

MissionLab:

Multiagent Robotics meets Visual Programming

Ronald C. Arkin
 Mobile Robot Laboratory
 Georgia Tech
 arkin@cc.gatech.edu


GT Mobile Robot Laboratory 1



Research Agenda



- Reactive social (multiagent) behaviors
- Design for communication
- Usable autonomous interfaces
- Mission specifications software using visual programming and reusable components
- Learning in hybrid architectures
- Targeted niche design
- Alteration of societal "personality"


GT Mobile Robot Laboratory 2




Supported by (recent past and present):

- National Science Foundation
 - Communication
 - Mobile Manipulation
 - Ecological Robotics
- DARPA
 - Tactical Mobile Robotics
 - Real-time Planning and Control/UGV Demol
 - Mobile Autonomous Robotics Software
 - Unmanned Ground Combat Vehicle
 - FCS-Communications SI&D
- Westinghouse/Savannah R&D
- C. S. Draper Lab
- SAIC
- USA Army Applied Aviation Directorate
- Honda R&D
- Sony Digital Creatures Lab







GT Mobile Robot Laboratory 3




GT Mobile Robot Lab Robots



GT-MRL


GT Mobile Robot Laboratory 4



AuRA: Autonomous Robot Architecture

- Hybrid deliberative/reactive system
- Arkin-1986
- Planning is viewed as configuration
- Initial A* planner integrated with schema-based controller
- Provides modularity, flexibility, and adaptability

GT Mobile Robot Laboratory 5



AuRA

Learning

Plan Recognition
User Profile

Spatial Learning

Opportunism

On-line Adaptation

User Input

User Intentions

Spatial Goals

Mission Alterations

Teleautonomy

Mission Planner

Spatial Reasoner

Plan Sequencer

Schema Controller

Motor

Perceptual

REPRESENTATION

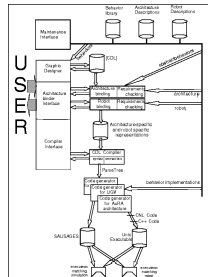
Hierarchical Component

Reactive Component

Actuation Sensing

GT Mobile Robot Laboratory 6

MissionLab



- Objective: To empower robot users to specify, evaluate, and execute missions and complex tasks

MissionLab



Problem Statement

- Constructing robot control configurations is *ad hoc* and tedious
- Configurations are difficult to re-target to new robots
- Component reuse is difficult yet needed at all levels of abstraction
- Support needed for evaluation of multi-robot configurations

MissionLab System



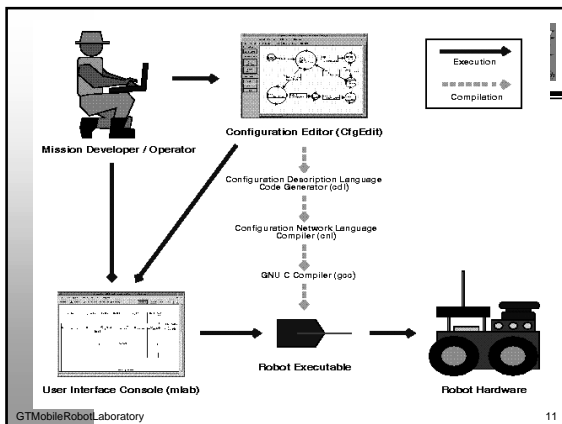
- Recursively build missions from reusable elements
 - Behaviors, behavioral assemblages (robots), teams of robots
- Generalizable to the robot control methods
 - Code generators for architectures
- Different levels of functionality for different user types
 - Novice users work with "useful" assemblages
 - Sophisticated users may develop new behaviors and assemblages



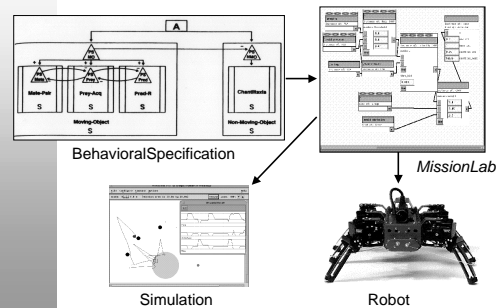
What MissionLab is



- missions specification tool
- a platform-independent robot interface
- a robot configuration tool
- a robot-to-robot communication mechanism
- missions simulation environment
- usability-tested human interface
- mission execution monitor
- a bridge between simulation and execution



