

AC 2008-1827: AN "EIA" APPROACH TO SUPPORT LABORATORY LEARNING ENVIRONMENTS

Nabil Lehlou, University of Arkansas

Nebil Buyurgan, University of Arkansas

Justin Chimka, University of Arkansas

An EiA Approach to Support Laboratory Learning Environments

Abstract

When developing or expanding hands-on laboratory environments that rely on technology, one faces various challenges. Such inconvenience varies from expensive technological renovations to the reliance of devices on human intervention, to the non-standardized communication between networked objects that use different native programming languages. To overcome these specific problems, an “Everything is Alive” (EiA) approach is proposed. To explore the potentials and investigate the effectiveness and usefulness of this strategy in hands-on laboratory environments, the idea is implemented and tested on a laboratory system that uses RFID equipment, servers and databases, and moving mechanisms. The set of the different involved agents include RFID readers of different brands, motorized RFID tags and antennas, remote databases that store the RFID reads, and friendly Graphical User Interfaces (GUIs). After constructing the structure and its framework, students and interested individuals are able to change RFID experiments’ setups, control different types of RFID readers, gather the read data, perform computational processes; all is carried out remotely and through one easy-to-use interface. Thusly, educational tools can be built on top of this agent infrastructure and teaching modules can be embedded to the GUI, yielding a creative way to convey the technical knowledge. Furthermore, since they are independent and easy to integrate, agents can be separately developed by advanced-level students in academic milieus, a fact that leads to considerable practical experience. Partial support for this work was provided by the NSF CCLI program.

Everything is Alive

EiA Concept

Some people envision the future as an environment where people can talk to all objects. A person can tell the lights to switch on, a house can tell a police station about a threat or robbery, or even more extreme, the soil can tell the water that it is thirsty. This vision became an inspiration for one of the research topics: “Everything is alive¹,” also known as EiA. “Everything is Alive” is a concept that says that all objects in the world can be connected to a network called the “Internet of Things²” and be intelligent enough to communicate with humans as well as other objects without human intervention. Once these abilities are acquired by an object, it is called an “EiA agent,” and it is said alive because it can interact with humans and agents.

While many people think that the vision of the future is just a dream and may not be feasible at all, EiA does have a plan to make it come true. Ideally, one can integrate a microchip into an object, program it to create some kind of intelligence, then, plug this “thing” to the ubiquitous “internet.” This entity therefore becomes an agent that is ready to communicate with its outside world. However, this idea needs a while before being implemented because of the cost of the integrated microchips and their installation. Fortunately, there is a cheaper way to achieve our EiA goal. The solution is to use computers for they are powerful and can connect to the internet. Now, imagine that XML, which is a structured programming language used in the web technology, is chosen to be the standard language that every agent speaks. One can then write a

computer program that acts like an interpreter from XML to the native language of a device and vice-versa. In other words, the hardware device is wrapped with a software translator, and therefore, mutated into an agent that can speak XML to other agents. So, where do we go from here?

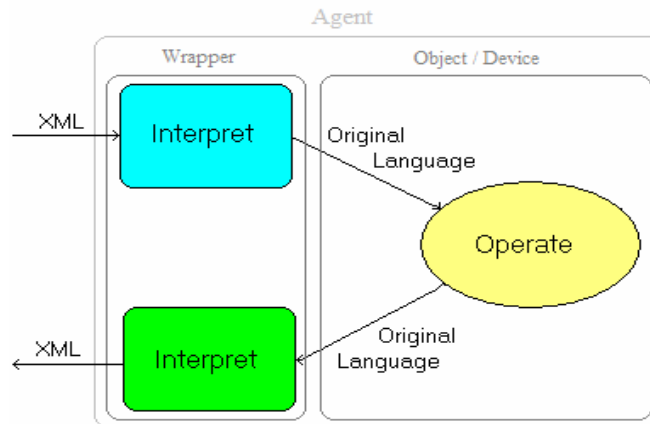


Figure 1 Illustrating how an object operates after it is wrapped by an EiA interpreter

EiA Benefits

Let's start with one of the most popular areas in engineering: system design. Systems that involve software can greatly benefit from EiA, because by its nature, EiA leads to constructing agents that are independent of the outside world. To clarify this, think of a system as a set of independent, but interrelated, agents. If the system breaks, only the responsible agents are fixed or replaced. If the system needs to perform better, only the inefficient agents are upgraded. If the system requires an expansion, one or more agents are added to the system, without affecting the existing ones. One can see that EiA provides engineers with a way to architect a system that is simple, robust, easy to maintain, and flexible to change. Another advantage gained when using an EiA approach to build a system is the ability to communicate with a specific type of devices through one unified interface, even though the devices' brands might be different. For instance, RFID readers (discussed in the next section) that are supplied from different manufacturers have different control interfaces. Nevertheless, after these different devices are wrapped by EiA interpreters, one software interface is enough to control them all.

