Real-time Cooperative Behavior for Tactical Mobile Robot Teams

April 2000

Ronald C. Arkin and Thomas R. Collins
Georgia Tech

Overview

- Usability studies
  - test scenarios
  - procedures and plans
- Real-Time Advisor
- MissionLab
  - Enhancements made in support of studies
- Robot Platforms
  - Urbie developments and SICK sensor

Usability Study Objectives

- To validate that *average users* are capable of generating effective robotic missions for TMR scenarios using *MissionLab*
- To provide effective methodologies that *evaluate* the performance of TMR systems from an end-user's perspective
- To provide methods and tools in support of *cognitive modeling* of the interaction of users with TMR systems
- To create *meaningful TMR applications* that can serve as prototypical tasks for the research community
- To suggest refinements to the *MissionLab* GUI

Usability

- A combination of:
  - – ease of learning,
  - – high speed of user task performance,
  - – low user error rate,
  - – subjective user satisfaction,
  - – user retention over time

    (Schneiderman 92)

Georgia Institute of Technology

Honeywell

Usability Requirements

- Understanding the users' abilities and goals through user and task analysis
- Involving the user in participatory design where feasible
- Preventing user errors
- Optimizing user operations
- Keeping the locus of control with the user
- Assisting the user to get started

(Hix93)

Experimental Testbed

- I-Observe (Interaction, OBServation, Evaluation, Recording and Visualization Environment) interface usability evaluation environment, consisting of:
  - Logging tools
  - Analysis tools
  - Visualization tools

Usability Lab

Georgia Institute of Technology

Honeywell

Experimental Procedures

- Administered by third party
- Uniform introduction to toolset provided to participants
- Participants given one task at a time
- Left alone in the usability lab to complete
- Observed via one-way glass and video camera

Georgia Institute of Technology

Honeywell

TMR Usability Experiments

- Two phases
  - Phase 1 test scenarios (underway)
    - Back-and-forth and CoC approach (tutorial examples)
    - Hospital approach
      - single robot
      - tests map interface for placing waypoints
    - Airport incursion
      - multiple robots, also with map interface
  - Phase 2 test scenario
    - Hostage counter-terrorism (room searching/clearing)
    - Multiple robots with complex interactions
- Test subjects specify missions using Configuration Editor
- Analysis of verbal protocol of participants speaking aloud to provide information to improve interface

Behaviors for Usability Studies

- Behavioral States
  - Alert
  - EnterAlternateHallway
  - EnterRoom
  - GoTo
  - GoToSoundSource
  - LeaveRoom
  - MarkDoorway
  - MoveAhead
  - MoveAway
  - MoveCompassHeading
  - MoveInFormation
  - MoveToward
  - ProbeObject
  - ProceedAlongHallway
  - ProceedAlongPath
  - Stop
  - Telop
  - Terminate
  - TrackObject
  - UnmarkDoorway
  - Wander

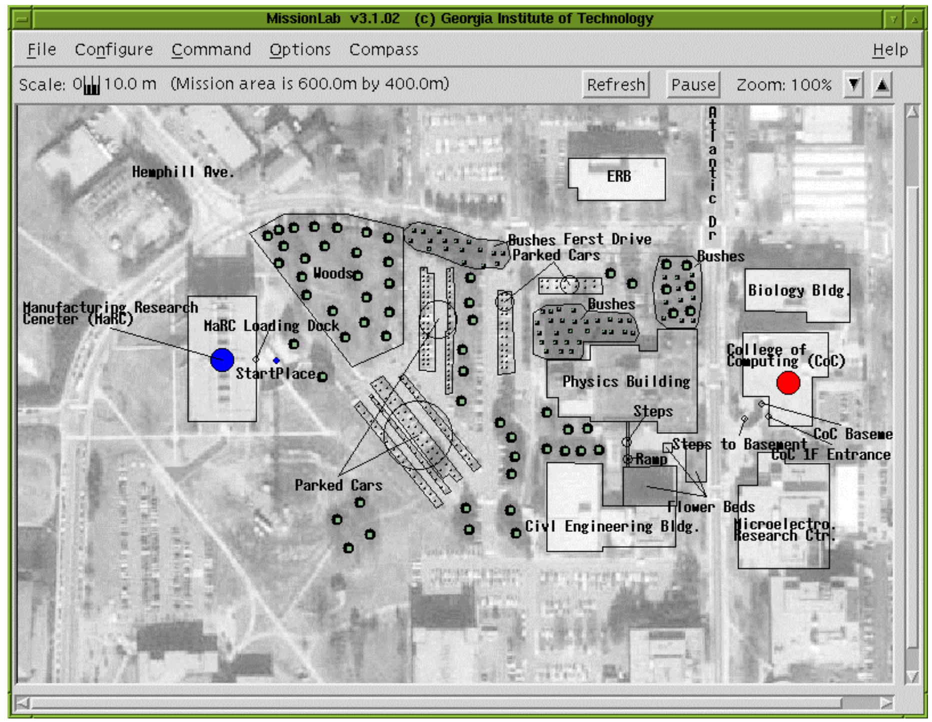
- Triggers
  - Alerted
  - AtDoorway
  - AtGoal
  - AtGoalInFormation
  - AwayFrom
  - Detect
  - DetectAlternateHallway
  - DetectSound
  - HasTurned
  - Immediate
  - InHallway
  - InRoom
  - IsFacing
  - MarkedDoorway
  - MovedDistance
  - Near
  - Never
  - NotDetectAltHallway
  - NotDetected
  - SenseSignal
  - TelopComplete
  - UnmarkedDoorway
  - Wait

