*IDEA GROUP PUBLISHING* **ITB13193**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This paper appears in the publication, *An Imitation-Based Approach to Modeling Homogoneous Agents Societies* **by Goran Trajkovski** © 2007, Idea Group Inc.

## Chapter XII

# On a Robotic Platform for MASIVE-Like Experiments

with Brian Whitman, Towson University, USA

## Abstract

*This chapter overviews a robotic platform developed at our Cognitive Agency and Robotics Laboratory (CARoL, n.d.) for the purpose of carrying out Interactivist-Expectative Theory of Agency and Learning (IETAL) and Multi-Agent Simulated Interactive Virtual Environments (MASIVE)-like experiments in a realistic environment. Performing IETAL and MASIVE-like experiments with a robotic agent(s) requires specialized agent(s) in a specialized environment. The solution overviewed here is done on a shoestring budget and is easy to replicate and modify.*

## Introduction

As a programmer some of the most rewarding code one can write is the code that autonomously interacts with the world. Like many creative processes in life, the more rewarding programming projects are more difficult. Designing and imple-

menting a platform for robotics experimentation is no different. In fact, in some ways it is one of the more difficult projects. It is very much a merger of an embedded project and a framework project — one inherently small, one typically big. The embedded aspect of the project is fairly obvious; autonomous robots that operate for any useful amount of time must be energy conscious. In addition, the wide variety of sensors and sensor interfaces used in robotics requires access to interfaces not typically available on personal computers. The embedded solutions available cater to this type of project and frequently provide low power consumption systems with interfaces not supported by more mundane hardware. In juxtaposition to this is the reusable framework portion of the project. One goal is to be able to easily swap devices and experiments without having to reinvent the basic components for each project. Two of the drawbacks often encountered with application frameworks is the structure overhead as well as code overhead. In an embedded environment this type of overhead can become problematic when memory space becomes limited. A framework is essential for creating reusable components that can be ported to multiple platforms; it provides a structure for how components should be implemented so that they can be used and reused easily. Why use embedded? Our overall goal in designing a reusable platform was to make it easier to utilize all the varied hardware we have in the lab (CARoL, n.d.), and more importantly to provide a way to quickly implement autonomous robotic experiments. It turns out embedded systems have many advantages when the design calls for extended autonomous runtimes.

## Hardware Design

This section overviews the hardware platform of our robotic agents.

In robotics there is a mechanical side, therefore size matters. As we will later see, size matters for code too, but in hardware mechanical terms, the bigger the robot the more energy required to move it. There exists some very nice robot assemblies whose size forces them to use more energy than our projects need; ER1 (EvolutionaryRobotics, n.d.) and Pioneer (ActivRobotics, n.d.) are two excellent robot platforms that are just too massive for many of our projects. In addition, some, like ER1, require a laptop. At first thought a laptop might not seem like a bad selection for processing in a robot platform, and while it is true modern laptops are small and very powerful, they may be too powerful for many applications. Anyone who has ever used a modern laptop in his lap knows that in under load conditions laptops can generate an enormous amount of heat. That means it is also consuming a lot of energy. A typical laptop power supply is 19