# A Multi-Agent System for Building Large-Scale Distributed, Hierarchical Control Systems

Vardan Gyurjyan, C Timmer, D Abbott, G Heyes, E Jastrzembski, B. Moffit, E Wolin
Thomas Jefferson National Accelerator Facility, Newport News, VA 23606 gurjyan@jlab.org

## Introduction

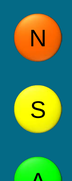
Multi-Agent Framework for Experiment Control Systems (AFECS) is a pure Java based software framework for designing and implementing distributed control systems. AFECS creates a control system environment as a collection of software agents behaving as finite state machines. These agents can represent real entities, such as hardware devices, software tasks, or control subsystems. AFECS agents can be distributed over a variety of platforms. Agents communicate with their associated physical components using range of communication protocols, including tcl-DP, cMsg (publish-subscribe communication system developed at Jefferson Lab), SNMP (simple network management protocol), EPICS channel access protocol and JDBC. A special control oriented ontology language (COOL), based on RDFS is provided for control system description as well as for agent communication.

## Design Architecture

- Collaborating autonomous agents.
- Each agent represents a hardware or software component.
- Messages between agents invoke actions.
- Agents behave as finite state machines.
- Agents communicate with physical components through standard protocols.

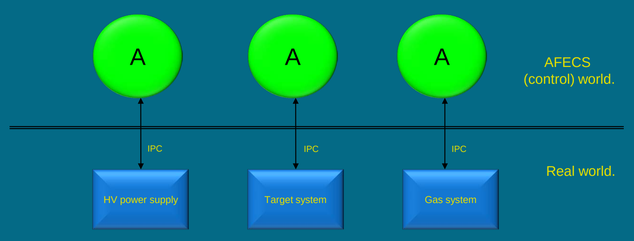
## Agent Categories

- Normative agents
  - Platform administrator
  - Container administrator
  - Registration system
- Supervisor agents
- Component agents



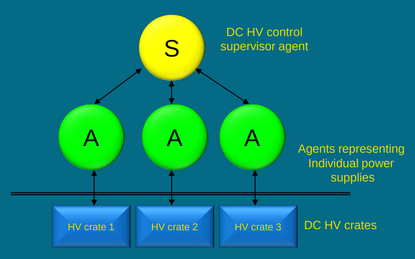

## Component Agent

Agents mimic the state of the real (physical) component, they can invoke actions which change the physical component state.



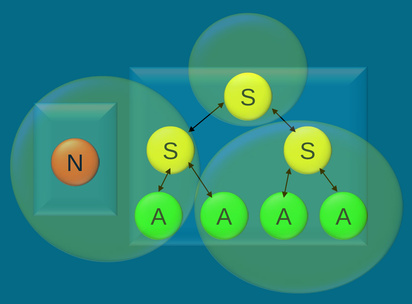

## Supervisor Agent

Supervisor agents have discrete states and they respond to messages from other agents (supervisor or component).

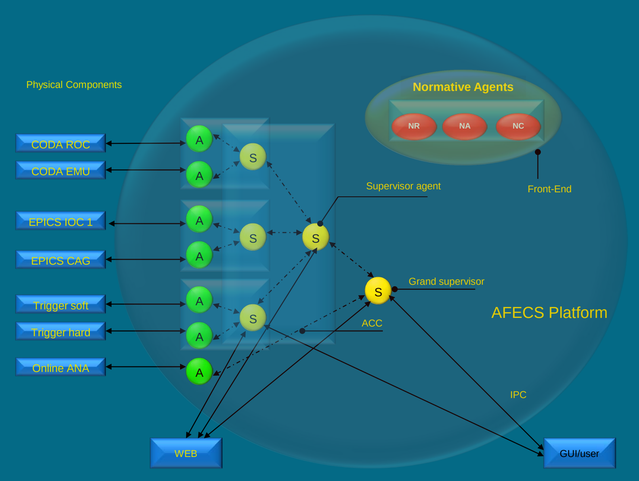



## Platform distribution

- Agents are physically distributed at the configuration time and at run-time




## Hierarchy of Domains




## Communication Protocols

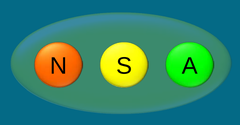
Agents communicate with their associated physical components using range of communication protocols including:

- Tcl-DP (legacy protocol)
- cMsg (publish-subscribe messaging protocol, JLAB)
- EPICS channel access protocol
- SNMP (simple network management protocol)
- JDBC
- OS shell interface

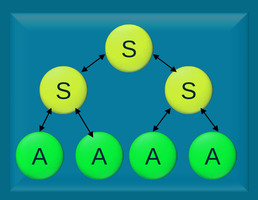
Agents communicate with each other using cMsg pub/sub communication protocol.