Georgia Institute of Technology

Honeywell

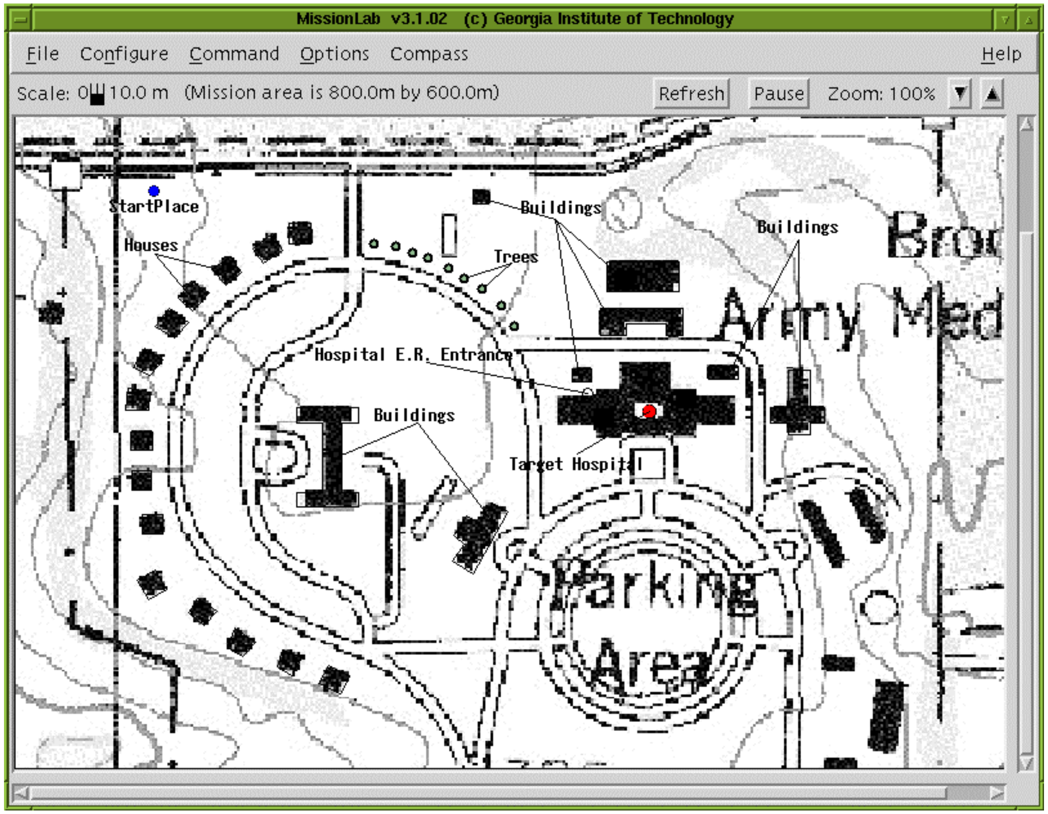
Tutorial scenarios

- Back-and-Forth
  - Simple exposure to basic robot behaviors and configuration editor
  - “Start” plus two states
  - Allows user to create a successful simulated robot with help as required
- CoC (College of Computing) approach
  - Introduces overlays and waypoint designation
  - Test administrator still available for assistance as needed
  - Requires user to think spatially and consider the locomotive capabilities of the robot



Hospital Approach

- Robot is to approach rear (ER) of Ft. Sam hospital, taking advantage of cover
- Conceptually similar to CoC approach, but
  - Map, not photo
  - User works on their own
  - Must consider use of cover/concealment and avoidance of occupied areas

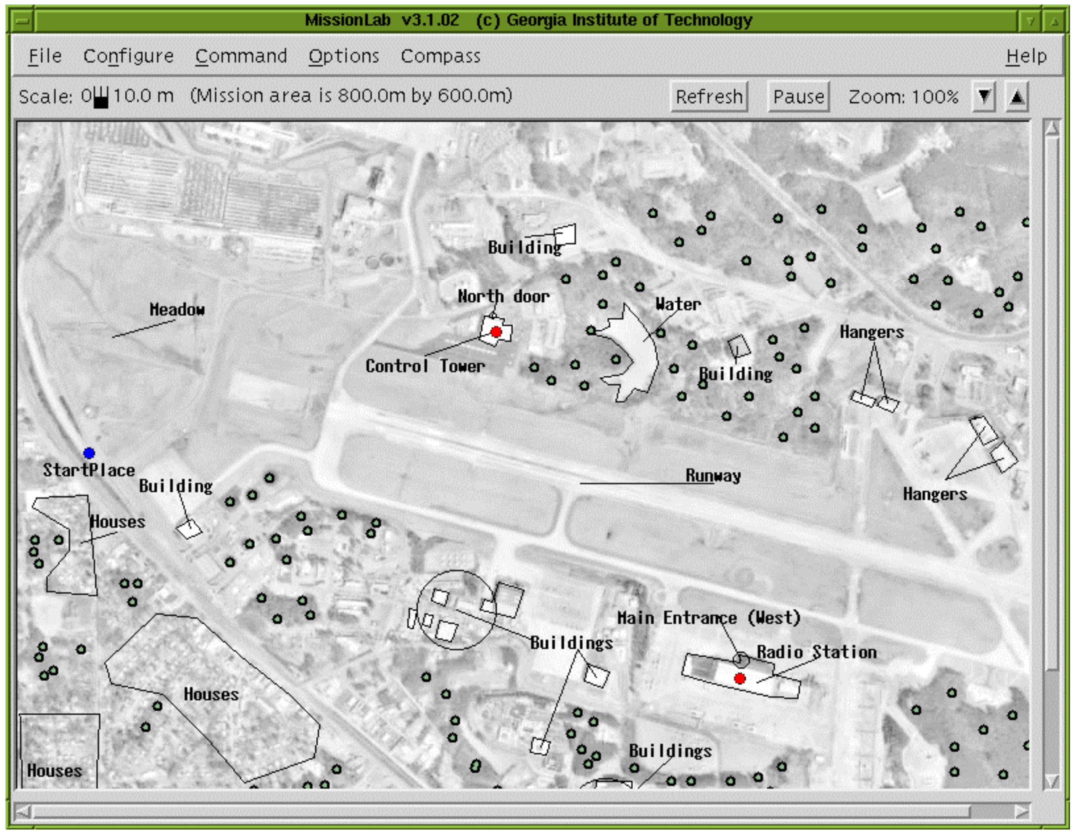


Georgia Institute of Technology

Honeywell

Airport Incursion

- User's task is to use two robots to monitor different locations
- Requires user to configure multiple robots
- Also requires consideration of good locations for surveillance

