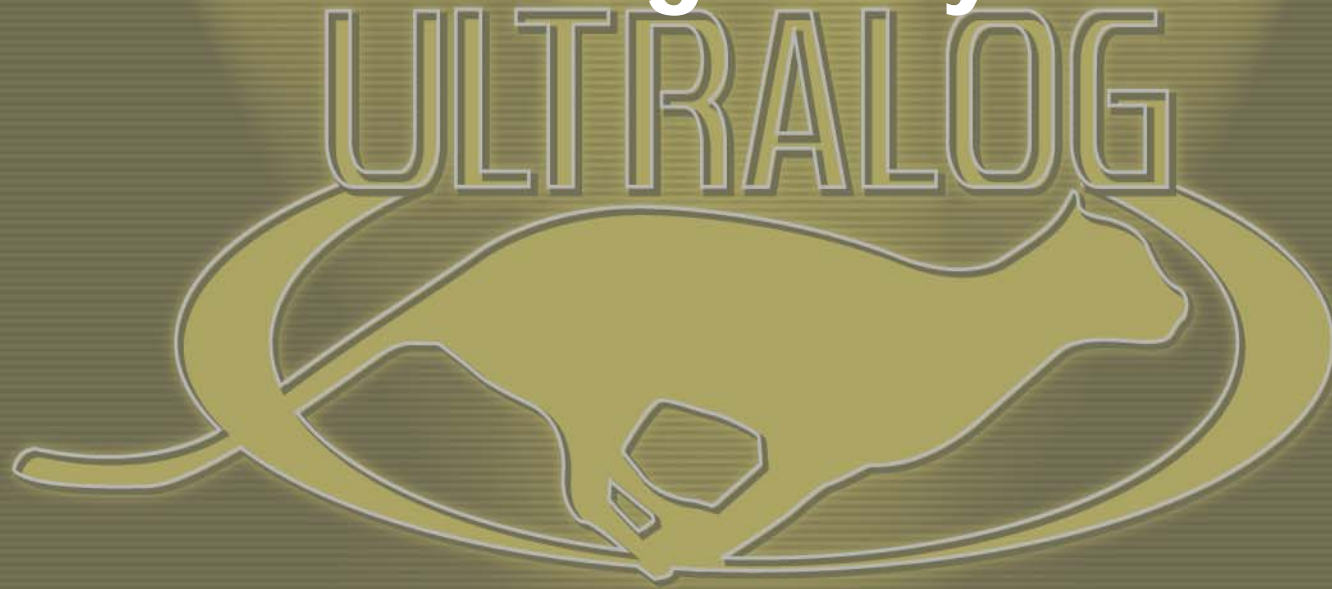


Adaptive Defense Coordination for Multi-Agent Systems



David Wells – OBJS

Paul Pazandak – OBJS

Marian Nodine – Telcordia Technologies

Anthony Cassandra – St Edwards University

- **Large MAS need a variety of autonomous & coordinated Defenses**
 - ◆ Cougaar/UltraLog is an example of such an infrastructure & application
 - emphasis on survivability in the face of kinetic & info warfare attacks
 - 1200 agent logistics planning society
 - ◆ This will also be true of *Application Communities*
- **Problem:**
 - ◆ Defenses generally are myopic, so make mistakes in diagnosis
 - ◆ Even if correct diagnosis, lack “big picture” of what matters most right now
- **Leads to:**
 - ◆ Responses that do not work – simply wrong
 - ◆ Wasteful responses – correct, but not helpful
- **Adaptive Coordinator**
 - ◆ Cope with uncertainty about present & future
 - ◆ Prevent undesirable combinations of actions
 - ◆ Select actions based on high-level criteria of what matters to the application

Kinds of Defenses



- **Corrective – fix a problem**
 - ◆ e.g., restart a Dead agent elsewhere
- **Compensatory – work around a problem**
 - ◆ e.g., try alternate communications to avoid an unfixable network problem
 - ◆ e.g., encrypt to counter an eavesdropper who can't be suppressed
- **Protective – keep a problem from occurring**
 - ◆ e.g., persist agent state more often to avoid loss of work
- **Application Tradeoffs – change the way work is done**
 - ◆ e.g., produce lower fidelity logistics plans to go faster

- **Some are “done & gone”, while others have continuing costs**

Problems Addressed



- Defense independence has advantages:
 - ◆ Ease of development
 - ◆ Reaction speed
 - ◆ Simplicity
 - ◆ Ability to work during partitions
- But:
 - ◆ Local knowledge misdiagnoses problems
 - So we do the wrong thing
 - ◆ Local knowledge insufficient to see a problem
 - So we fail to do anything
 - ◆ Two or more components decide to act
 - Overwork, both trying to solve the problem
 - Trying to solve two conflicting diagnoses of the problem
 - ◆ Local components unable to weigh importance of other survivability properties over their own priorities
- **Global knowledge needed and global priorities must be considered w/o creating a bottleneck or single point of failure**

Adaptive Control Goals



- Prevent undesirable combinations of actions
- Intelligent use of information when selecting responses
 - ◆ Current status
 - ◆ Expectations of future behavior
- Cope with uncertainty when selecting responses
 - ◆ Conflicting diagnoses
 - ◆ Missing diagnoses
 - ◆ Imprecise projections about future stress profiles
- Selection of responses is sensitive to high-level policy
- Adaptive
 - ◆ To changes in High-Level policy (changes to MAU curves)
 - ◆ To changes in perceived stress environment