RFID Laboratory Development

Until at least a small portion of EiA is implemented and tested, it stays only a theory. We thusly decide to test our concept in an application that is useful to the engineering world because we are more likely to obtain support. In fact, we found interested students who want to work with us as well as obtained funding from the NSF CCLI program. Our chosen project is developing a laboratory learning environment and the selected technology is RFID.

RFID Technology

Radio Frequency Identification (RFID) is a data collection technology that utilizes a wireless radio communication (radio frequency signals) to identify, track, and categorize objects. The basic RFID system consists of three main components:

- The RFID reader, which by itself contains the processing unit, antennas, and the cables joining them; its main task is send electromagnetic waves to the surrounding environment and listen for electromagnetic responses from the RFID tags. Upon receipt of the tags' data, the reader submits the RFID reads to the target database.
- The RFID tag, which is a microchip that is bound to a small antenna and that transmits the data stored in it as the electromagnetic response to the reader.
- The database where all the raw read data is to be amassed, and most likely converted into meaningful numbers and patterns.

This system can be extended with a set of middleware devices, a variety of software applications, a network of readers, and a powerful database management system (DBMS) to facilitate data acquisition and management.

Laboratory System

Now that the reader is a bit familiar with RFID, it is a good idea to start envisioning what the RFID laboratory system looks like. In other words, one should look for the different agents that form our system:

- RFID reader agent: upon reading a tag, it sends the read data to the database (DB) agent.
- DB agent: it stores the RFID reads and provides the GUI agents with RFID data when a query request is received.
- Graphical User Interface (GUI) agent: this lets the user query the DB agents and control the other agents.
- Movable agents that carry RFID tags: these can be any digitally controllable devices that move in the laboratory.

The first two agents are relatively easy to build, since the hardware, which is purchased from the manufacturer, already supports messaging over the network. The only thing that is needed is to encode those internet messages in XML. The GUI agent may take some time to build since it is subject to different requirements, tastes, upgrades, new agent entries to the system... but it is also reasonably doable. The main idea of it is to link the GUI's buttons to XML message generation mechanism on the one hand, and on the other hand, display the incoming XML messages in a user-friendly way. The last kind of agents, the movable ones, is the one that needs strategic design. Basically, everything starts from a motor, which has three states: rotate clockwise, rotate counter-clockwise, and idle. To digitally control a motor, it is to be connected to a circuit that has a power supply and a switch that is digitally controllable. Such switches are called relays. Relays can be turned on/off by receiving low voltage signals, which can be imported from a computer's port. To allow two motions however, two relays are needed, as shown in the below figure.

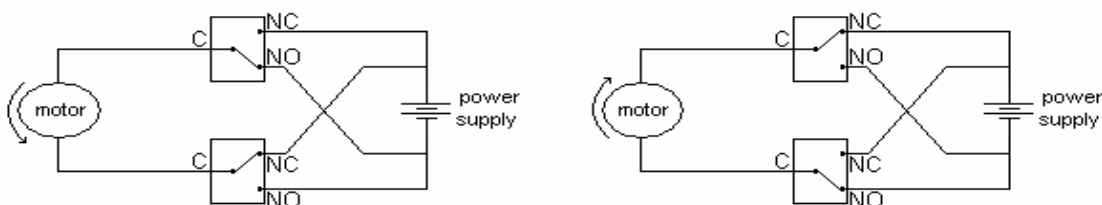


Figure 2 Circuit design for one motor

Note: for relays, NC and NO designate Normally-Closed and Normally-Open respectively. The signal control for this circuit is:

Relay 1 signal	Relay 2 signal	State of motor
0	0	Idle
0	1	Counter-clockwise rotation
1	0	Clockwise rotation
1	1	Idle

