Agent-Supported Mission Operations Teamwork

Jane T. Malin and team
NASA Johnson Space Center

10/28/03

NASA Human Centered Computing
Task-level Review
University of Central Florida

Team Members

- Analysis and design methods
  - Carroll Throgesbery, Kathy Johnson, David Overland, Grace Hua

- System management agents and testbed
  - Debra Schreckenghost, Lapid Fleming and Luis Flores

- Information assistant agents and team tools
  - Arthur Molin, Kenneth Jenks, Kevin Kusy, Dan Smith

- Team tools and mission spin-offs
  - Patrick Oliver, David Overland, Gene Peter and Kevin Taylor (DV) and Kathy Johnson (SD – SMART project for SURGEON/BME)
Goals for Users

• Make automation by agents easy to use, supervise and direct
  - Smooth transitions into close supervision and intervention
• Manage information and communication to decrease distraction, interruptions, workload and errors
  - Smooth handovers across groups and shifts
• Reduce mission impact of off-nominal situations
• Increase morale and decrease turnover

Agents and Team Tools
Accomplishments

1. Collaborative agents - mixed initiative and creation of instructions for mediating agent
2. Methods for prototyping, evaluating and evolving socio-technical systems
3. Technology infusion: teamwork tools in missions
4. Demonstrations in simulation testbed

1. Agents that are Team Players

• System management agents: Mixed initiative interaction
• Mediating assistants: Easy to manage
Mixed Initiative Interaction

- Either user or agent can steer the interaction
- First step toward automation that works with the user
  - Balances both user needs and its own goals in deciding to act
  - Determines whether conditions support user-requested action
- Demonstrated: interaction to “fill out” and execute incomplete command
  - User: Execute the N₂ leak test.
  - ISMA: Now, or plan in the future?
  - User: Now
  - ISMA: OGS is shutdown and the N₂ leak test can be performed now. Should I start the test? [or, Please wait for OGS steady state]
  - User: Yes [or, No or Abort - stop interaction]
  - ISMA starts test when gets the OK. [or, ISMA aborts if test conditions change or higher priority action is needed]

Mediating Assistants

- Information manager: Intelligent Briefing and Response Assistant
  - IBRA Engine watches data stream for triggers and executes Act-Whenever actions
  - User-defined actions can be added for a particular domain
- Users specify how IBRA should handle events with Briefing and Response Instructions
  - Triggers: Patterns to recognize when event of interest has occurred
  - Actions: What IBRA should do when event occurs
    - Collect additional data, create log entries, make reports
    - Notify appropriate people that event has occurred
Briefing and Response Instruction (BRI) Editor

• Simple for discipline specialist to create a BRI
• Create trigger based on:
  - Previous data logs
  - Data declaration files
  - Existing BRIs
• Choose actions from palette
• Trigger and action structures can be simple or complex

Example: Specifying Leak Test Instructions

• Set of BRIs that are activated together
  - BRIs can activate and deactivate other BRIs
  - May be loaded into a special-purpose IBRA
• Measure time to reach threshold values
• Reporting with triggered alternatives
  - If the rate of change of pressure in the tank is too high, report leak
  - If the rate of change is nominal, report no leak
N2 Leak Test

- N2 Tank has been observed to refill several times in a 24 hour period
- Suspicion—the tank is leaking
- Test—refill tank and monitor loss of pressure
- Log rate of loss and time to reach milestone pressures

N2 Leak Test Procedure

- Note time and tank pressure at start of test
- Note time when pressure reaches 190 psi. Calculate and record rate of loss.
- Note time when pressure reaches 185 psi. Calculate and record rate of loss. End of test.
IBRA to track procedure

- One BRI for each data point
- BRI structure
  - Trigger: when pressure reaches designated point
  - Actions:
    - Calculate rate of change from previous pressure/time measurement (except the first BRI)
    - Record information as log entry in logger
    - Save pressure/time information for next calculation (except the last BRI)

Review existing IBRA

- View BRI in BRI Editor
- Decide to add another data point, at 187 PSI, for additional precision
- Create new BRI based on existing one
Create New BRI

