Adjustably Autonomous Control Systems

David Kortenkamp

NASA Johnson Space Center

ER2

Houston TX 77058

kortenkamp@jsc.nasa.gov

http://www.traclabs.com/~korten

Adjustable Autonomy Systems

A control system that has the ability to:

- be completely in control
- supervise manual control
- be somewhere in between
- shift among these control extremes in a safe and efficient manner

is an *adjustable autonomy* system

Adjustable Autonomy

- A system's adjustable autonomy can involve changes in:
  - The complexity of the commands it executes
  - The resources (including time) consumed by its operation
  - The circumstances under which it will either override or allow manual control
  - The circumstances under which it will request user information or control
  - The number of subsystems that are being controlled autonomously

Motivations

- Complexity of commands
- Uncertainty/changing environment
- Safety/monitoring/politics
- Maintenance or calibration
- Training
- Flexibility
- Resource allocation

Benefits

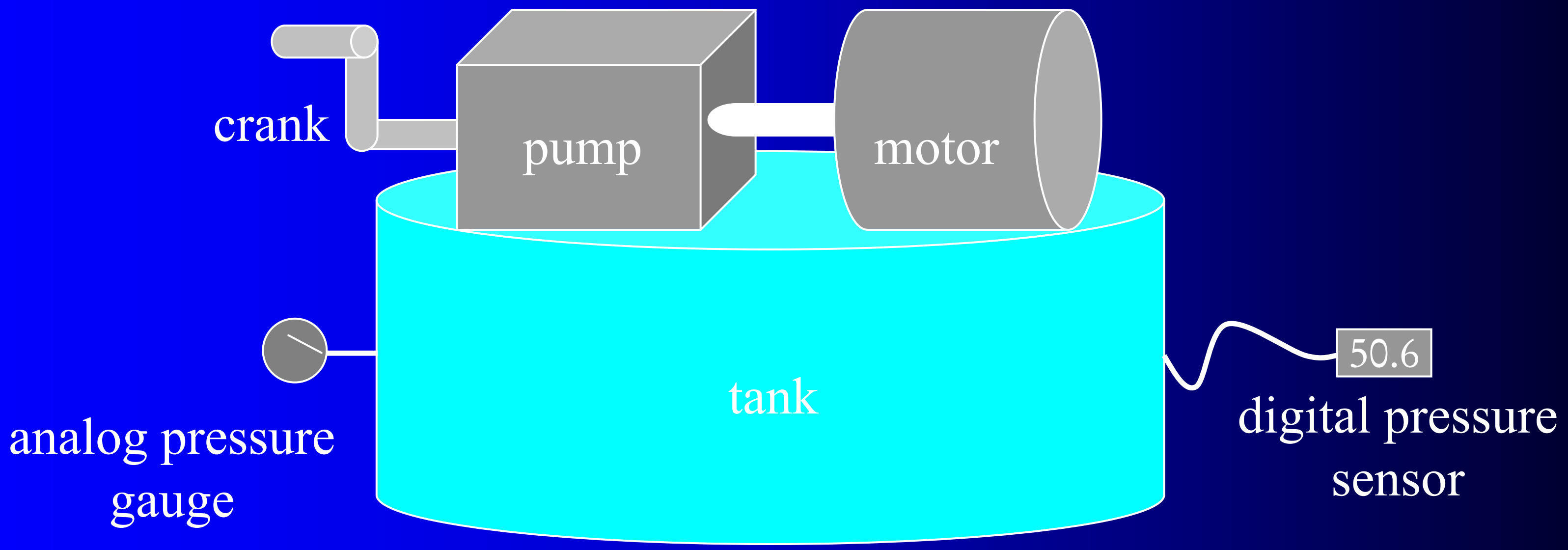
- Partial autonomy where full autonomy not possible or desired
- Lower cost
  - difficult-to-automate parts of the system can be left for humans
- Safety and reliability
  - human experience brought to bear when needed
- Operator acceptance of autonomous systems

Design of AA Systems

- Strive for full autonomy where possible.
- Build in appropriate sensing, even for situations in which the human is in control
- Build in capabilities that enable multiple methods for humans and system to achieve goals
- Plan for changes in autonomy as much as possible
  - system-planned changes in autonomy
  - system-initiated changes in autonomy
  - human-initiated changes in autonomy

