Distributed Computing Architecture for Agent-in-the-loop, Human-in-the-loop, Hardware-in-the-loop, Distributed Simulation

Xiaobing Zhao (xiaobing@email.arizona.edu), Ameya Shendarkar (ameya.shendarkar@gmail.com), and Young-Jun Son (son@sie.arizona.edu)

1. Distributed Computing for Agent-in-the-loop Simulation

To achieve the second objective, we have developed a distributed computing platform based on DOD High Level Architecture (HLA) (now IEEE 1516 Standard) to facilitate integration of an agent (JACK software, Java application in charge of planner), real human, and the environment (real or simulated (Arena software)).

Task 1: Figure 1 illustrates the relationships between the components of the agent and real-human in-the-loop system. The agent can be implemented in different languages (e.g., JACK, AgentSpeak, Jadex, etc). The agent can interact with either a real environment (automated shop floor control system in our case) or a simulated environment. Also, the simulator (environment) can be implemented in different languages (e.g., Arena™, AutoMod™, ProModel™, etc). The direct interaction of the application software (both agent software as well as simulation software) with the RTI (executable component of HLA) is quite complex and cumbersome. Therefore, we have developed a Distributed Simulation (DS) Adapter, based on COM/DCOM technology, to provide mechanisms for distributed application similar to those provided by the HLA RTI, but with a level of complexity that is manageable by the development resources available in the non-computer science communities. The DS Adapter provides a simplified time management interface, automatic storage for local object instances, management of lists of remote object instances of interest, management and logging for interactions of interest, and simplified object and interaction filtering.

Figure 1: Integration of agent, human, and environment (real or simulated) based on HLA and DS Adapter

Contribution to the distributed simulation field: The Distributed Simulation (DS) Adapter (based on COM/DCOM technology) that we have developed in this work provides...
mechanisms for distributed application similar to those provided by the HLA RTI, but with a level of complexity that is manageable by the development resources available in the non-computer science communities.

The proposed human decision-making model is targeted for a complex system, where a human must adaptively adjust his/her behavior to the dynamically changing environment. Although our work has been developed and demonstrated in the context of the error detection and recovery personnel in a complex automated manufacturing environment, it is expected that the model is directly applicable to the human operators dealing with complex systems in Air Force (e.g. pilots) and in civilian systems such as operators in a nuclear reactor, power plant, and extended manufacturing enterprise.

2. Distributed Supply Chain Simulation Integrating Multiple Discrete Event Simulations

Figure 2 illustrates the relationships between the components of the distributed manufacturing simulation execution environment. All the simulation models have been created using Arena™ 9.0. “Supplier A” model is also built using ProModel™ 4.0. The assumptions involved, the modeling logic used in building the models, the running of the simulation in its entirety are on similar lines as developed by Son and Venkateswaran (2004). Refer to the following web sites to see the prototype systems:

![Supply chain simulation integration](Image)

Figure 2: Supply chain simulation integration (Integration of multiple discrete event simulation models) based on HLA RTI and DS Adapter

3. VMI-based Distributed Supply Chain Simulation Integrating Multiple Discrete Event Simulations and System Dynamics Simulations

In this work, we used the Run-Time Infrastructure (RTI) from the High-Level Architecture (HLA) (Kuhl et al. 1999) and the Distributed Simulation (DS) Adapter to
integrate the various simulators and optimizers for a VMI (Vendor Managed Inventory) supply chain (see Figure 3). The SD models are implemented in Powersim® Constructor 2.51. The DES models are built using Arena® 8.0. The nonlinear optimization problem is solved using AMPL® and solver MINOS™ 5.5. The simulation models explicitly manage time through the exchange of two control messages: *SendMessage* and *ReceiveMessage* (see Figure 3). Upon receiving a message, the simulation advances time, completes internal processing, sends a response, and waits for the next message. The SD models advance time by integrating to the next time step; the DES models advance time by simulating a time period of 1 day. Refer to the following web sites to see the prototype systems:

- The following web site contains screen-captured movies (short version, long version) of the demonstration:
  - Short version: [http://tucson.sie.arizona.edu/CIM/Videos/simdemo_short.avi](http://tucson.sie.arizona.edu/CIM/Videos/simdemo_short.avi)
  - Long version: [http://tucson.sie.arizona.edu/CIM/Videos/simdemo.avi](http://tucson.sie.arizona.edu/CIM/Videos/simdemo.avi)

![Diagram of VMI supply chain simulation integration](http://tucson.sie.arizona.edu/CIM/Videos/simdemo_short.avi)

**Figure 3:** VMI supply chain simulation integration (Integration of hybrid simulation and optimization models) based on HLA RTI and DS Adapter