## Agents Groupings

- **Physical** (AFECS container)
  A java virtual machine (JVM) is an agent container.



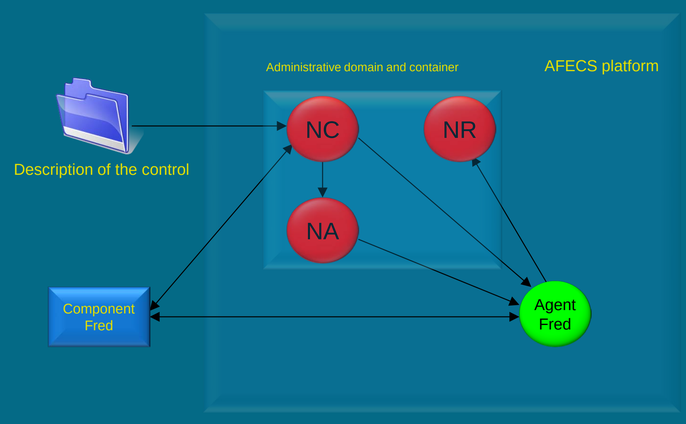

- **Logical** (AFECS domain)
  Agents are grouped into virtual clusters or domains according to their functions. Agents in each domain may be visible to other domains.




## AFECS Domain

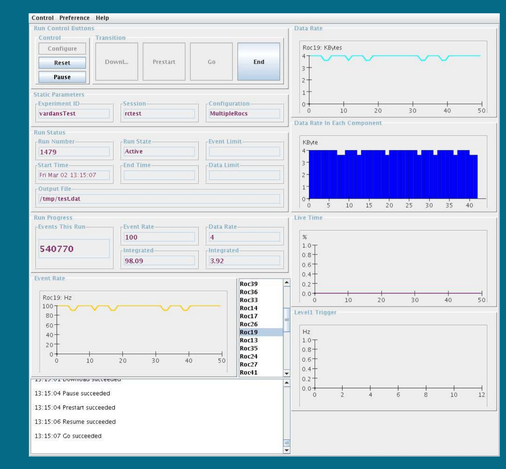
- Reduces complexity of large control systems.
- Encourages modularity and encapsulation.
  (only a few agents and their behaviors are visible outside of the domain limits)
- Promotes control system hierarchical structure.
- Can be redesigned and tested independently, without affecting the rest of the control system.
  (as long as the behavior of the visible agents remains the same)

## Physical Component Integration




## Implementation

The new run control system for the JLAB data acquisition system (CODA) has been developed using AFECS. This run control system is designed to configure, control, and monitor Jefferson Lab experiments. It controls data-taking activities by coordinating the operation of DAQ components (readout controller, event builder, event recorder, event transfer, etc.). The graphical user interface (GUI) for run control is intended to view the status of the data acquisition system and its components and to allow the user to control its operation.




## Conclusions

- Java based framework for designing and implementing hierarchical, distributed control systems with intelligent agents.
- Encourages abstraction, encapsulation, and modularity.
- Has been successfully used to develop run control system for the JLAB data acquisition system, and CLAS experiment web based monitoring system.

## References

Heyes G, et al 1994 The cebaf on-line data acquisition system *Proceedings of the CHEP 1994 Conference*
V. Gyurjyan, et al. "Jefferson Lab Data Acquisition Run Control System", *Proceedings of the CHEP conference. CERN-2005-002, Volume 1, page 151.*
E. Wolin, et al. "cMsg – Publish/Subscribe Package for Real-Time and Online Control Systems", *Proceedings of the 14th IEEE-NPSS Real Time Conference, Stockholm, Sweden 2005.*