- View new BRI
- Load BRI into IBRA
- Start IBRA
- Perform procedure

![Diagram of IBRA process]

Review log at end of procedure

- Final report from the IBRA agent in the Logger
- Report saved for later reference
- Used as decision support

![Log of IBRA procedure]

15

16
BRI Editor Advantages

- Users can create additional IBRA capabilities without involving software developers
  - Adjust existing BRIs to fit changing needs
  - Add new ones (from scratch, copy and modify)
- Users can inspect existing BRIs to understand them better

2. Human-Centered Methods

- Evolvable systems with “effortless specification”
- Requirements collection during prototyping
Low Effort Specification

• Enabling evolution of sociotechnical systems
  – Users can customize software support
  – No need to go through formal software changes
  – Extends to unanticipated uses (as tasks change)
  – Extends to new groups of users
• Case
  – Logger: Snippets for common long text entries

Using A “Snippet” Item

- Reuse text in log entry ("PPCR 2019 CEVIS Troubleshooting")
- Using a Snippet item is easier than re-typing the entry – fast and correct
- Using a Snippet results in a complete, consistent entry (both the document number and title)
- No need to consult external source to find document number
Adding A “Snippet” Item

- Highlight text to be replaced with Snippet string (“PPCR xxxx” from Quick menu entry, which attaches “paperwork” metadata)
- Click Snippet button
- Discover target text is not in list, decide to add it, select “Modify”
- Type in new text (“PPCR2011 Add TVIS weekly maintenance”)

Adding A “Snippet” Item: Benefits

- Adding a new Snippet is no more work than typing in log entry
- No new route is needed to add a Snippet (when user sees it isn’t already on list, user selects “add new Snippet item”)
- Snippets allow us to learn additional user requirements as they use the logger (recurring Snippet items are candidates for Quick menu entries)
Requirements from Prototyping

- Prototypes are developed to discover requirements
  - Inspection
    • Application itself is an inspectable artifact
  - Evaluation
    • Allows discovery of new requirements
  - Use
    • Prototype creates new requirements by changing work practice

- Prototypes are not mission-hardened
  - Inappropriate as product
    • Messy under the hood
    • “Layered” form evolution, and incomplete
  - Need to capture requirements in a document

Requirements Benefits

- A Requirements Specification allows hand-off to production development team
- Some requirements cannot be implemented in current prototype due to early design choices
- Helps with transition to production and operations environment
- Aids identification and clarification of requirements which come out of prototype inspection
Progress on WorkIT Requirements

- Analysis of design artifacts begun
- Specification format under development
  - Combining both high-level and "as-built" requirements
- Identification of requirements vs. design choices
  - Rationale capture by Design Team and developer
  - Non-implemented reqm'ts will be specified with less detail
- Will do example trace from design artifact to specification

A Methodology for Generating Software Requirements from Prototyping: WorkIT Case

2 General Prototyping Approaches
  2.1 Exploratory prototype
  2.2 Experimental prototype
  2.3 Evolutionary prototype
  2.4 Prototype construction techniques

3 A Prototyping Case Study
  3.1 Background
  3.2 Early exploratory prototyping
  3.3 Second prototyping iteration
  3.4 Prototype evaluation
  3.5 Expansion of customer base
  3.6 Incremental development prototyping

4 Generating Requirements from Prototypes
  4.1 Understanding the task
  4.2 Basic functionality
  4.3 Data model and user interface
  4.4 Usability
  4.5 Utilities
  4.6 Look and feel

5 Artifacts of Prototyping
  5.1 Mission operations documents
  5.2 Walkthrough evaluations
  5.3 Early requirements specification document
  5.4 Hands-on evaluations
  5.5 On-line feedback
  5.6 Design team meetings
  5.7 Design sketches
  5.8 Requirements rationale
Team Products and Requirements

