LTL and Robot Integration

Abe Shultz, Dan Brooks, and Holly Yanco
University of Massachusetts Lowell
MURI Pipeline

- Parsing
- Semantics
- Pragmatics
- Robot
- World Model
- PAR
- LTL
iRobot ATRV Jr

- SICK LIDAR
- Hokuyo URG LIDAR
- Canon PTZ Camera
- DGPS
- Direct Perception Pan/Tilt Unit
- 2 Sony NTSC Cameras
- Indigo Systems IR Camera
- Logitech Wide Angle Camera
- 26 Unit Sonar Ring
- 3GHz Core2 Duo Logic Processing
Linear Temporal Logic Mission Planning

LTLMoP

Wednesday, November 3, 2010
Robot Software

- Modular Autonomous Robot Controller (MARC) created at our lab
- Hardware drivers and robot simulation provided by Player/Stage Project
Player/Stage

- Widely used hardware abstraction layer
- Player provides convenient interfaces to robot hardware
- Stage provides a simulation environment for rapid development
MARC Architecture

Interfaces
- LTL
- LTL Proxy
- UI Proxy
- Player Proxy

Processing
- Autonomy Code
- Pose Estimation
- Vision Processing
- openCV

Connections:
- LTL to LTL Proxy: JSON, UDP
- UI Proxy to Player Proxy: IPC
- Autonomy Code to Pose Estimation: IPC
- Vision Processing to openCV: Library Calls
- Player to UI Proxy: Library Calls
Internal Behavior and State Representation

- Robot has set of predefined behavior patterns
- Listens for commands and returns state information
- Everything gets recorded into a database
  - Sensor values
  - Decisions made dictating behavior
  - Current system state
- Reports subset of information back to LTLMoP upon request
Robot Behaviors

- Driving Commands
  - **GO** to [location]
  - **SEARCH** for [item, ...]
  - **FOLLOW** [target]
  - **STOP**

- Auxiliary Commands
  - **RADIO** when you find [item, ...]
  - **DEFUSE** [target]
Invocation of Robot Behaviors

• LTLMoP has actions which may be activated when certain conditions are met

• Activated actions are passed on to the robot for execution

• Successful execution is reported in updated state information received by LTLMoP
Integration with LTLMoP
JavaScript Object Notation (JSON)

• A flat text format for representing objects and data structures

• Bindings in many languages, including C++ (MARC) and Python (LTLMoP)
Example JSON Conversation

LTLMoP

- Commands
  - "action": {"go": "hall"}
  - "action": {"search": "bomb"}
  - "action": {"radio": "bomb"}

MARC

- State
  - "sensors": [], "action": {"go": "hall"}, "location": "classroom"
  - "sensors": [], "action": {}, "location": "hall"
  - "sensors": ["bomb"], "action": {"search": "bomb"}, "location": "hall"
Demonstration Environment

- Portion of the 3rd Floor of Olsen Hall
- 4 rooms
- Both robot and LTLMoP receive a map of the environment before the run

Wednesday, November 3, 2010
Demonstration
Scenario 1

• LTLMoP directs robot to a series of locations selected at random

• Demonstrates LTLMoP’s ability to command robot in a dynamic environment
Demo Video 1
“Explore the third floor”
Demonstration
Scenario 2

- LTLMoP directs robot to drive into a room and radio if it finds any hostages
- Robot determines if a person is a hostage based on the color of their shirt
Demo Video 2
“Enter the lab and radio if you find a hostage”
Demonstration
Scenario 3

• LTLMoP directs robot to visit rooms selected at random, looking for a “bomb”

• Once the robot reports it has found a bomb, LTLMoP sends further instructions telling the robot to stop

• The bomb is represented as a box with reflective markings
Demo Video 3

“Search the third floor for a bomb. Stop and radio when you find it.”
Future Work

- Continue work with LTLMoP to expand list of supported robot behaviors
- Autonomous robot collaboration and information sharing for multiple robot support