To sum up, to make a motor rotate in a certain direction, or stop rotating, we send the target signal combination from a computer to the relay through a wire. The relays switch on/off, allowing the circuit to open/close and leading the motor to rotate/idle. Furthermore, to make a motor agent, it suffices to write an interpreter that receives XML messages and converts them to the right signal combination. It is also desirable to give more power to the user by allowing him/her to program the motor for a set of motions. That is done by integrating a programming language (PL) in the GUI agent whose interpreter converts the English syntax to an XML message. For example, assume our programming language accepts the following syntax:

```
Motor1 clockwise 2 seconds;           //motion (1)
Motor2 clockwise 2 seconds;           //motion (2)
Motor1 counterclockwise 2 seconds;    //motion (3)
Motor2 counterclockwise 2 seconds;    //motion (4)
```

```
<Scenario>
  <Command Motor="Motor1" Direction="clockwise" Size="2"
Unit="second"/>
  <Command Motor="Motor2" Direction="clockwise" Size="2"
Unit="second"/>
  <Command Motor="Motor1" Direction="counterclockwise" Size="2"
Unit="second"/>
  <Command Motor="Motor2" Direction="counterclockwise" Size="2"
Unit="second"/>
</Scenario>
```

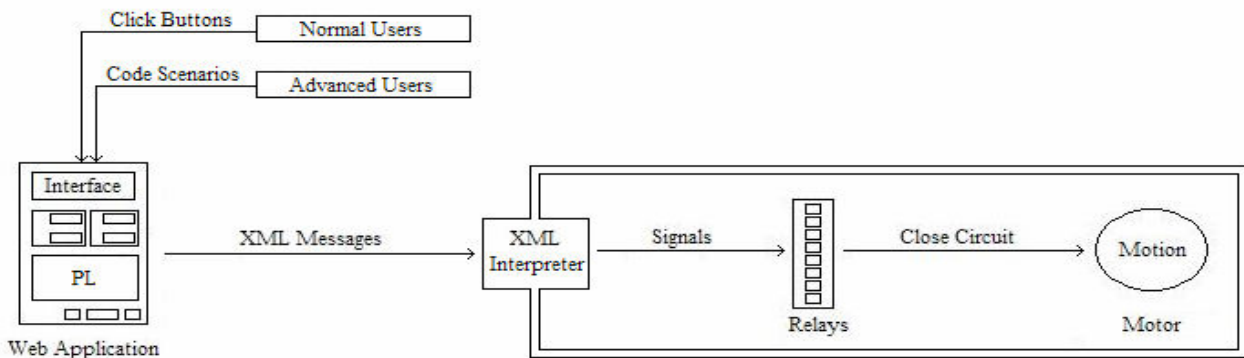


Figure 3 The structure of the laboratory system

Then, the GUI agent would generate the above XML message. This message would then be sent to the motor agents, or possibly to a larger agent such as a robot, and the user would see the

desired set of motions that s/he programmed. After that the GUI agent is integrated in a website, the laboratory system would look like the structure in figure 3.

Integration of RFID

The next important thing is to integrate the RFID technology in the laboratory system. That is achieved by attaching the RFID readers' antennas to motors, allowing rotational and translational degrees of freedom. Furthermore, since a basic robot is just a set of motors and segments that are moved by these motors, then our motor agents can be governed by a (dominant) robot agent, which is controlled just like any one EiA unit/agent. In the mean time, the RFID tags can be stuck on other movable agents/objects such as car and train toys, or simply carried by items on a conveyor belt. This ability of changing the RFID laboratory setup yields a very flexible testing environment for experiments. The agent's interaction in the lab system is illustrated in figure 4.

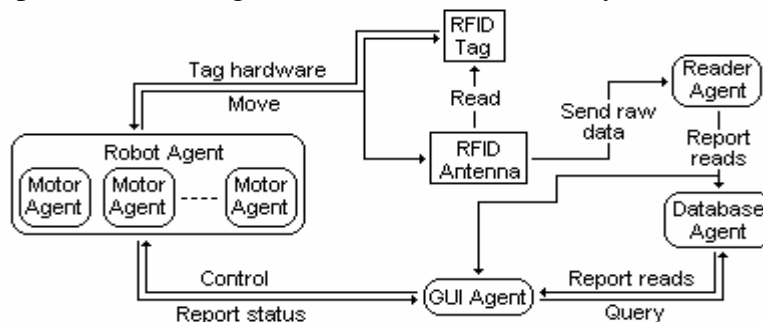


Figure 4 An RFID testing environment built on an EiA framework

Another important component is the presence of a good quality web-camera that prompts the user to view and zoom on a specific part of the laboratory (see figure 5). Such device helps the RFID learner in many ways such as obtain a visual of the laboratory and the RFID equipment, control the hardware with ease, and take snapshots of the lab view for future documentation.