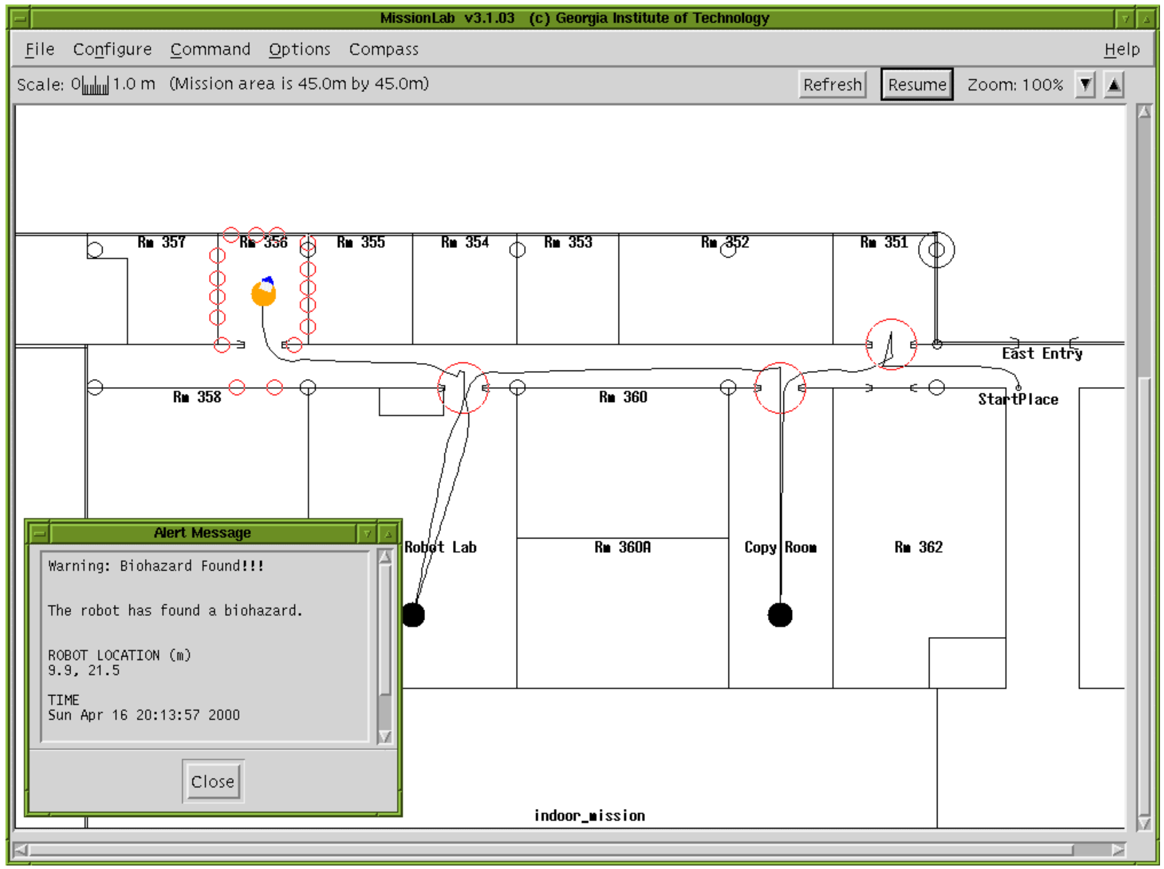


Georgia Institute of Technology

Honeywell

Phase 2 scenario

- Initial task is a single-robot biohazard search, as shown
- Second task uses a larger search area and two robots
- In both cases, the user does not see the map in advance and cannot use waypoint designation