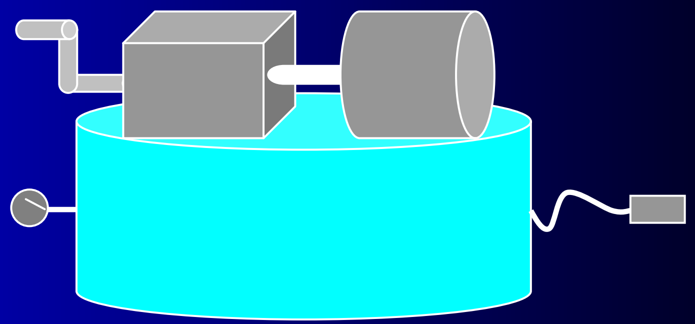
A Simple Example

- Here's a simple, non-robotic example



Our Running Example

- Controller
  - human who decides whether pressure should be increased
  - computer that decides whether pressure should be increased
- Actuator
  - pump that human cranks
  - pump that motor activates

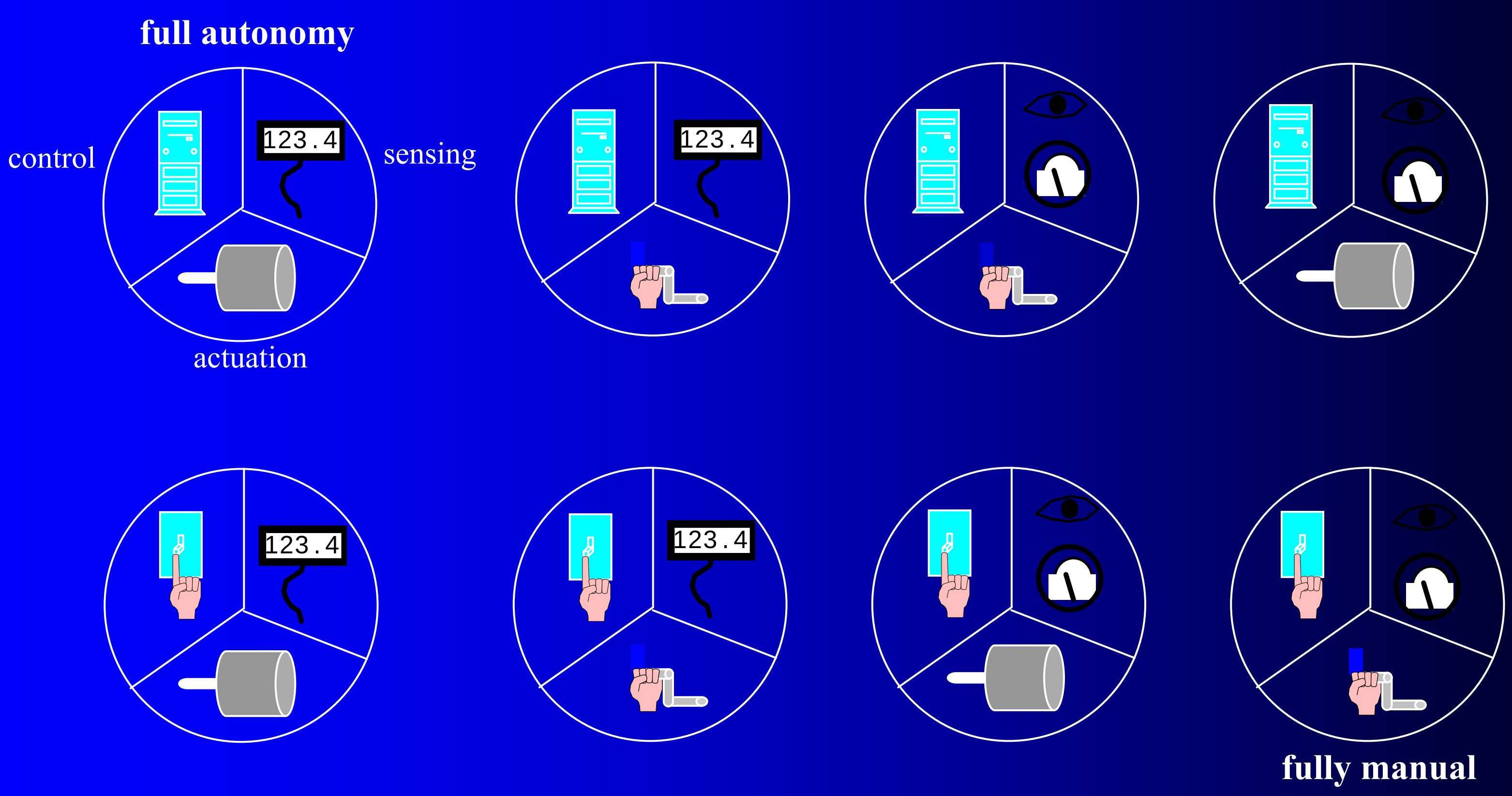


- Sensor
  - digital pressure sensor that computer reads
  - analog pressure gauge that human reads

Adjusting Autonomy Example

- Control system believes motor is broken and is no longer responding to commands
  - Belief is either by internal processes or external command
