

Distributed Crew Interaction (DCI) with Advanced Life Support Control Systems

Year 1 - Architecture for Collaboration

February 2002

JSC

Debra Schreckenghost
David Kortenkamp
Carroll Thronesbery

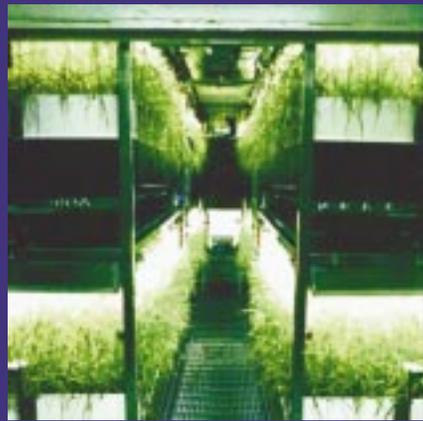
Cheryl Martin
Pete Bonasso
Tod Milam

Ohio State University
David Woods

Project Summary

Autonomous Control of Life Support Systems

Plant Growth



Air Revitalization



Water Recovery



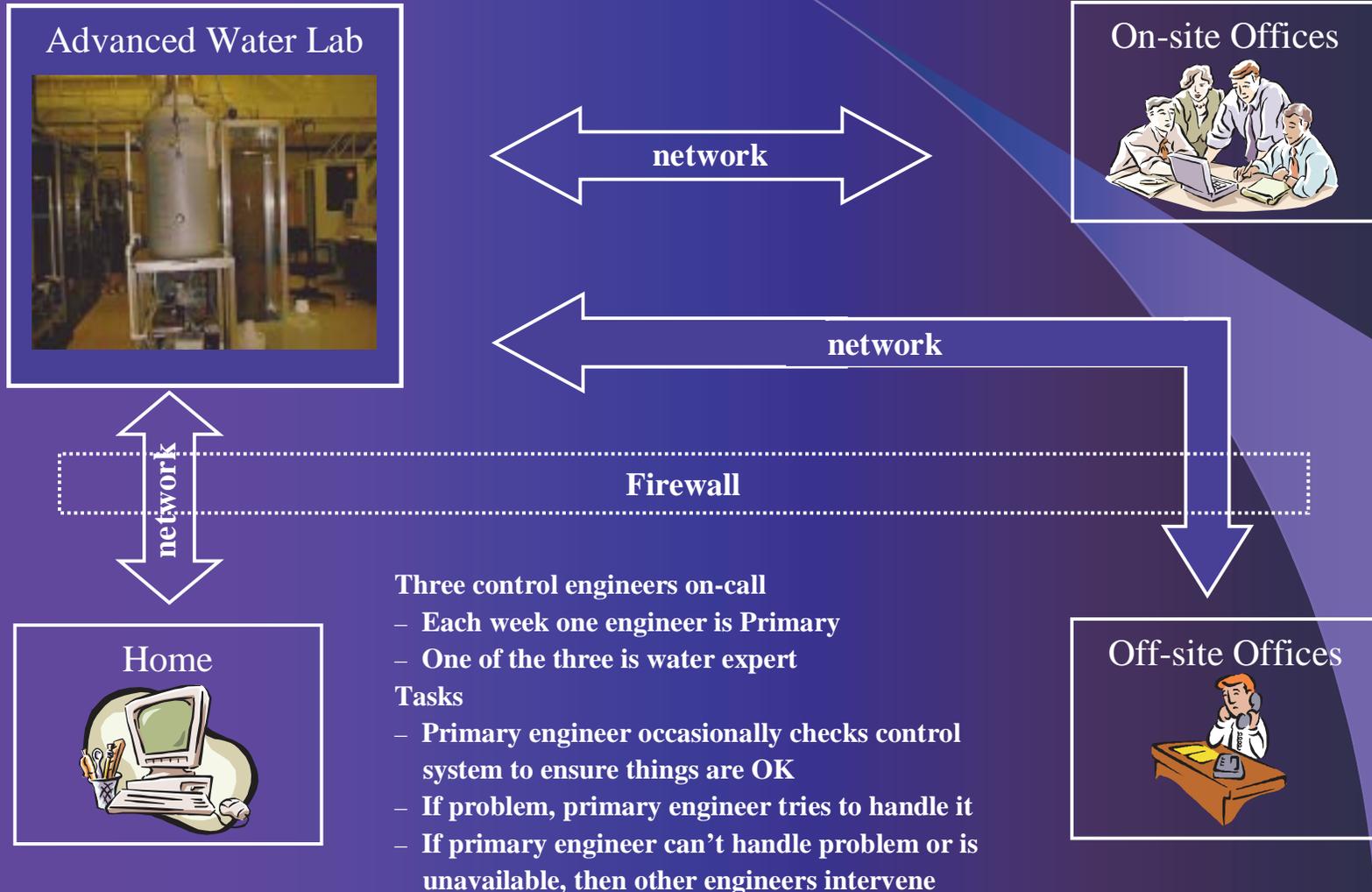
Phase 1 Human
Test

Phase 3 Human
Test

Advanced Water
Lab



Supervising Advanced Water Lab



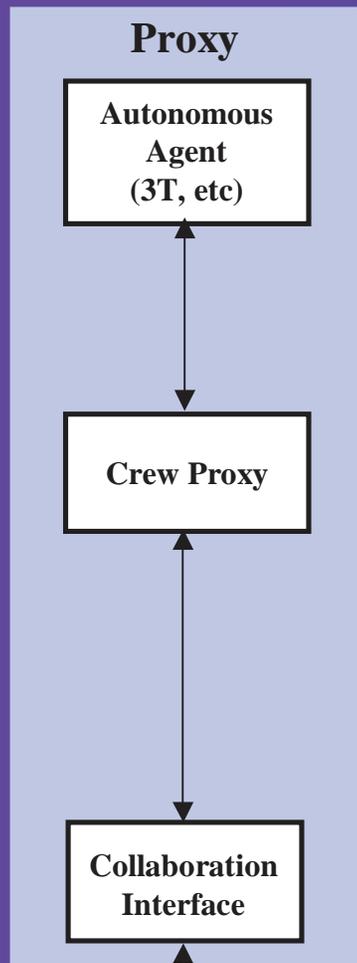
Requirements for Distributed Interaction

- **Visualization** tools that characterize control circumstances
- **Notification** based on role, location, and preference
 - Primary engineer should be the first notified when anomaly occurs and reminded to take action if required
 - Water expert should be notified of all anomalies
 - Support is needed for locating available engineer when primary unavailable
- **Task management** for control engineers
 - Notify/remind engineers of manual tasks
 - Track completion of manual tasks
- **Remote commanding** strategies
 - Primary engineer should be able to start, reconfigure, and stop automated control from his office or home
 - Commands authenticated and conflicts resolved for manual commands
 - Automatically **adjust control autonomy** for manual commands
- Assist engineers in **handling interruptions** to normal activities

Year 1 Collaboration Architecture Study

- **Objective:** design an architecture for remote human collaboration and interaction with distributed, computer-based autonomous control
- **Approach**
 - Identify functional and interaction requirements for collaboration
 - Evaluate candidate technologies for use in our architecture
 - Develop criteria to be used in evaluating candidate technologies
 - Survey technologies and evaluate how well they support collaboration
 - Design the architecture based on the collaboration requirements and the results of technology evaluation
- **Status**
 - Year 1 study is complete
 - Results documented in Year 1 Report
 - Architecture design
 - Survey and evaluation of technology for agent interaction and collaboration