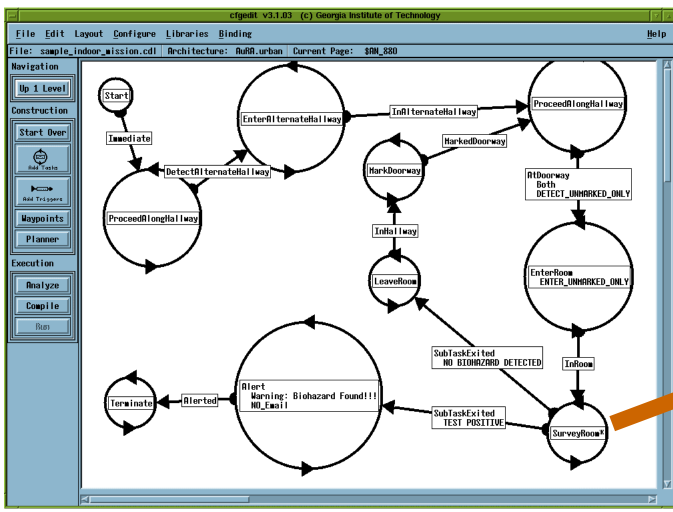


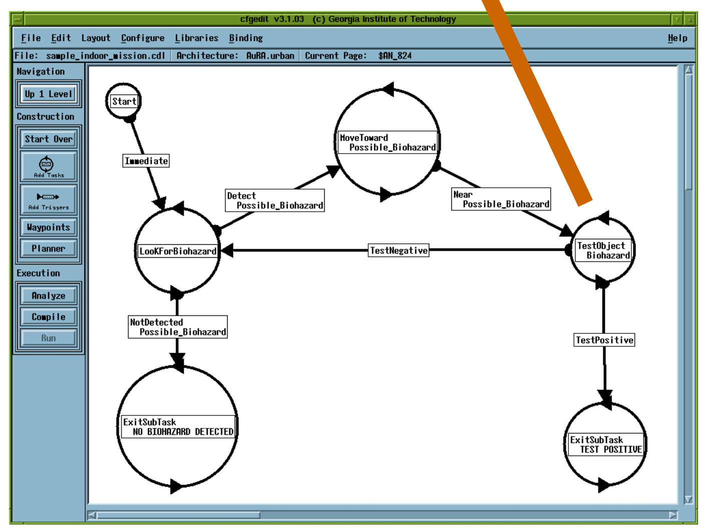
Georgia Institute of Technology

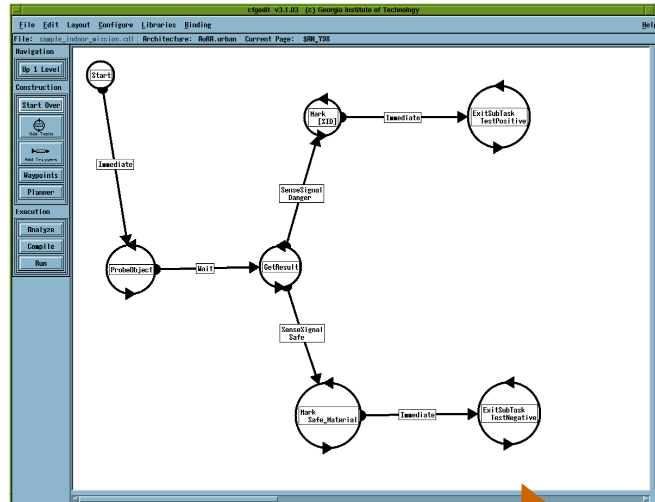
Honeywell

Use of Abstractions in task

- Test subjects are given sub-FSAs to use -- hierarchical behaviors that can be reused
- Main FSA, sub-FSA (SurveyRoom), and sub-sub-FSA (TestObject) shown here







Georgia Institute of Technology

Honeywell

Usability Study Schedule

- Scenarios and supporting MissionLab features completed
- Preliminary usability studies (with lab personnel) completed
- Refinement of scenarios and behaviors completed
- Experimental studies underway
- Analysis - late summer

Preliminary results (as of 14 Apr.)

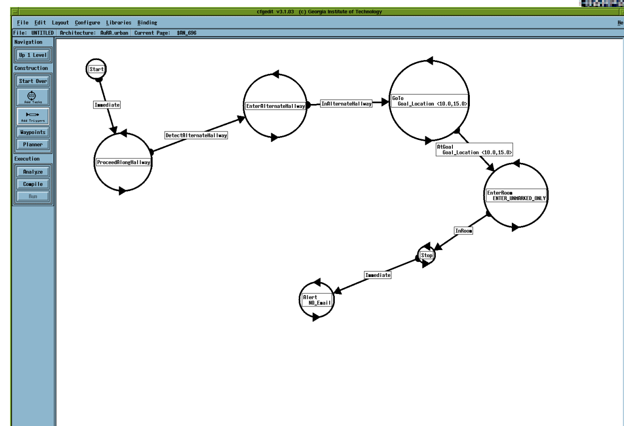
- 10 subjects have been tested, all in phase 1
- Phase 2 subjects begin April 21 (10 this semester)
- Both phases continue in summer (3 scheduled to date)
- All subjects are undergraduates (some ROTC)
- 90% completed hospital task successfully
- 80% completed airport incursion task successfully
- Avg. number of compilations:
  - For task 1 : 1.4
  - For task 2: 2.4