- Autonomy is switched from autonomous (motor) to crank (human) for pump
- Autonomous control system still decides when pump needs to be activated
  - Human is only an actuator, not a controller

Different Autonomy Modes



New Project

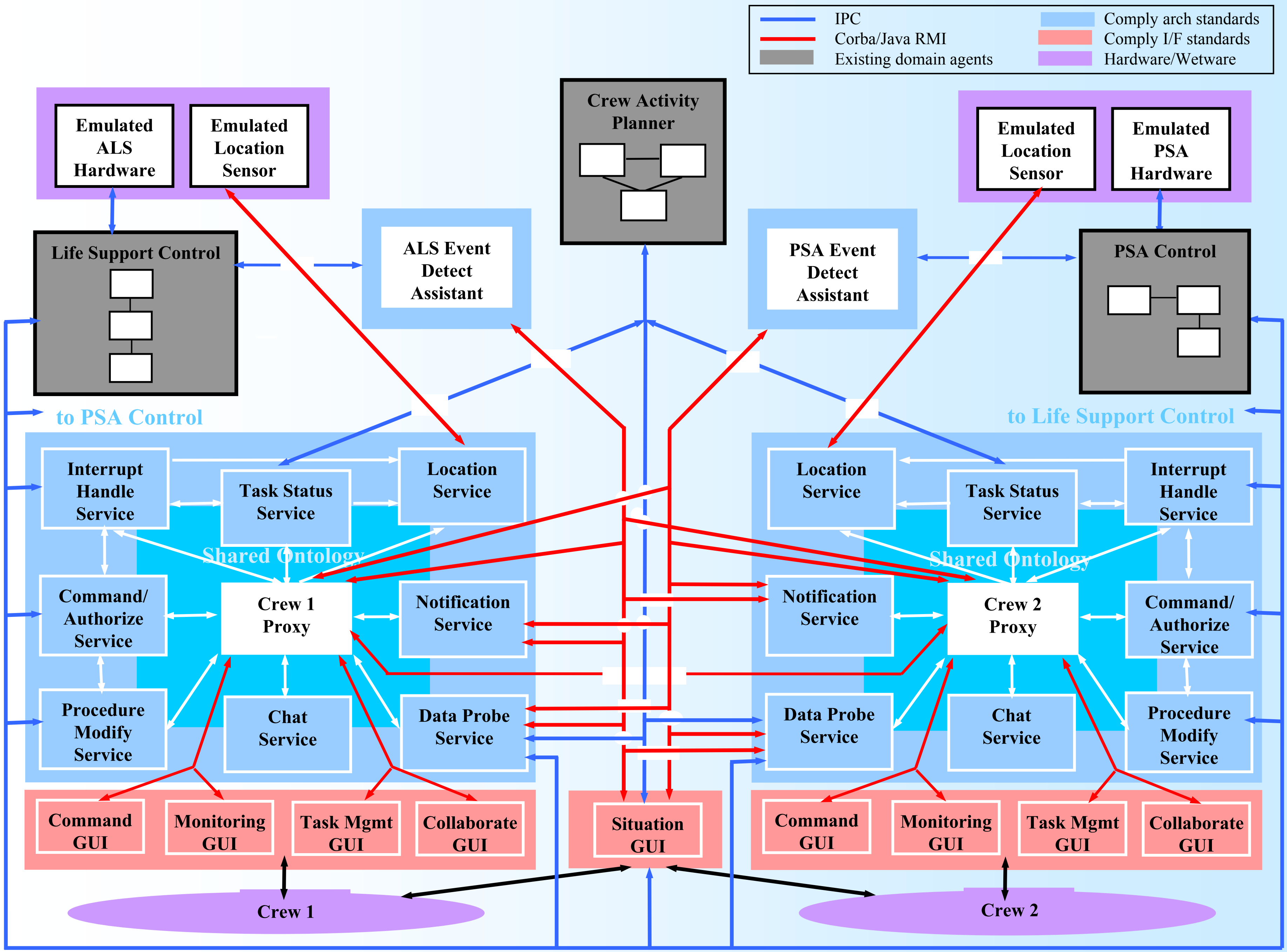
- Architecture for interfacing multiple adjustably autonomous control systems to multiple crew members
- Control systems may be distributed
- Crew may be distributed
- Interface must be intelligent
- NASA-funded basic research
- Title: Distributed Crew Interaction with Advanced Life Support Control Systems