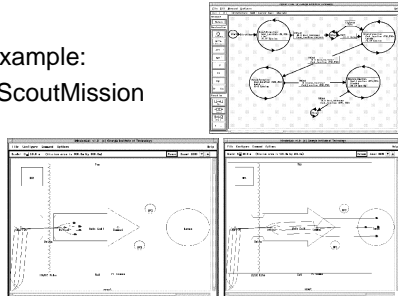
Development Process with MissionLab



MissionLab



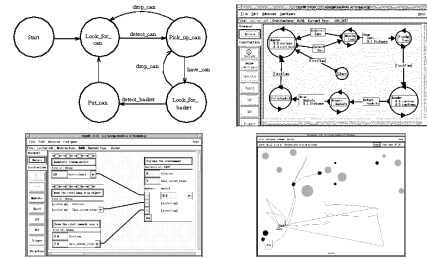
Example: ScoutMission



MissionLab



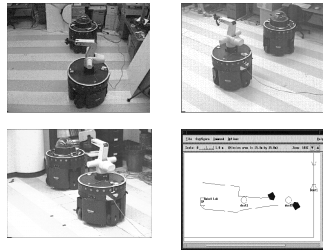
Example: Trashbot (AAAI Robot Competition)



MissionLab



EXAMPLE: LAB FORMATIONS

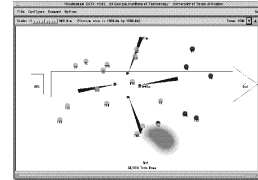


MissionLab

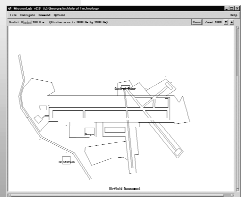


Example: Reconnaissance Mission

- Developed by University of Texas at Arlington using MissionLab as part of UGV Demoll
- Coordinated sensor pointing across formations



Airfield Assessment



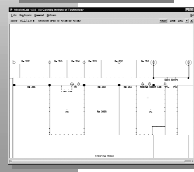
- Simulation test
- Includes deployment of "Throwbots"
- Initially developed for DARPA TMR
- Variation proposed as a multi-robot scenario for other DARPA program



Hospital Assessment



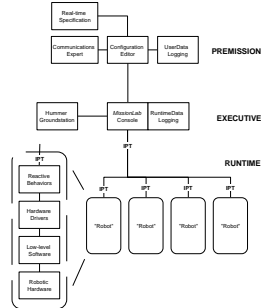
- Robot deployment and cross-country traversal information with differential GPS
- Stair-climbing robot (Urbie)
- Room-to-room search on upper floor (both simulations and hardware demonstrations exist)



Video



MissionLab Subsystems



Executive Subsystem



- The executive subsystem consists of the
- MissionLab console
 - faster-than-real-time simulator, and
 - runtime data logging components.

MissionLab Console



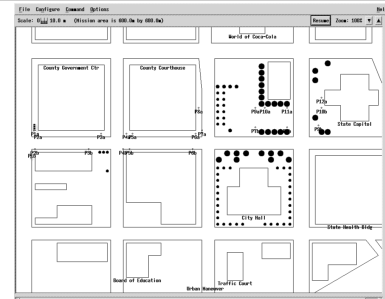
- Loads precompiled robot control programs and overlay description files
- Configures the display
- generating obstacles for simulations
- altering the scale of the display
- changing the virtual time for simulations
- scaling the size of the display (zooming)
- Provides a Command interface that permits interactive step-by-step command issuance by the operator using CMDL, a structured English language
- has the ability to execute, stop, pause, restart, rewind, single step, and abort missions during execution
- has the ability to use team teleautonomy by directing robot to particular regions of interest or by altering their societal personality().

MissionLab Console



- Provides display options
 - leave trails where the robot have been
 - highlight obstacles that affect the robot
 - show instantaneous directional reaction of robot to its environment
- The MissionLab console also provides a display that shows:
- The output of a simulated robotic mission that is run faster than real time
 - An operator mission display screen where robots in the field report back their position and relevant mission data that is shown on the lab display to provide situational awareness and context for higher level decisions regarding aborting, continuing, or biasing the mission in various ways

Console Display



Usability Data Logging

- Additional software is used to record user actions during premission planning.
- This includes data such as the number and type of keystrokes and mouse clicks, time to create certain objects, and other relevant data.
- These data are then used to interpret the skill by which a user is capable of achieving within the system, and after subsequent usability analysis, is used to refine the design interface itself.
- It is a support tool geared for formal usability studies.
- (More later)