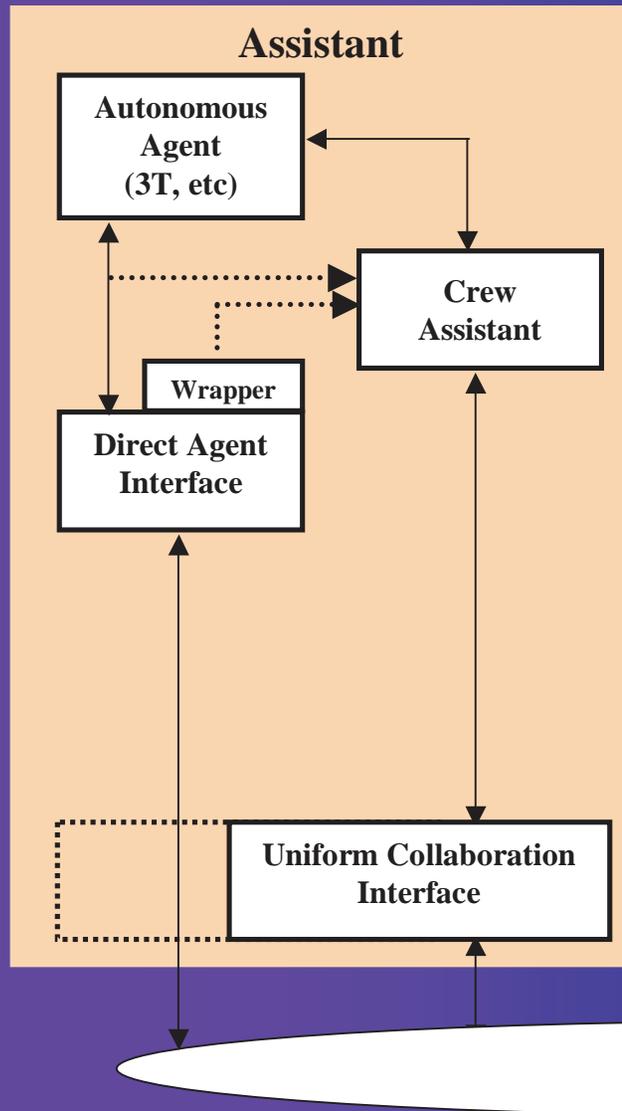
Models of Crew Interaction



Proxy Model

- All interaction with other agents is done through that agent's proxy
- Integration at level of shared models and agent communication
- Design Approach
 - Provides functionality for a single agent by coordinating collaboration services based on logical dependencies among these services
 - Services are defined independent of a specific agent
 - Services are configured for a specific agent during execution
 - Provides uniform access to information about its agent that supports collaboration with other agents
 - Utilizes standard models in the architecture to delineate and represent collaborative information

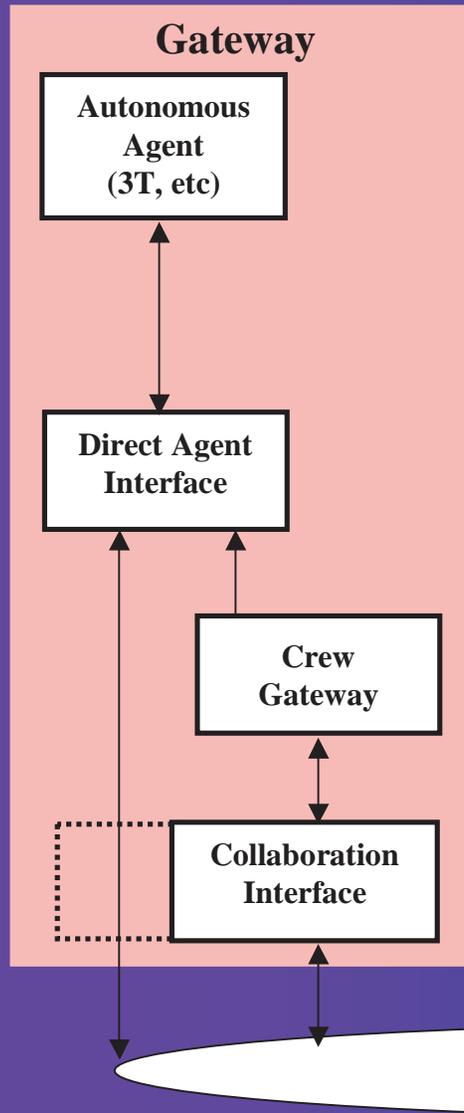
Models of Crew Interaction



Assistant Model

- Agent interacts either directly with other agents or interact through it's assistant
- Direct agent interaction can be monitored by the assistant
- Integration at level of (1) models & communication and (2) presentation level
- Design Approach
 - Supports passive monitoring or mediating of direct agent-agent interaction (i.e., interaction outside our architecture)
 - Supports coordination of collaboration services and utilization of collaboration information much like a proxy