Distributed Crew Interaction Enables Autonomy

- Crew should be able to monitor & control autonomous operations remotely
  - Access situation information quickly and execute control from anywhere at remote site
  - Interleave monitoring & control operations with unrelated tasks
- Crew at disparate locations should be able to collaborate among themselves and with automated systems distributed throughout site
  - Make control decisions jointly
  - Manage conflicting control commands
- Crew override of automated control should be infrequent and on-demand
  - Respond to anomalous control situations outside range of expected behaviors
  - Take advantage of opportunities that require novel control approaches
- Crew should supervise activity planning & control adaptation
  - Build and update activity schedules (human & software)
  - Manage resources critical to life support (e.g., food, water, gases)
  - Change control policies in response to system degradation and crew & mission needs

Key Technical Issues

- **Providing context for collaboration**
  - Support common grounding in task as shared frame of reference
  - Share knowledge about ongoing tasks, goals and intended effects of those tasks, and beliefs about environmental and system states
- **Handling interruptions without error**
  - Assist human in suspending ongoing tasks and resuming them later
  - Update crew plans when tasks are interrupted
- **Notification and reminding**
  - Detect and notify crew of important control events
  - Investigate visualization and data summarization for notification
  - Remind crew of pending or interrupted tasks
- **Synchronous and asynchronous collaboration**
  - Automatically collect & route information based on task needs
  - Provide information and views that characterize situation after the fact
  - Manage the timeliness of collaborative response
  - Interleave collaborative interaction with latency


IPC
Corba/Java RMI
Existing domain agents
Comply arch standards
Comply I/F standards
Hardware/Wetware
Emulated ALS Hardware
Emulated Location Sensor
Crew Activity Planner
Emulated Location Sensor
Emulated PSA Hardware
Life Support Control
ALS Event Detect Assistant
PSA Event Detect Assistant
PSA Control
to PSA Control
to Life Support Control
Interrupt Handle Service
Task Status Service
Location Service
Shared Ontology
Command/ Authorize Service
Crew 1 Proxy
Notification Service
Procedure Modify Service
Chat Service
Data Probe Service
Location Service
Task Status Service
Interrupt Handle Service
Shared Ontology
Notification Service
Crew 2 Proxy
Command/ Authorize Service
Data Probe Service
Chat Service
Procedure Modify Service
Command GUI
Monitoring GUI
Task Mgmt GUI
Collaborate GUI
Situation GUI
Command GUI
Monitoring GUI
Task Mgmt GUI
Collaborate GUI
Crew 1
Crew 2

Scenario: Adjustable Autonomy

Crew 1 is online and Crew 2 is offline when scenario begins

- Event Detection Assistant detects anomaly in Biological Water Processor (BWP) (I.e., high pressure in nitrifier tubes) & passes information to Control System and Crew Proxies
- Crew Proxies get information on crew notification preferences from Notification Service
- Crew1 Proxy notifies Crew 1 directly since online; notifies Crew 2 (who is offline) through database
- Crew 1 reviews situation assessment using Situation Viewer
- Crew 1 requests Crew1 Proxy for command capability of Biological Water Processor
- Crew1 Proxy activates Command Service for Biological Water Processor
- Command Service request Authorization Service to determine if Crew 1 is authorized to command BWP
- Authorization Service determines that Crew 1 is authorized to command the Biological Water Processor and no other crew member is currently commanding
- Command Service activates Command GUI for (1) direct commanding, and (2) procedure modification
- Crew 1 temporarily disables autonomous commanding in the Biological Water Processor using procedure modification but requests to be notified of command recommendations; this permits manual control
- Crew 1 begins to manually slough the nitrifier tube with high pressure by increasing flow through tube; requests to be notified when tube pressure drops to normal level

Scenario: Adjustable Autonomy (cont.)