Integrated Multi-layered learning

- CBR Wizardry
 - Guide the operator
- Probabilistic Planning
 - Manage complexity for the operator
- RL for Behavioral Assemblage Selection
 - Learn what works for the robot
- CBR for Behavior Transitions
 - Adapt to situations the robot can recognize
- Learning Momentum
 - Vary robot parameters in real time

THE LEARNING CONTINUUM:

Deliberative (premission)

Behavioral switching

Reactive (online adaptation)



Human-Robot Interaction

- Will robots be able to complete useful and meaningful tasks in conjunction with people?
- Will the results being produced in academic laboratories have impact in real-world everyday robotics?
- Will end-users of these systems be required to have a Ph.D. in robotics in order for them to be of any value?

Usability

A combination of:

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

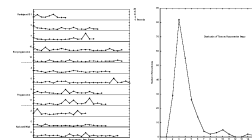
(Schneiderman 92)

Usability Requirements (after Hix93)

- 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

Formal Usability Studies

- Test in usability lab
- Subject pool of candidate end-users
- Used for both *MissionLab* and team teleautonomy
- Requires development of usability criteria and metrics



Usability Objectives

- To provide **effectivemethodologies** that evaluate the performance of multiagent robotics systems from an end-user perspective.
- To provide **methods and tools** in support of cognitive modeling of the interaction of users with multiagent robotics systems.
- To create **meaningful applications** for robotic teams, that can serve as prototypical tasks for the research community.

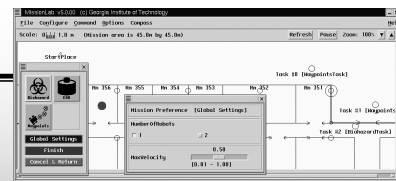
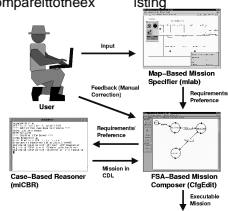
Experimental Procedures

- 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



Test: Wizard for *MissionLab* (in progress)

- Mission planning tool to assist creating mission plans
- Empowers less-skilled robot commander to develop sophisticated missions while avoiding complexities of directly building FSAs
- Current capabilities support 3 types of tasks, each with 9 aspects
- Usability studies are recurrently underway to compare it to the existing FSA-based mission planning technique



Output from map display

Task	WaypointsTask	Task	WaypointsTask	Task	BiohazardTask
StartYCoord	33.95	StartYCoord	41.29	StartYCoord	36.98
StartXCoord	25.26	StartXCoord	21.10	StartXCoord	18.65
Agressiveness	0.5	Agressiveness	0.5	Agressiveness	0.5

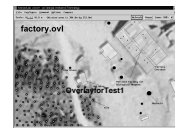
FSA generated by CBR



Case-Based Reasoning Mission Planning Wizard

Usability Study

- **Comparison of Two Subject Groups**
 - **Group A**: Using only traditional *MissionLab* functions (FSA, Waypoints, etc.)
 - **Group B**: Using the CBR mission planning wizard
- **Four Basic Tutorials for Both Groups**
 - Teach basic use of the *MissionLab* toolset
 - Teach how to create robot missions using the traditional FSA editor
- **One Extra Tutorial for Group B (Wizard people)**
 - Teach how to create robot missions using the CBR mission planning wizard
- **Two Tests (45 minutes each)**
 - **Test 1**: Indoor assessment mission for single robot
 - **Test 2**: Indoor assessment mission for two robots that must avoid the enemy entry in the building



Conclusions of Preliminary Usability Results



- The wizard **does reduce the time** required to create a suitable mission plan
- Wizard **increases accuracy** of mission plans created by users once they get used to the system
- Users find it **easier** to create a mission plan using the wizard than without
- Users seem to **require more wizard training** than they received during this study (one 20 minute tutorial). A help agent that can help teach the wizard would be a good addition to *MissionLab*.

Overarching Goals



To move robotics beyond the laboratories, a principled, structured design process is required which will:

- Support complex multi-robot missions and tasks
- Support portability, maintainability, and reuse of configurations
- Enable users to create and evaluate configurations without requiring programming skills
- Make hardware binding explicit to support delayed binding (least-commitment design) and retargeting

Robotics for the Masses

For more information



- <http://www.cc.gatech.edu/ai/robot-lab>
- arkin@cc.gatech.edu