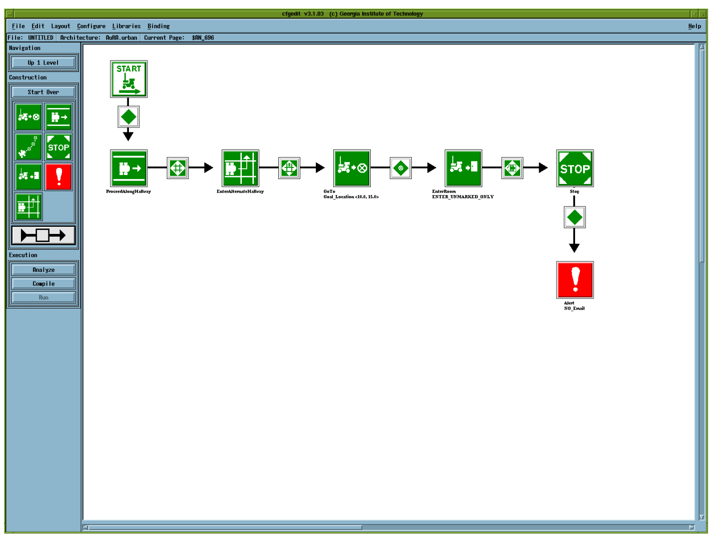
Experimental Details

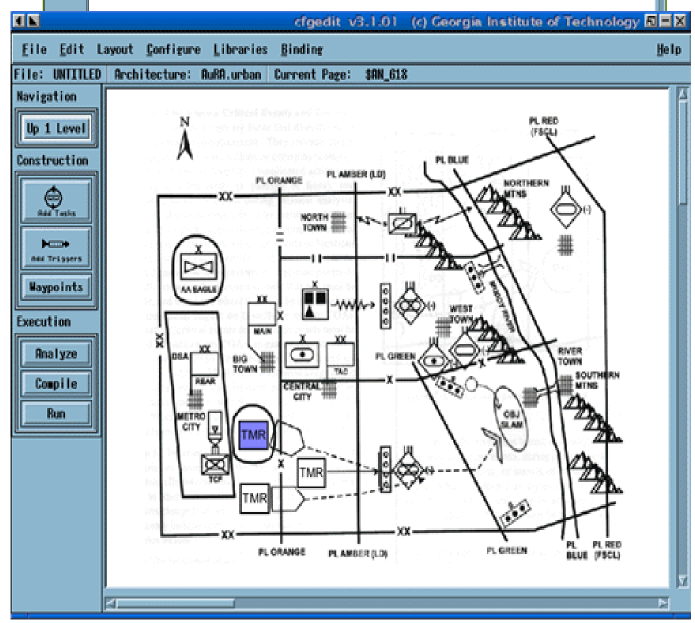
- Premature to draw conclusions -- no statistical analysis of variance and correlation has yet been made
- Data being gathered include
  - plotting of waypoints
  - quality of mission data (time for execution, etc)
  - compilations, start-overs
  - error-messages
  - edits (copy, paste, duplicates, setting state & trigger values)
  - subjective evaluation (post-test questionnaire)

Alternate User Interfaces

- Student project (not TMR-funded) is considering the use of alternate user design metaphors
- Questionnaire will be given to a subset of usability test subjects (probably all ROTC students)





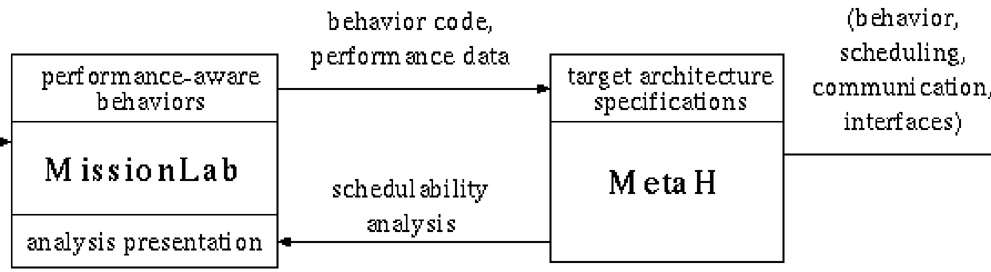


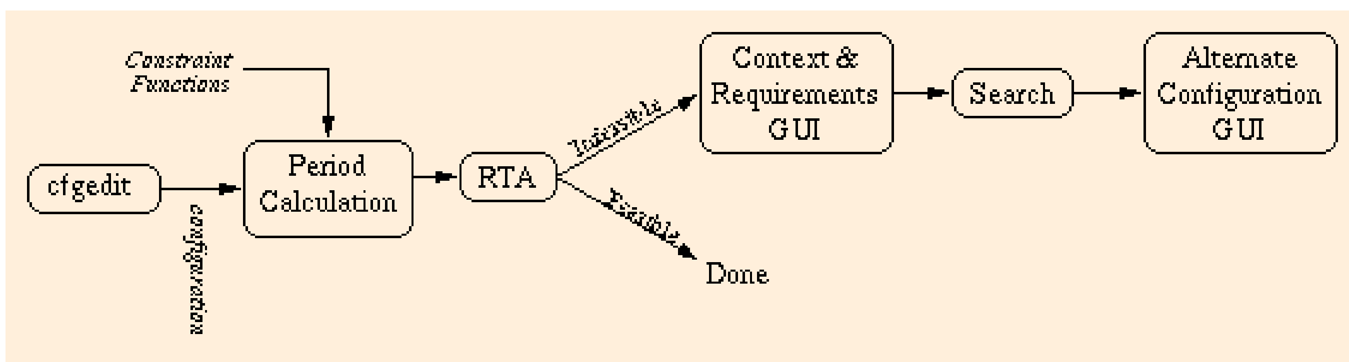
Georgia Institute of Technology

Honeywell

Real-Time Behavioral Specification

- Honeywell Technology Center's RT-MLab adds real-time analysis
  - ensures computation feasibility for arbitrary configuration
  - intelligently advises user how to change configuration to meet computation limitations



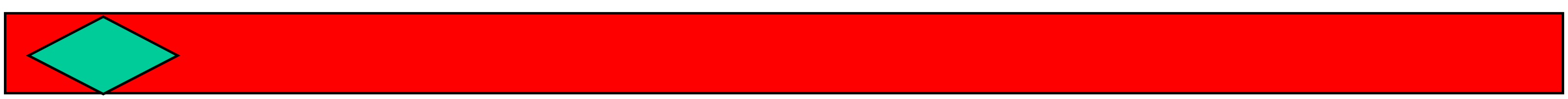


Problem: Processor Overload

- User can configure any set of behaviors s/he wants.

> Result: user can build behavior configurations that overload system, execute too slowly, and fail.

- Guaranteed computation of behaviors is critical to mission success.
- Solution: **RT-MLab** (Completed effort)



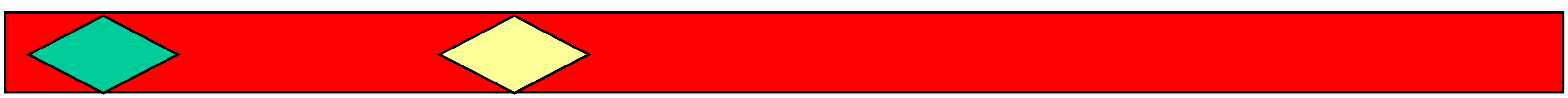
Processor Guarantees | Behavior Guarantees | Mission Guarantees | Mission Design

Problem: Capability Overload

- User can select any parameters s/he wants for sensors and actuators.

> Result: user can demand impossible performance, and fail.