- Crew 2 comes online and notices high flow rate in tube and is concerned the flow pump is malfunctioning
- Crew 2 requests command capability through Crew2 Proxy to reduce pump flow
- Crew2 Proxy activates Command Service for Biological Water Processor Command Service requests Authorization Service to determine if Crew 2 is authorized to command BWP
- Authorization Service determines that Crew 2 is authorized to command the Biological Water Processor but Crew 1 is currently commanding as "sole commander"
- Command Service denies command access because Crew 1 is commanding; options for handling denial include
  - Request change in command strategy from sole commander to consensus command
  - Request take over from crew 1 (requires agreement of crew 1 or override due to authorization hierarchy)
  - Abandon request
- Crew 2 reviews Biological Water Processor situation and contacts Crew 1 via electronic "chat" for more information
- Crew 2 determines Crew 1 has it under control, and proceeds to other tasks
- Crew 1 returns the Biological Water Processor to normal operating mode via Command GUI

Architecture for Adjustable Autonomy

- Collaboration Architecture
  - One crew proxy per crew member that provides following services
    - Command Service
      - Request Authorization Service to determine if crew is authorized to command BWP
      - Activates Command GUI for (1) direct commanding, and (2) procedure modification
      - Handles authorization problems with user selectable strategies (single commander, concensus commanding, or majority vote; first available vs authority hierarchy)
    - Authorization Service
      - Determines if (1) crew is authorized to command the Biological Water Processor and (2) if other crew member is currently commanding
    - Notification Service
      - Handles differences in notifying crew when online vs offline
      - Notifies crew when operational events occur (like anomalies)
    - Crew Chat Service
      - Initially provide free-form text exchange among crew
      - Evolve to formatted text, forms, etc. that permit monitoring crew activities

Interface for Adjustable Autonomy

- Collaboration Architecture (cont)
  - Crew Proxy Services (cont)
    - Location Service
      - Determines crew location
      - Determines if crew is online or offline
  - Event Detection Assistant
    - Detects significant events (including anomalies) in Biological Water Processor (BWP) & passes information to Notification Service in Crew Proxy
- Collaboration Interface
  - Situation View for situation assessment
  - GUI for Remote Commanding
  - GUI for Procedure Modification
  - Notification GUI that handles different platforms and modalities

Conclusions

- Effective adjustable autonomy minimizes the *necessity* for human interactions but maximizes the *capability* for humans to interact at whatever level is most appropriate for any situation at any time.
- Adjustable autonomy must be designed in from the beginning -- assume pesky humans will always want to be meddling with the autonomous system!
- Often full autonomy is not possible (for technical, political or economic reasons) and adjustable autonomy is the only solution.
- By asking the right questions at design time adjustable autonomy can be safe and practical.

Design Decisions

- What tasks can be done only by humans? Only by automation? By both?
  - are there certain times or situations when a task should only be done by a human or automation?
- Who can set the level of autonomy for a task?
  - can the level of autonomy change at any time or only under certain circumstances?
  - is the level of autonomy fixed at run-time or is it flexible?
- What are the timing issues with respect to a change in autonomy?

Design Decisions cont.

- Arranging the hierarchy
  - can autonomy setting at one node apply to all descendants?
- What are possible autonomy level transitions? What transitions are not permitted?
- Is information necessary to control the system available to the user or to other agents?
  - current state, tasks, goals
- Are there multiple ways to accomplish the same task? Are they selectable by a user? By a planner?

Design Decisions cont.

- What parts of the system are commandable from outside?
  - by humans?
  - by other systems?
  - how are they commanded?
- How is success and failure of other agents recognized?
  - feedback?
  - observation?
  - timeout?