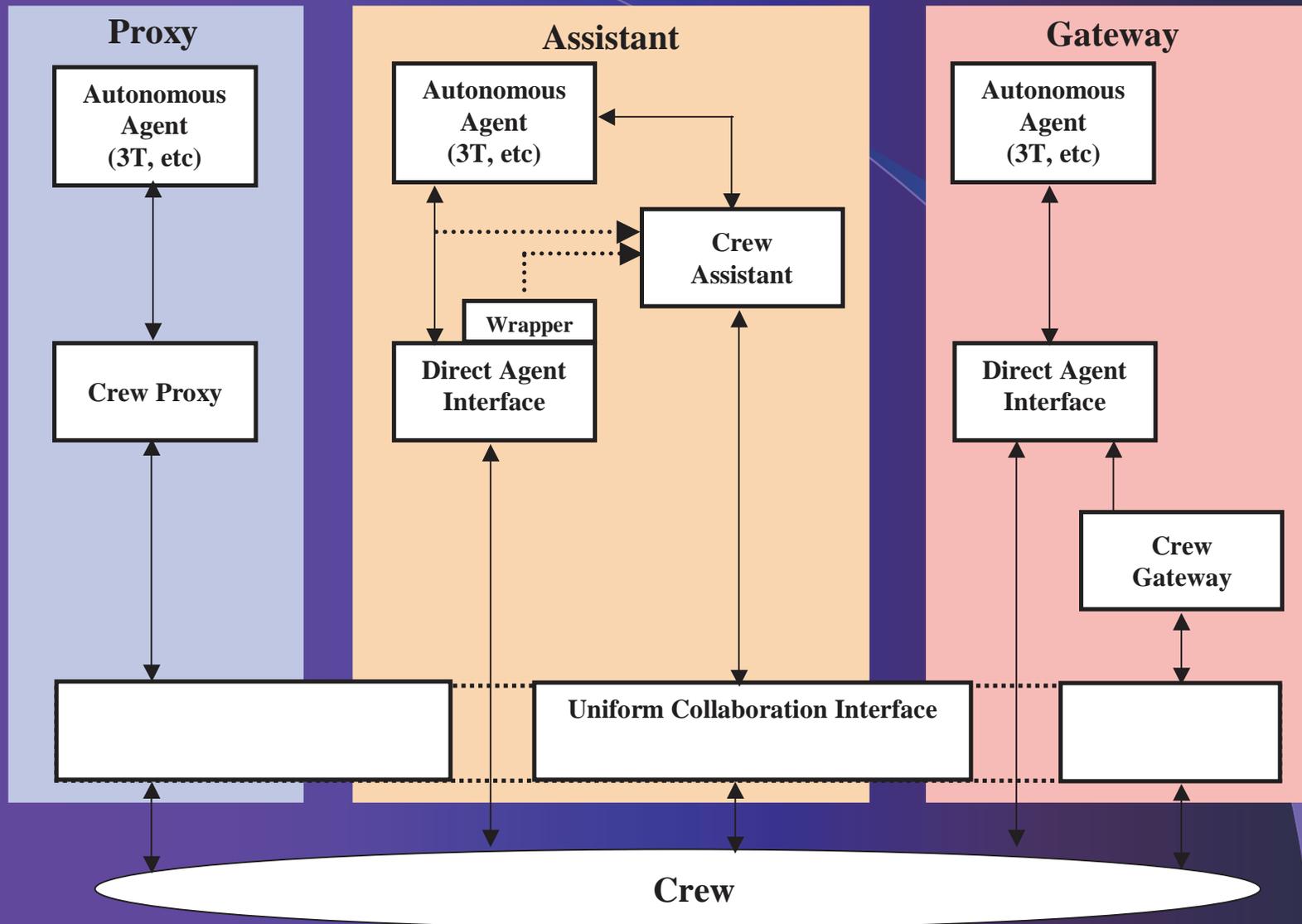
Models of Crew Interaction



Gateway Model

- Agent interacts directly with other agents, and gateway mediates initiation and termination of that interaction
- Gateway does not supports passive monitoring or mediation of task interaction
- Integration at presentation level
- Design Approach: provide “thin” collaboration services when initiating or terminating direct interaction
 - Access control (e.g., user authorization)
 - Interface configuration (e.g., parameter passing when launching applications)
 - Timeout of abandoned processes (e.g., shutdown process if no interaction for some elapsed time)

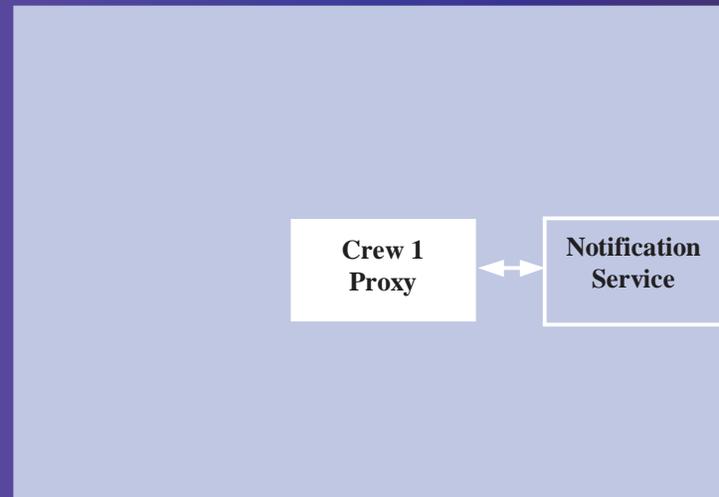
Models of Crew Interaction



Services for Crew Proxy

Notification Service

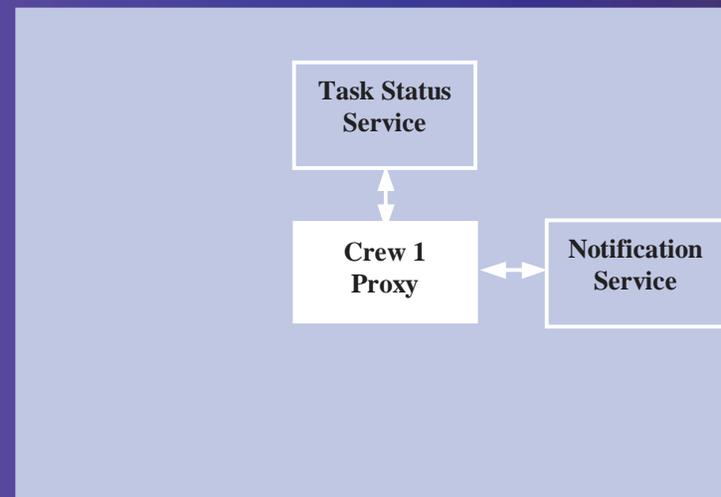
- Determines who to notify of an event and how to notify them
- Combines crew state (e.g., online vs offline), crew role, and crew notification preferences to do this
- Interesting events include environmental changes or control actions



Services for Crew Proxy

Task Status Service

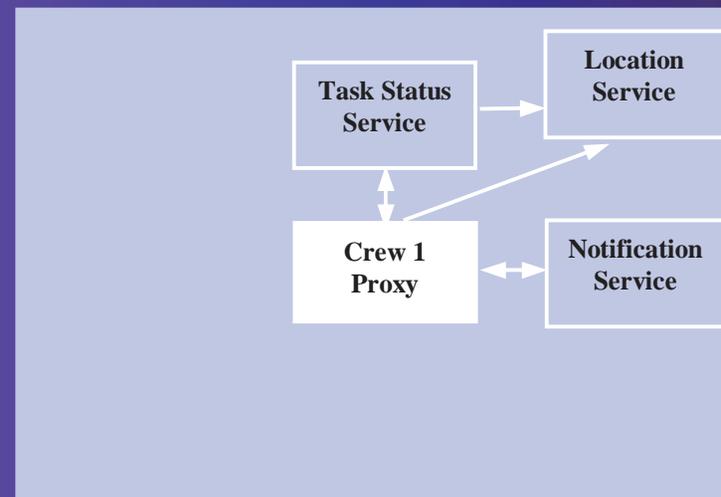
- Provides activity tracking and task management capabilities for both crew and autonomous control agents
 - Assess the completion status of planned crew tasks
 - Notify the crew of scheduled tasks, including pending tasks and deadlines
 - Notify automated control agents of the completion status of crew activities



Services for Crew Proxy

Location Service

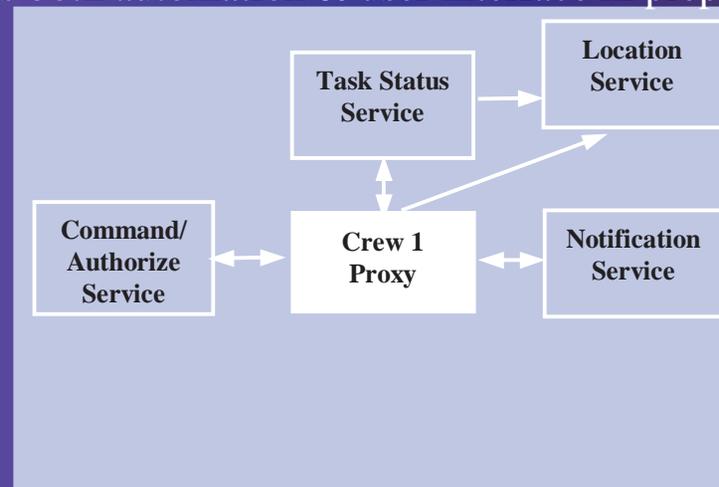
- Determines crew location as physical location, current computing platform, and whether crew is online or offline
- Provides crew location information for use in
 - Tracking the completion status of crew activities
 - Determining how to notify the crew of events
 - Customizing presentation of information