- Implications for Design Team of need and designs
  - Products of design team needed for requirements
    - General guidelines as well as specific design and implementation
    - Interface, functional and back-end requirements
  - Team Products are Design Artifacts
    - Annotated Designs, Screen Shots
    - Alternative designs
    - Wish list for reimplementations is important source
    - Prototype as artifact
  - Rationale needed as prototype can’t capture all requirements
    - Early design choices preclude implementation of some requirements

3. Technology Infusion of Teamwork Tools

- Exploration Planning and Operations Center (ExPOC) use of WorkIT in NEEMO-5
- Preparation for operational evaluation of WorkIT version 4.0
- Logger evaluation
- Preparation for operational evaluation of Logger version 1.0
WorkIT Evaluations

- BME walkthrough evaluation with some free use (videotaped) - WorkIT 2
- Stuffing the BME database with content from past issues (designers and feedback forms) – WorkIT 3
- ExPOC evaluation in training walkthrough for NEEMO-5 Mission (videotaped and feedback forms)
- ExPOC drop-in interview evaluation during NEEMO-5 Mission, June 2003
- BME evaluation in operational use – WorkIT 4 (Surgeon/BME in ISS Expedition 8)

WorkIT in NEEMO-5
WorkIT support in NEEMO-5

- WorkIT prototype use and informal evaluation during the 6/03 NASA Extreme Environment Mission Operations (NEEMO-5) mission in ExPOC
  - NASA crew living in Aquarius Underwater Research Facility off the coast of Key Largo, Florida.
  - Practice for long-duration space habitation, research and construction, with underwater structures simulating Space Station assembly activities
- WorkIT tool provided information management and action tracking for the NEEMO topside operations teams at Johnson Space Center and in Florida
  - WorkIT was not specifically designed for this group of about 10 people
  - Primary use: Handling issues that involved assigning actions
- Evaluations
  - Videotaped training walkthrough that produced a wealth of feedback
  - Drop-in interviews and observations of three users during the mission
- Results
  - Surprised and delighted with automated services and low overhead
  - Found it intuitive and easy to use, even without using the tutorial
  - Found that it reduced rather than increased workload in handling actions
  - Eager to use during upcoming missions and to help prepare for missions

WorkIT 4.0

- New capabilities
  - NOTE item for entering a text item without attached file
  - Automatic creation of Status Report
  - More comprehensive search that includes Task Logs
  - WorkIT tutorial and Help system framework
- Improvement in
  - Feedback system (from users to developers)
  - Navigation and UI presentation
  - Capacity for handling large status and task logs
  - Error handling
- Accommodation of new configuration in Surgeon/BME database and server platform
Console Logger 1.0

- Logger Objectives
- Logger Features
- Example Views (Quick Entries, Searches, Reports)
- Evaluation
- Future Plans

Logger Objectives

- Support console logging tasks
  - Assist entry of common log note types
  - Assist in making complete, consistent entries
  - Search log notes based on topic, author, activity, discipline
  - Make logs available to flight controllers not on console
- Support report generation from log notes
- Make logger accessible to other console support agents
  - Automated log notes from telemetry, ELog messages (IBRA)
  - Integration with WorkIT, ReportMaker
Logger Prototype Features

- Web based application (accessibility, user acceptance)
- Data base orientation
  - Each time-stamped entry is a separate record
  - Records organized by discipline and activity (flight increment, simulation)
  - Users can search entries by topic, author, discipline, activity
- ReportMake integration
  - Selected log notes go to a specific part of the Handover Report, depending on the category assigned to them
  - Additional report types (Daily Summary) are definable
  - Reports are converted to Word format for printing
- Low effort entry for common types of log notes

Assistance for Common Entry Types

- Quick menu selection
- Auto text entry
- Auto handover category marking
- No need to enter timestamp unless different from now
Logger Searches

- Basic search implemented
- Full info on “hits”
- Capability to view “hits” in context of other log notes on same shift
- Future: control search of time-frame, disciplines, activities; re-execute common searches

Automated Reports

- Report format specifies log entry queries to include in Handover Report
- Specific text entry areas allow direct input into report
- Later, reports are printable in Word format
Logger Prototype Evaluation