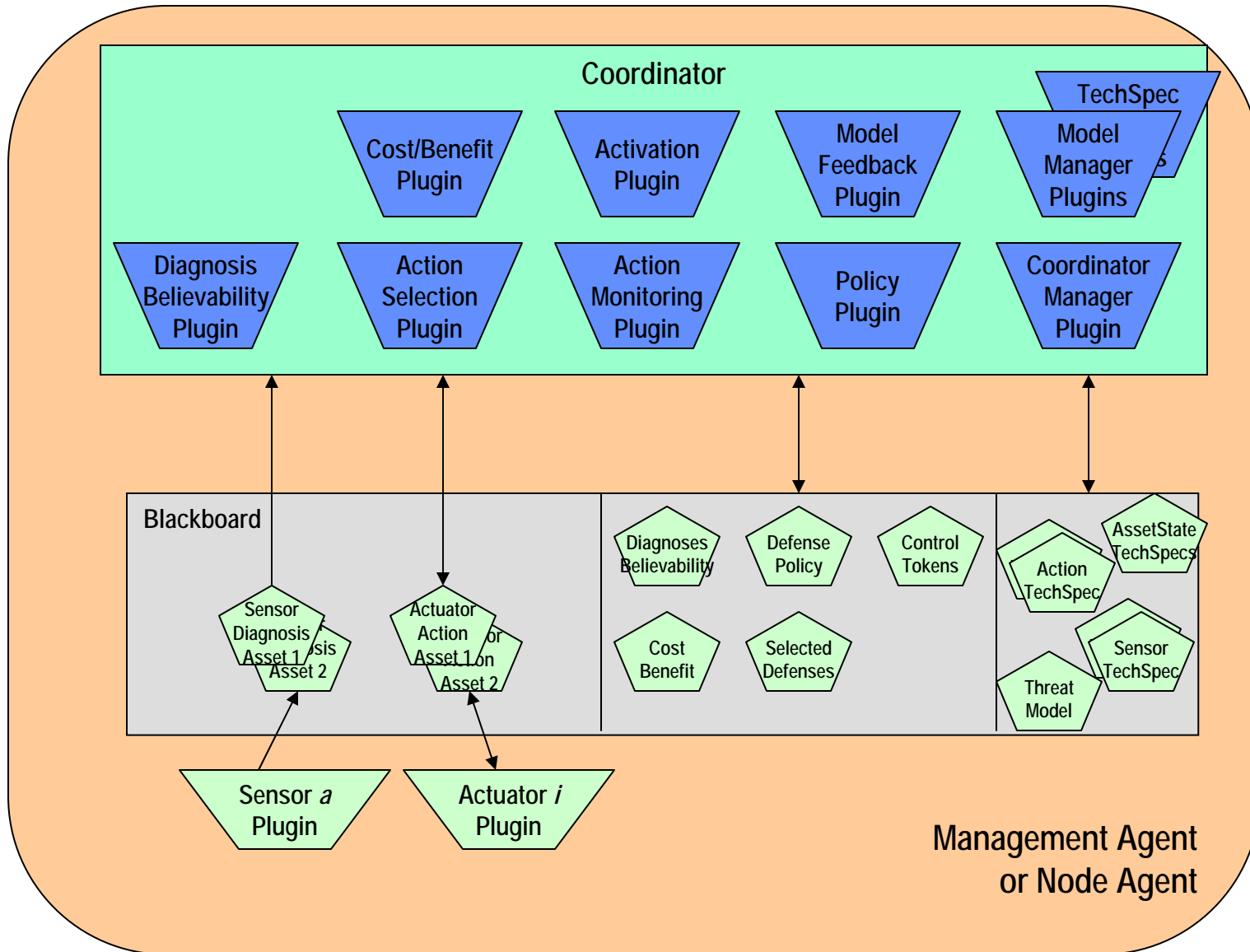
- Measured by improved Survivability as seen by the application
 - ◆ W.r.t. uncoordinated baseline & to an ad hoc pairwise control baseline

Decision Logic Overview



- **Estimate society state**
 - ◆ In the face of uncertainty
 - Incorrect diagnoses
 - Conflicting diagnoses
 - Missing diagnoses
 - ◆ Considering expected stresses in the environment
- **Policy driven action selection**
 - ◆ Base benefits of society assets being in a particular state
 - ◆ Policy changeable valuation of different system properties
 - ◆ Choice of objective functions when computing action benefits
- **Feedback loop to determine if an enabled action had the desired result**
- **Highly “plugIn-ized” Coordinator internal architecture**
 - ◆ Each basic capability can be accomplished in many different ways at varying degrees of complexity, accuracy, development effort, etc.
 - ◆ Heavy internal decomposition allows putting effort where the payoff is

Coordinator Internal Architecture





- **Freely available**
 - ◆ Cougaar Open Source
- **Portable**
 - ◆ Architecture is independent of
 - Specific sensors & defenses
 - Underlying agent framework
- **Model Fidelity**
 - ◆ How accurate can the models be?
 - ◆ How accurate do the models need to be? (sensitivity experiments required)
 - ◆ What is the correct level of abstraction for modeling?
- ***Application Communities* will require a capability like this**