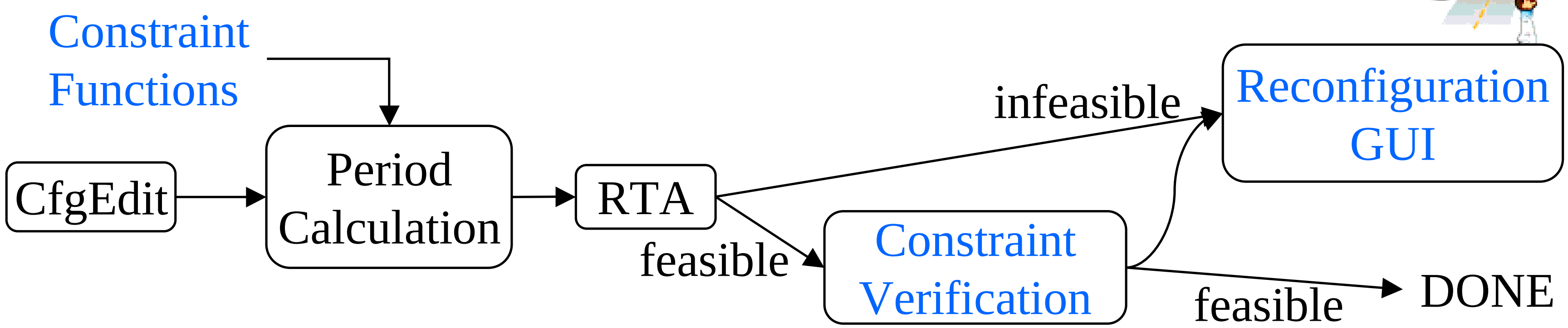
- Guaranteed performance of all devices (not just processor) is critical to mission success.
- Solution: **RT-Advisor** (Current effort)



Georgia Institute of Technology

Honeywell

RT-Advisor



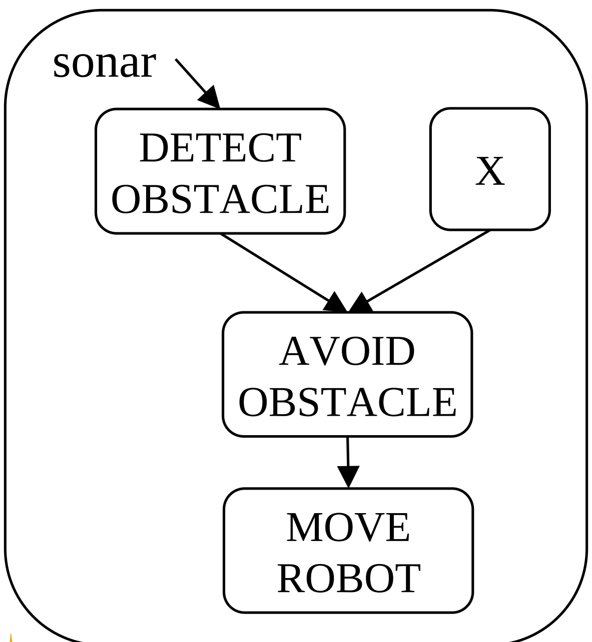
- Real-Time Advisor:
  - Constraint representation captures limits on both user and hardware parameters.
  - Advice to help user find feasible configuration:
    - How to change parameter values.
- Constraints express hardware limitations and user-expressed behavioral requirements.

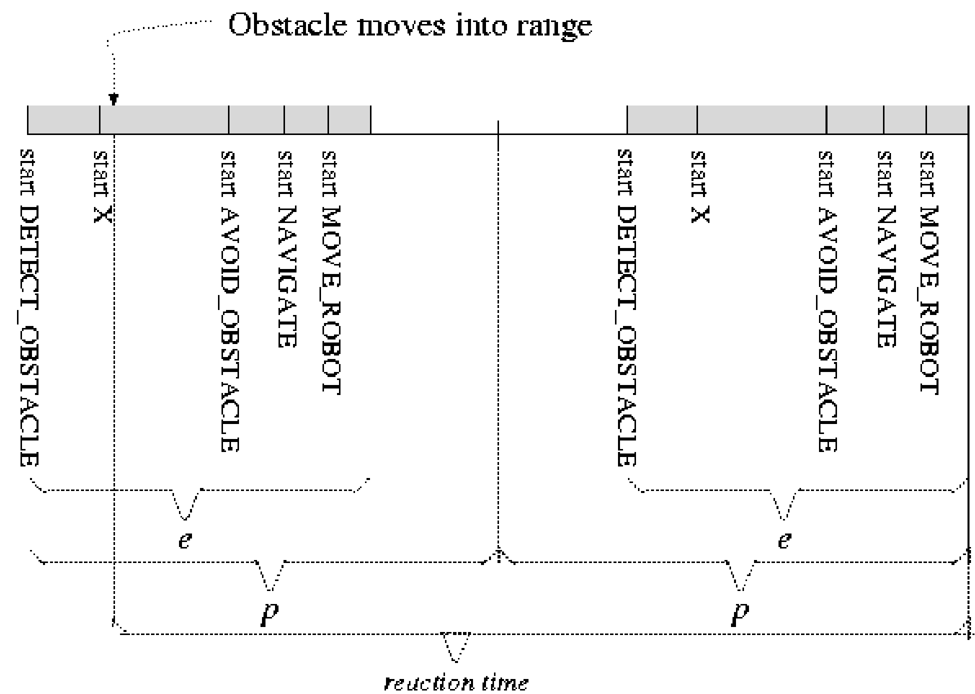
Where Do Constraints Come From?

- Hard constraints come from hardware specifications.
  - Sonar ping frequency, visibility distance.
  - Motor acceleration, velocity.
- Soft constraints come from behaviors & user-specified parameters.
  - Desired velocity, accuracy.
  - Desired behavior configuration.
- Constraint info is added to the behavior library.