Services for Crew Proxy

Command and Authorization Service

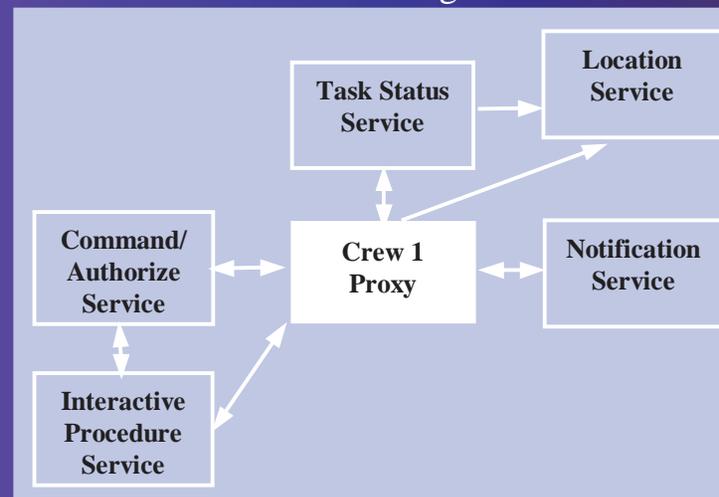
- Assists crew in remotely commanding the life support systems
 - Execution of *standard manual procedures* (e.g., maintenance like filter changeout)
 - *Direct commanding* of unique control sequences
- Implements a concept for adjustable autonomy
 - Determines if the crew is authorized to command (i.e., access control)
 - Resolves authorization conflicts when more than one crew member commands
 - Reconfigures both automation & user interface in preparation for commanding.



Services for Crew Proxy

Interactive Procedure Service

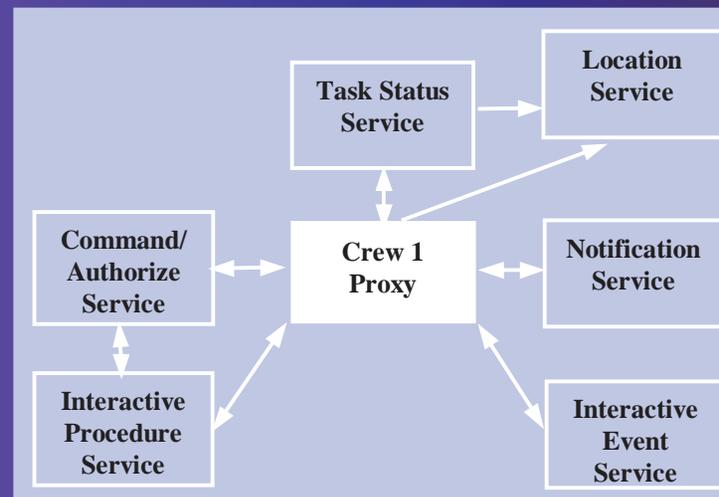
- Assists crew in temporarily modifying standard procedures in automated control
- Implements a shared discourse plan for Crew and Proxy to change procedure
- Guides structured modification of selected procedure
 - Change the steady state operating parameters (e.g., setpoints)
 - Change to an alternative or backup sensors
 - Add probes to export information about the execution of control tasks
 - Temporarily disable automated commanding



Services for Crew Proxy

Interactive Event Service

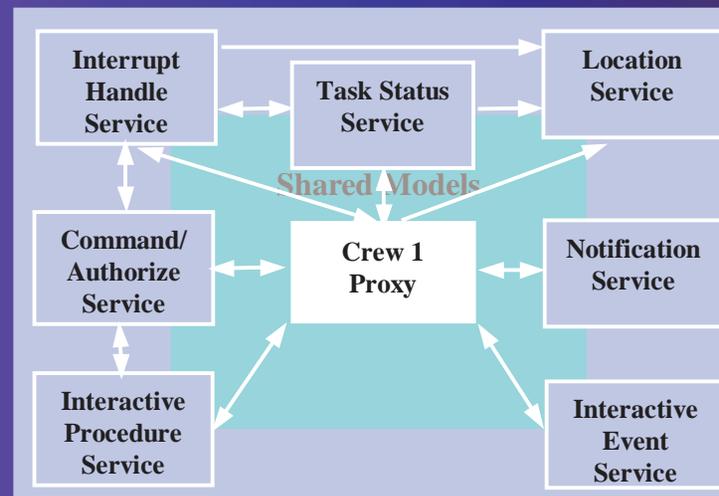
- Assists crew in interactively defining new operational events and controlling automated monitoring for these events
- Defined as data or action *probes* that are temporary event detectors and that generate information specific to the needs of a single crew member
- Provide specialized crew interface to the Event Detection software and Interactive Procedure Service just described



Services for Crew Proxy

Interruption Handling Service (implemented in FY03)

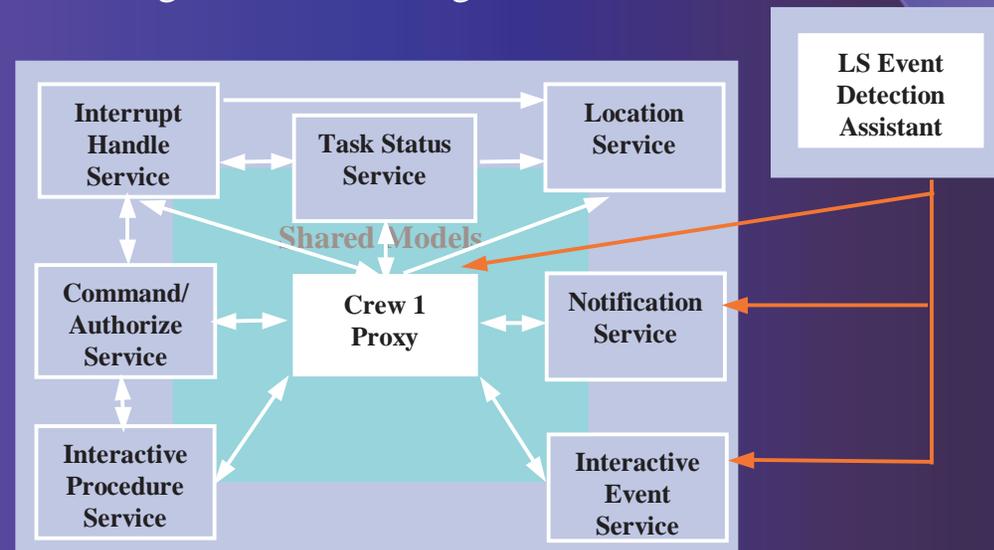
- Assists crew in responding to interruptions in normal operations
- Concepts for interruption handling
 - Determine if crew should be interrupted, and how intrusive interruption should be
 - Mark completion status of interrupted activities
 - Delegate an ongoing crew task by spawning a new automated task that "takes over" from crew
 - Assist crew in managing multiple, concurrent threads of activity