• Purpose: Formative rather than summative
  – Identify improvements needed in prototype
  – Not to compare performance to other software
• Method: Demonstration (with flight controllers exercising the logger functions) and Interview
• Evaluators: Four flight controllers
• Videotaped for subsequent analysis of specific features
  – Timing of system responsiveness
  – Reviewing for points where user interactions were difficult

Logger Progress/Plans

• Enhancements
  – Quick menu: New entries with fill-in formats for console support and AFD notes; arrow icon; highlighting
  – Snippets: Re-implemented to fix persistent bugs
  – Handover Report: Refined format for Surgeon/BME
  – Dedicated server to ensure fast system response
  – Help pages
• Surgeon/BME evaluation in mission simulations soon
• ExPOC evaluation in undersea mission soon
• Longer Range
  – Multi-discipline views of logs (e.g. front and back room)
  – User interface for customizing menus and report formats
  – IBRA-base automated logging of routine, telemetry-based entries
  – Assistance for tracking paperwork, to-do lists (make use of artifacts from users)
4. Demonstrations

- Simulation testbed
- FY02 Demonstration
- FY03 Demonstration
- Integrated Demonstration support

Simulation Testbed

- Testbed simulates operation of life support system hardware (air and water systems)
  - Dynamically interacts with ISMAs, to test new ISMA capabilities in teamwork to intervene during anomalies
  - Capability to simulate combined, cascading and global effects of local problems
  - Case server (Java) utility saves and replays simulation cases
    - Capability to remotely pause and resume supports scripted dynamic interaction between ISMA and user in FY03 demo
- Enhanced biological water processor model will simulate more off-nominal scenarios
  - Uses generic library of components from Model-Based Hazard Analysis project (ECS program), to support broader variety of types of failures and degradations in system
Model of Biological Water Processor

Configurable Failable Components

- Styles of modeling failures and degradation
  - Discrete changes triggered by failures and problem inputs
    - Immediate or delayed changes to state, behavior mode or control regime
  - Continuous degradation triggered by failures and problem inputs
  - Nontemporal algebraic relations
    - Performance level affected by conditions
    - Failures to operate or change upon inputs: stuck flags
    - Random variation in measurement or input
- Degrading and regenerating processing performance
- Reactors and separators with multi-component mixtures
  - Add and remove contaminants in rapid fluid composition changes
  - Migrate products, gas or liquid to wrong outflow
  - Imbalance process with feed or flow reversal problems
- Resource providers with alternative methods for reacting to excessive demands from multiple loads
- Leaks as specifiable additions to simulation scenarios
FY03 Demo

- ISMA autonomous control of air processing systems in space, with joint anomaly management
  - Mixed-initiative dialogue with ISMA to execute a leak test for Oxygen Generation System
    - Loose command and response leads to agreement on conditions and timing
  - IBRA helps maintain situational awareness of system status and ISMA operations with manual intervention
    - Automatically carries out customizable instructions to collect and present information in web applications
    - Ground controllers easily specify and activate instructions
      - Leak Test Instructions - triggered act-whenever requests
      - Actions include Logging and Report on Leak Test timing and states
        - Start and complete conditions, test start and stop, level measures

Demonstrated Benefits

- ISMA aids anomaly response and user intervention with mixed initiative dialogue
  - Basic support for moving from loose commanding to tight and complete plan
  - Supported by ISMA capabilities to delay and abort
- IBRA aids monitoring, logging and report generation for Leak Test intervention
- Tool suite provides basis for further teamwork on anomaly and knowledge capture
  - Reference materials and special and periodic reports
  - Searchable logs and action items with metadata
  - Customizable IBRA instructions
Plans for FY04-05

- Continue process of spin off into mission operations
- Advance technology infusion (HCC) methods
  - Use of teamwork artifacts for prototyping and customizing
  - Requirements collection that influences prototyping methods
  - "Effortless specification" strategies for evolvable systems
- Advance collaborative agent capabilities
  - Mixed-initiative interaction for joint problem solving and command completion.
  - Safety conscious agents that use simulation for checking response plans and resuming interrupted operations