Constraint Examples

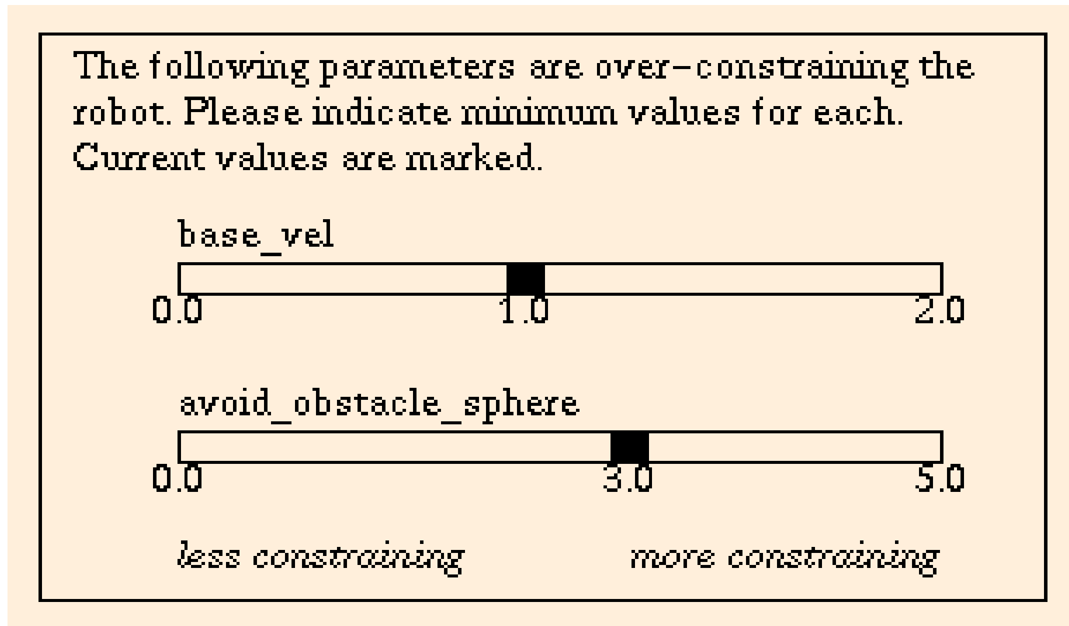
- User-specified parameters may result in infeasible robot executables.
  - e.g., velocity relates to sonar visibility.
    - ⇒ “Obstacle detect” must run frequently enough so robot can see obstacles.





RT-Advisor: Reconfiguration GUI

- User makes tradeoffs to meet real-time constraints.
  - Slow down robot (increasing reaction time).
  - Reduce sensor rates.
  - Adopt alternate behavioral ensembles.



Operator aid display concept

Georgia Institute of Technology

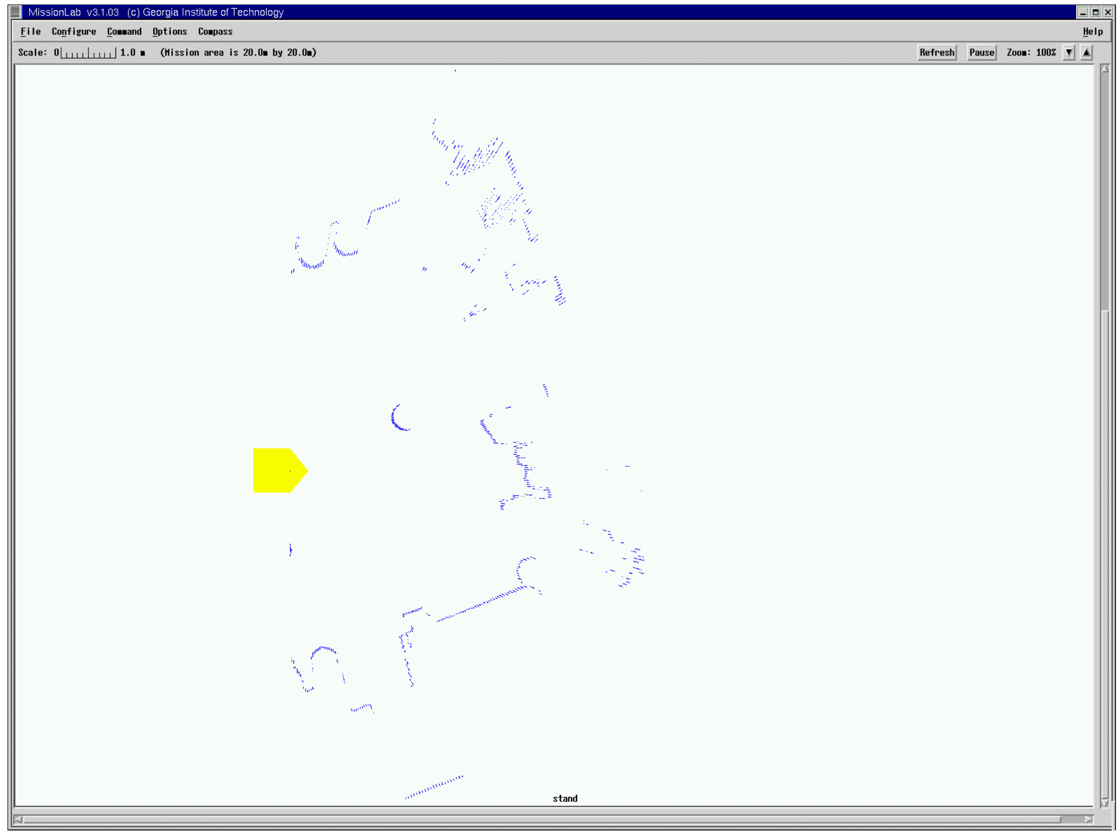
Honeywell

MissionLab Enhancements

- Developed primarily to support usability studies:
  - Window to report current state of robots
  - Start Over button
  - Waypoints button
  - New Control Measures: Wall, Room, Hallway
  - CDLReplay function
  - Data logging capability
  - New Run window - Ask for overlay
  - Other states & triggers
- Developed under other funded research and potentially useful:
  - Path Planner
  - Motivation-related states & triggers
  - Sound-related states & triggers