Control Assistant Software

Event Detection Assistant

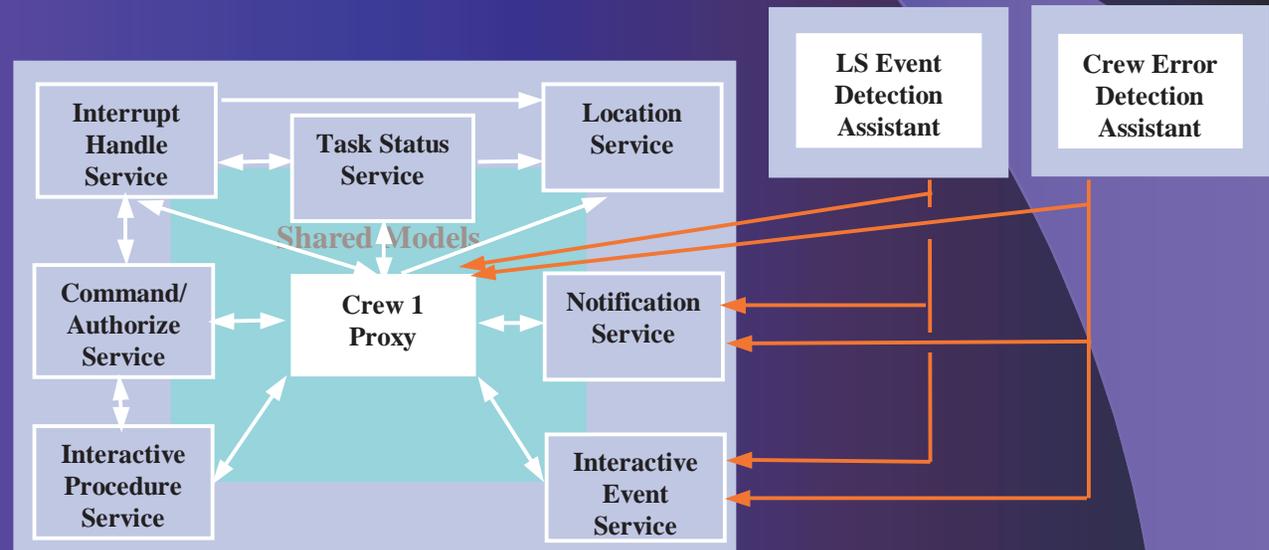
- Detects significant events (including anomalies) in life support systems
- Passes events to Notification Service of Crew Proxy
- Associates *simple events* with a single condition and a single time
- Detects *complex events* comprised of multiple simple events
- Ex: Detect automated slough event in Biological Water Processor



Control Assistant Software

Crew Error Detection Assistant

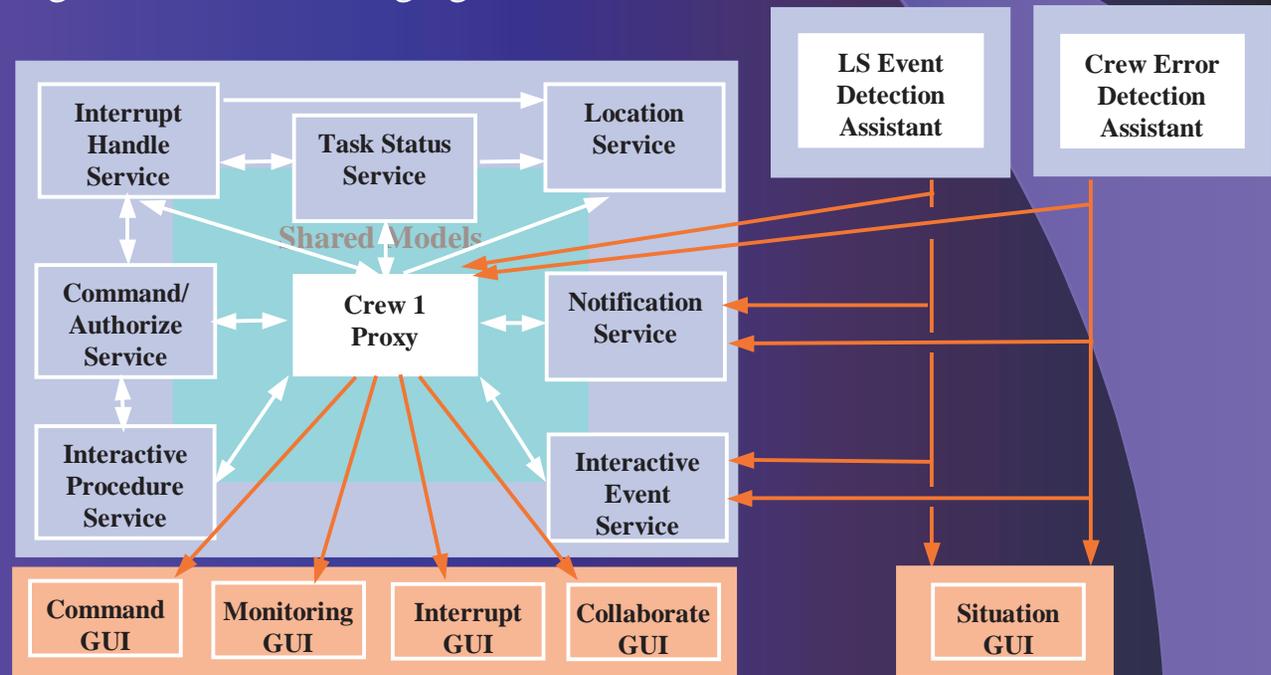
- Combines knowledge of crew tasks with knowledge of error pattern to detect and notify of crew error
- Example: BWP water flow inadvertently turned off during manual slough; no water flow over bacteria for significant time period



