fipa.org/

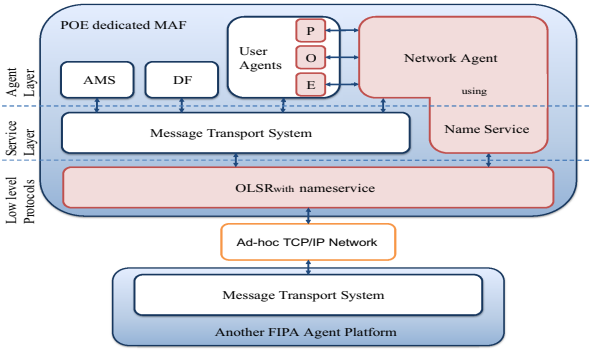

Fig. 2. Bio-inspired Framework

Facilitator (DF) and Agent Management System (AMS) respectively provide yellow-pages services and agent managing services to the system. Next to these two mandatory agents are three fundamental agents of the PERPLEXUS framework.

1. *The Phylogenetic agent (P)* handles all evolutionary-related processes of the actual Ubidule. It is responsible of evaluating the fitness of the embodied agent and also participates to the distributed genetic algorithm execution that eventually results in the installation of a new population of functional individuals in the platform.
2. *The Ontogenetic agent (O)* is responsible of growth-related mechanisms in the system. Particularly, it handles the process that instantiates new agents when a new generation is being installed
3. *The Epigenetic agent (E)* handles all non-evolutionary related mechanisms and therefore mostly artificial neural networks that dictate the behaviour of the actual agent

As depicted in figure 2, all messages transmitted or received by the local agents are channelled through the Network agent (N). This agent provides services of two classes:

1. Agent-level, that facilitate various platform-level operations that may prove useful or necessary for some applications. Among the proposed services, event collection, peer discovery, timestamping or semaphore resource access find natural use in many cases.
2. Service level, that take care of ensuring communication reliability in close interaction with the low-level layer discussed below. Provided services aims at answering various needs such as keeping an up-to-date list of accessible nodes in the platform, resolving peers names (nameservice) or gathering information from the physical network topology such as identifying neighbours, etc.

The Ubidules forming an ad-hoc network of mobile nodes, some dedicated mechanisms are mandatory for ensuring network reliability. These mechanisms are integrated in the lower-level layer that implements the Optimized Source Link

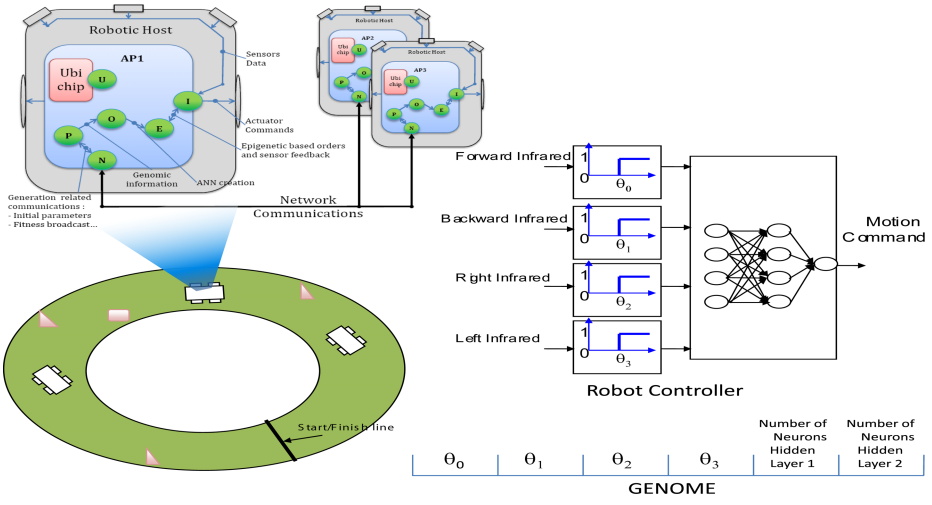

Fig. 3. Sofware application example

State Routing (OLSR [4]). This adaptive routing protocol dynamically changes routes for maintaining the logical network structure that allow any node to contact any other (multi-hop routing). The network agent makes exclusive use of some services of OLSR for retrieving physical network information; for some of its services such as neighbours discovery.

3.1 Validation Application

In order to prove the reliability of the platform, a simple validation application based on a race of robotic toys was developed (Fig. 3). Robots which participate use learning (Epigenesis) for improving their performance which is expressed as lap time. Furthermore, the robot population is evolved (Phylogenesis) over multiple generations in order to obtain a species that is better adapted to the problem. The individuals have an average lifetime that is set by the application; whenever an individual dies a new one resulting from the Phylogenetic process is grown (Ontogenesis) onto the same robot. Each robot is equipped with four infrared sensors required for the perception of its environment; these sensors are well-arranged on the four sides of the robot and compute the distance with the nearest obstacle. The robots are capable of executing four basic motions that can be combined: moving forward (MF), moving backward (MB), turning right (TR) or turning left (TL); the Interface agent I provides methods to access to these sensors and actuators. Later, an Ubicom agent U will be implemented in order to provide methods dedicated to the management of the bio-inspired circuit. In our application, the population is composed of three robots, the genome (Fig. 3) of a robot is composed of the values of four thresholds which are used to convert the analog information issued by the infrared sensors into a binary word representation; the data provided by the infrared sensors are compared to the

values of the threshold contained in a genome, if a value is lower than the specified threshold, then an obstacle is detected otherwise it is not. The fitness function reports the quality of the expected solution; in our application the fitness is represented by the lap time. Figure 3 shows a schematic representation of the robot controller we have implemented. The output of the system which issues the motion commands (MF, MB, TR, TL) are computed by a software ANN which receives as inputs the four boolean informations of the proximity sensors. This controller represents the behaviour of the Epigenetic agent of our multi agent platform. The genomic description of an individual is made of an array containing all threshold values that are used by the sensors (expresses sensorial abilities) and the specificities of the ANN. In our application, the genome is coded as a 48-bit string as described in Figure 3. A demonstration video is available online at: `http://www.lirmm.fr/~brousse/Ubibots`.

4 Conclusion

This paper presents some of the work realized within the confines of the Perplexus Project. These contributions provide a reliable middleware to support both bio-inspiration and agent-oriented programming for distributed pervasive platforms. The proposed framework has been designed with broader application fields in mind and should prove appropriate for many sensor-network applications where adaptability brings advantages. This software environment has been tested through a POE (Phylogenesis, Ontogenesis, Epigenesis) robotic application that proves the reliability of the platform with satisfying result concerning communication and basic POE computation. Future work relies on enabling transparent migration of some software agents (JADE agents) to the bio-inspired integrated circuit in order to increase the performances of the platform.

Acknowledgements

This project is funded by the Future and Emerging Technologies programme IST-STREP of the European Community, under grant IST-034632 (PERPLEXUS). The information provided is the sole responsibility of the authors and does not reflect the Communitys opinion. The Community is not responsible for any use that might be made of data appearing in this publication.

References

1. Shoham, Y.: Agent Oriented Programming Journal of Artificial Intelligence 60, 51-92 (1996)
2. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing Multi-Agent Systems with JADE. Wiley, Chichester (2007)
3. Lawrence, J.: LEAP into Ad-Hoc Networks. In: ACM Workshop on Agents in Ubiquitous and Wearable Computing, AAMAS (2002)
4. Jacquet, P., et al.: Optimized link State Routing protocoll for Ad-Hoc Networks, INRIA Roquencourt, HiPERCOM project (2001)