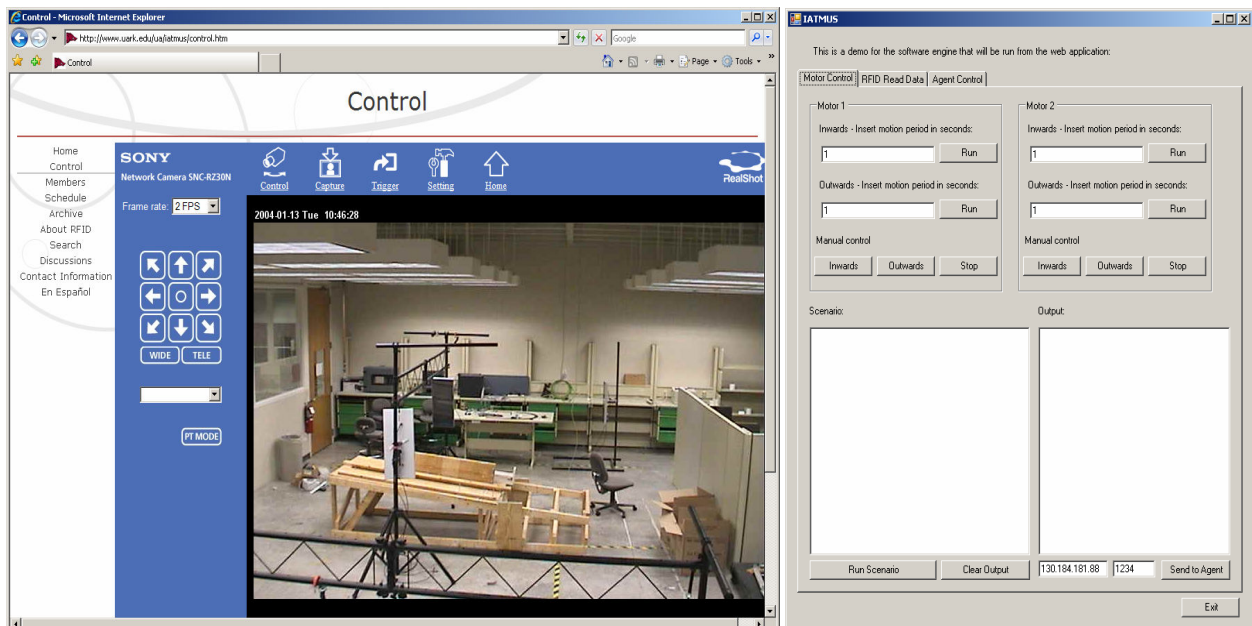


Figure 5 Lab view accessed through a website, and a connectable agent-control GUI

Imagine that the RFID material is viewable and controllable through the web, people from any part of the world can access the laboratory facility, perform experiments, obtain and analyze data, and deduce conclusions. This not only helps gain hands-on experience in the technology, but it also permits to evaluate the equipment's performance and find its flaws. Furthermore, since a website is used to host the GUI agent, the site can include other pages that broaden the navigator's knowledge about different aspects of the used technologies. For example, specialized articles about RFID and its components can be posted, tutorials of the previously mentioned programming language can be included, detailed technical discussions can be shared, and classes about applications of the target technology in the real world can be taught through creative educational ways. The result of such an environment is a powerful highly accessible learning tool that provides the user with the theoretical knowledge, the practical hands-on experience, and solutions to technical problems.

Note: This project is still under implementation and needs some time before completion, especially that a significant part of it depends on developing a reliable hardware system without exceeding the budget's limit. Moreover, efforts from another side of the team are focused on adjoining the RFID learning environment to classes and programs, as well as on designing steps on how to introduce and familiarize students with the new system. Assessments on students' learning in this environment are expected to be revealed next year.

Summary

Everything is Alive (EiA) is an agent-based architecture in which software applications and hardware items can be interconnected by computational processes that allow the translation, interpretation and generation of standard messages among them. These processes therefore allow the translation from a standard language to the native language of a device and vice-versa. The interconnected objects turn into agents that can interact with each other without human interference; they can hence be called "alive!" As a result, the communication among system devices becomes standardized and the integration of new ones will not require heavy or expensive renovation. Such strategy is implemented in a laboratory environment with RFID materials to test its robustness. The systems include agents such as RFID reader agents, database agents, a GUI agent, and other motorized agents. The GUI lets the user change RFID experiments' setups, control different types of RFID readers, gather the read data, and perform computational processes. The website that is used to host the GUI agent can be extended to include articles, tutorials, discussion boards, and teaching modules about the target technology. At the end, the constructed system provides an online laboratory learning environment for technologies in general, and RFID in particular.

References

1. C. Thompson, "Everything is Alive," *Architectural Perspectives Column, IEEE Internet Computing*, Jan/Feb 2004.
2. "Internet of Things," ITU, http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summary.pdf