Collaborative User Interface

Collaborative User Interface

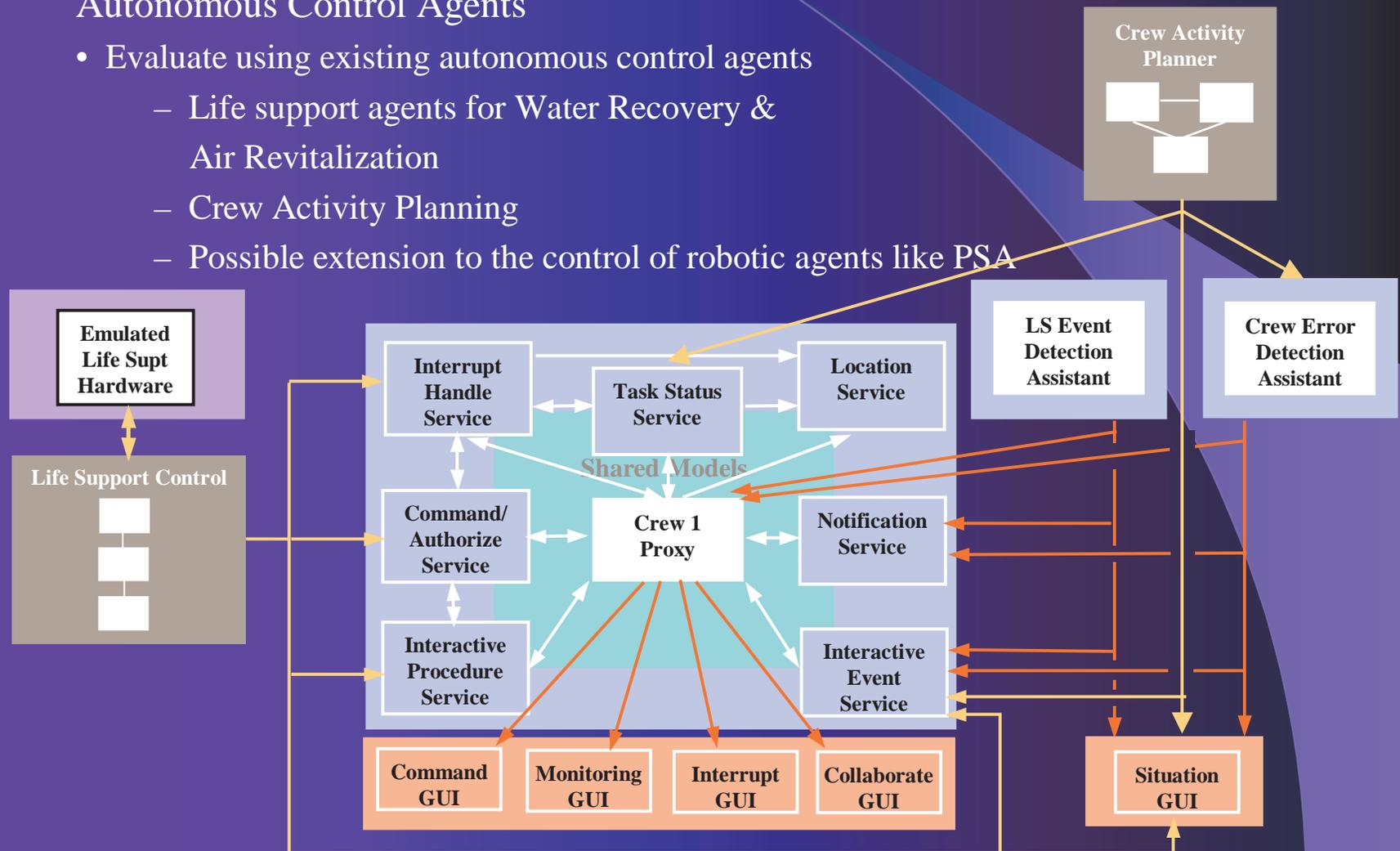
- Situation Interface for situation assessment; design from HCAAST project
- Monitoring Interface for event notification and task management
- Remote Commanding Interface; includes authorizing and issuing commands
- Collaboration Interface supporting crew-to-crew interaction
- Interruption Handling Interface for managing concurrent tasks



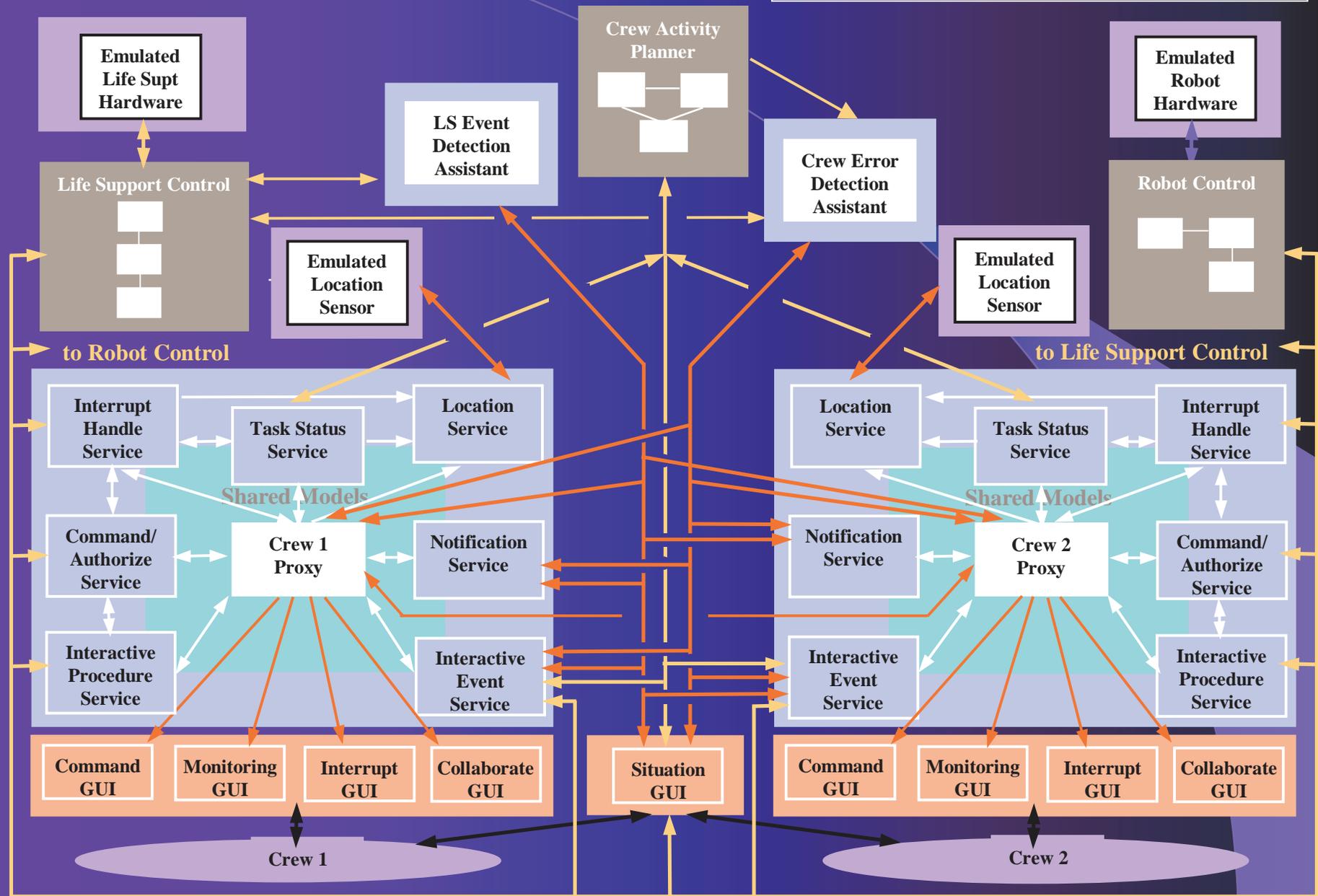
Autonomous Control Agents

Autonomous Control Agents

- Evaluate using existing autonomous control agents
 - Life support agents for Water Recovery & Air Revitalization
 - Crew Activity Planning
 - Possible extension to the control of robotic agents like PSA