Robot Platform Developments

- SICK sensor has been integrated
  - Software integration – MissionLab
  - Hardware integration – portable, but focusing on Urbie
- Integrating Triclops vision system into MissionLab



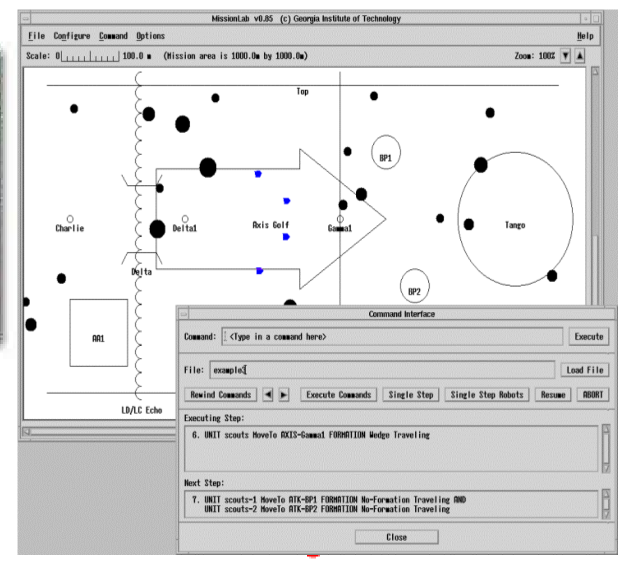
Future Work (Follow-on Interests -- GT)

- **Usability Related**
  - Run-time environment experiment design, testing, analysis, and support
  - Extension of ongoing human factors study
- **Mission Specification Related**
  - Interface to Operator Control Unit
  - Integrated multirobot mission development environment
  - Mission specification for new TMR missions (configuration of a robot executable suitable for a mission)
  - Integrated OCU/mission specification usability-tested interface targeted at novice military end users
  - Ongoing development of library of useful perceptual schemas and behaviors for new TMR missions
- **Adaptation and Learning**
  - Leverage DARPA-MARS work in MissionLab

Adaptation and Learning Methods

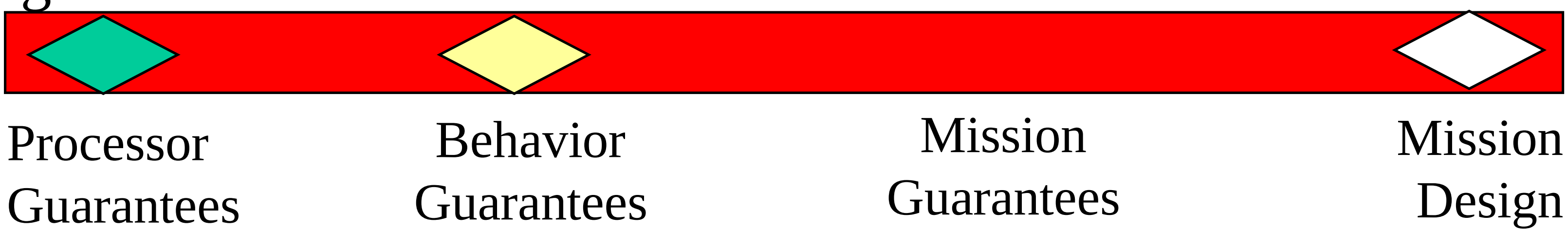
- Case-based Reasoning for:
  - deliberative guidance ("wizardry")
  - reactive situational- dependent behavioral configuration
- Reinforcement learning for:
  - run-time behavioral adjustment
  - behavioral assemblage selection
- Probabilistic behavioral transitions
  - gentler context switching
  - experience-based planning guidance


Available Robots and *MissionLab* Console

Future Work (Follow-on Interests - Honeywell)

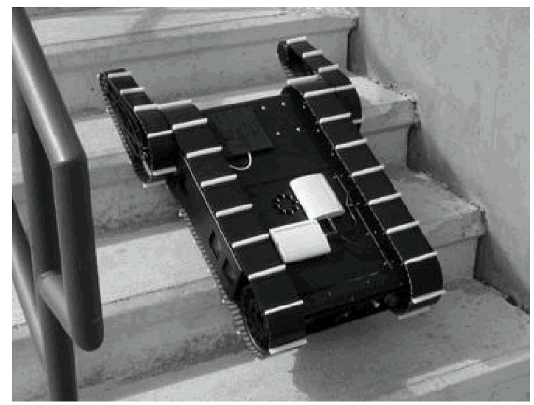
- Advanced guarantees, closer to ideal goal of mission guarantees.



- More demanding mission requirements & robot capabilities.
- Automated search engine and GUI to help user find feasible configuration.
- Automated mission design & planning to reduce cognitive load on user.

Future vision for RT Aspects

- Fully automatic command interface:
  - User issues high-level command for mixed robot team.
  - System searches for behaviors, configurations, and parameters.
  - RT Verification suite guarantees system behavior.
- Synergistic ties to playbook GUI, voice, gesture-based commanding.



For further information . . .

- Mobile Robot Laboratory Web site
  - http://www.cc.gatech.edu/ai/robot-lab/
  - http://www.cc.gatech.edu/ai/robot-lab/tmr
- PDF versions of pertinent papers
  - http://www.cc.gatech.edu/ai/robot-lab/tmr/archive.htm
- Videos
  - http://www.cc.gatech.edu/ai/robot-lab/tmr/videos.htm
- Contact information
    - Ron Arkin: arkin@cc.gatech.edu 404-894-8209
    - Tom Collins: tom.collins@gtri.gatech.edu 404-894-2509

Georgia Institute of Technology

Honeywell