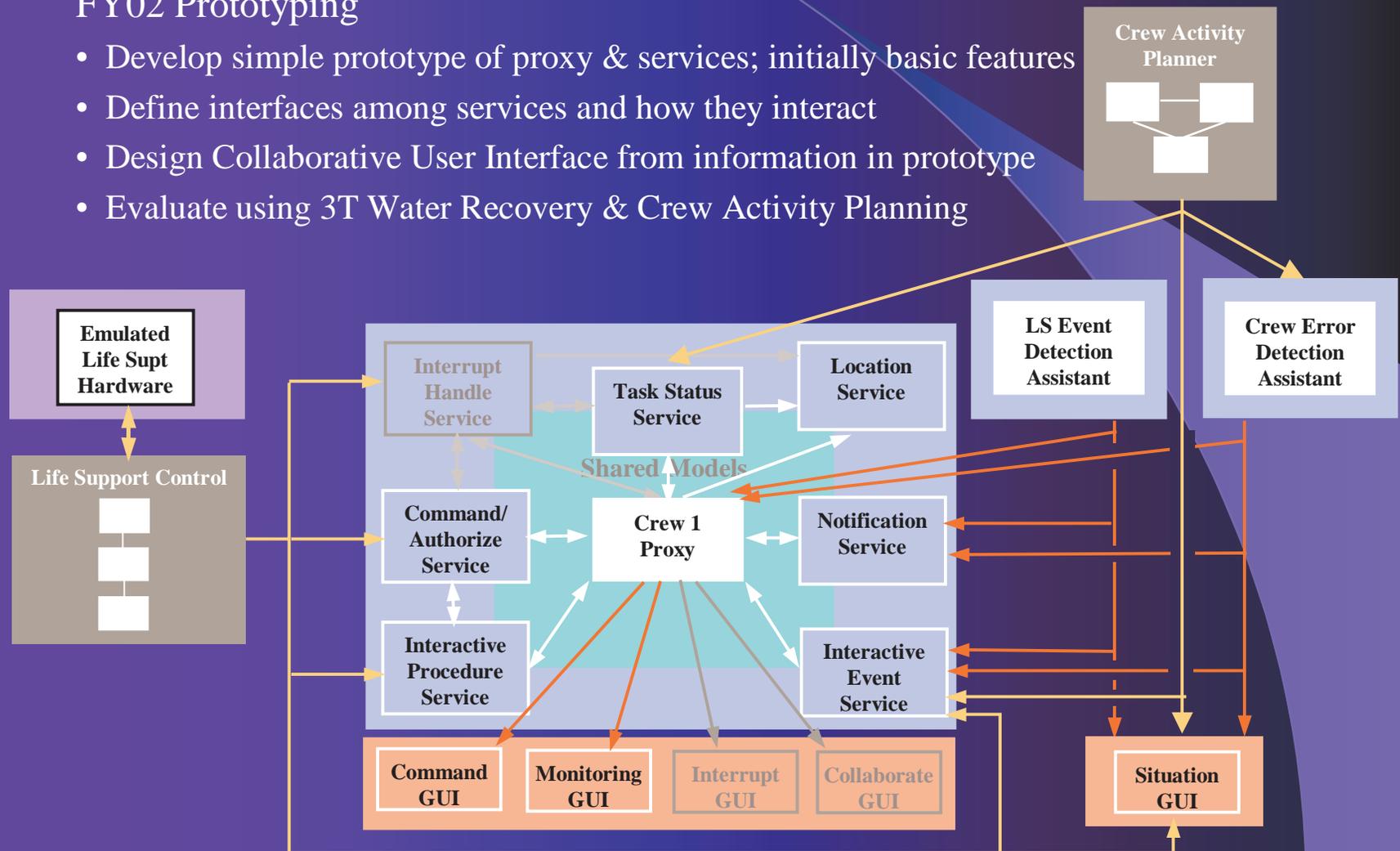
Architecture Design

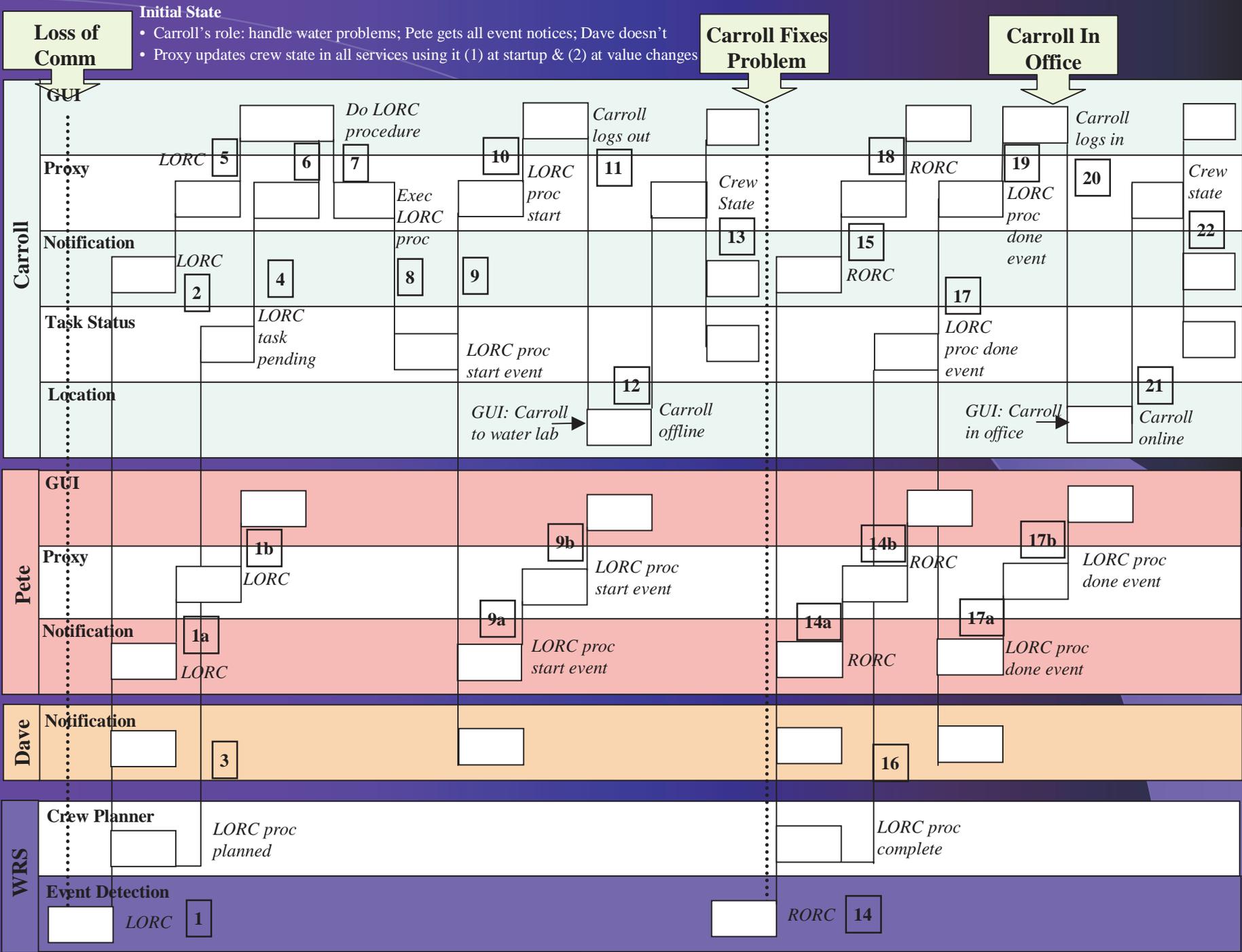


FY02 Collaboration Architecture

FY02 Prototyping

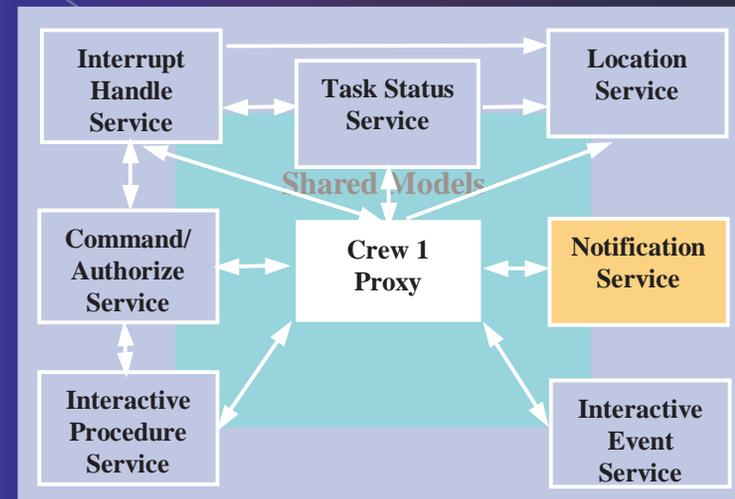
- Develop simple prototype of proxy & services; initially basic features
- Define interfaces among services and how they interact
- Design Collaborative User Interface from information in prototype
- Evaluate using 3T Water Recovery & Crew Activity Planning





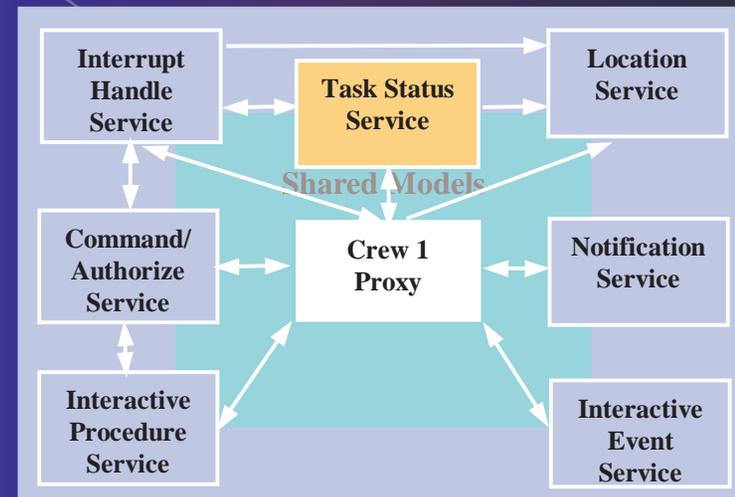
FY03 Interruption Handling

- Determine if human should be interrupted, and how intrusive interruption modality should be
 - Is human online
 - Importance & urgency of current task
 - Importance & urgency of responding to interruption
 - How close to completing current task
 - Crew preferences based on both task and temporal considerations
- Extension of Notification Service that addresses how interruptible is crew member



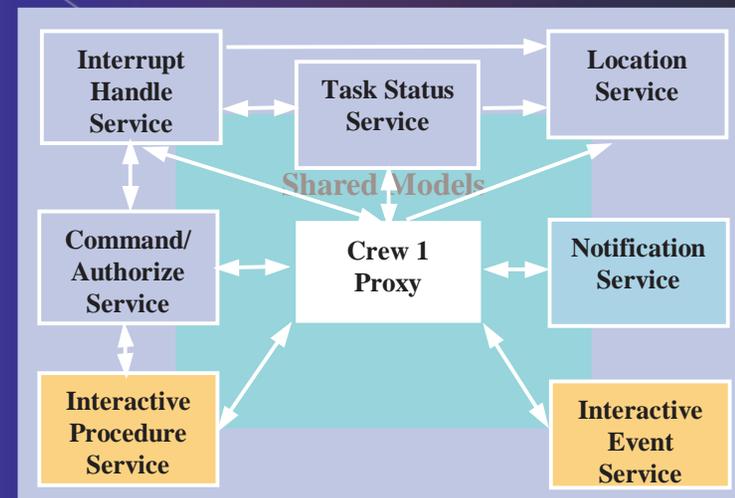
FY03 Interruption Handling

- Mark completion status of interrupted activities & add new activities in response to interruption
 - Status at interruption could include
 - Suspend
 - Delay with partial completion
 - Delay and start over
 - Abandon
 - Delegate/reassign
 - Used by crew and automated planning software for plan monitoring and re-planning
- Extension of Task Status Service to detect new status values for interrupted tasks



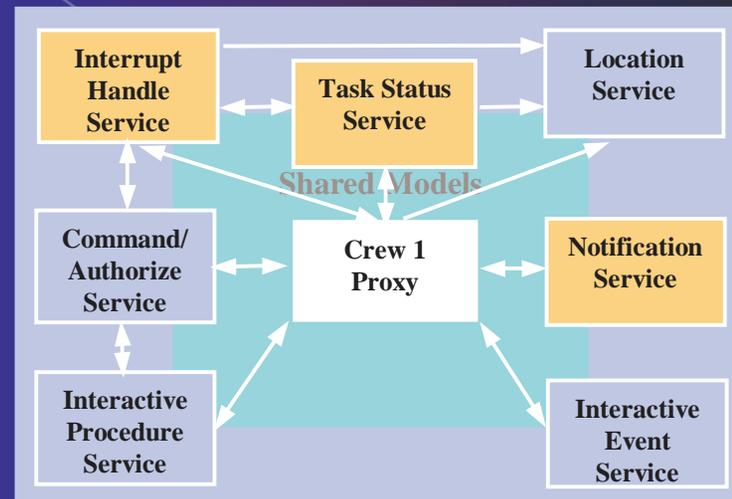
FY03 Interruption Handling

- Delegate task by initiating novel automation that "takes over" the old task
 - Combines features of Interactive Event Service and Interactive Procedure Service
 - Crew defines novel triggering conditions similar to Interactive Event Service
 - Crew associates an action with these conditions similar to Interactive Procedure Service
 - Example
 - Crew is part way through emptying gas tank manually for maintenance
 - Crew requests Crew Proxy to monitor for tank pressure below 10 psi, then turn off gas pump and notify him
 - Intended for temporary, one step mini-procedures, and not for new automated operating procedures



FY03 Interruption Handling

- Assist crew in managing multiple concurrent task “threads”
 - When an interruption occurs, interrupted crew member "converses" with his Proxy
 - Transition to new task
 - Select point in the ongoing task to halt
 - Store task state information when suspended
 - Determine whether to resume task, reschedule task later, or abandon task
 - Identify deadlines and critical events associated with old and new tasks
 - Reconfigure for both tasks
 - Notify crew of deadlines and critical events associated with suspended tasks
 - Resume the suspended task
 - Restore stored task state information to ease task resumption
 - Remind crew of what's left to perform on old task
- New service, possibly based on implementation of SharedPlan for discourse management



Summary

- **Year 1 Report - *An Architecture for Collaboration***
 - Documents the results of Collaboration Architecture Study
 - Contents: Architecture design; survey and evaluation of technology
 - Report can be downloaded from postdoc.arc.nasa.gov
Johnson Space Center/Robotics and Automation/DCI Public
- **Year 2 Iterative Prototyping - Collaboration Architecture**
 - Develop initial prototype of collaboration architecture with simple implementations of proxy and core services
 - Evaluate initial prototype by providing proxies for control engineers supervising the Advanced Water Lab
 - Revise initial prototype with more complete implementations of core capabilities; includes crew interface
 - Evaluate revised prototype using simulated life support
- **Year 2 Methodology - Design Techniques**
 - Evaluate animated mockup design technique from Dr. David Woods at OSU for representing collaborative designs