Mobile Robotics:
A Component Based Approach

ECE 750 – T4: Component Based Software Systems
Prof. L. Tahvildari

Yasmin Hovakeemian (20165205)
Adam Bertrand (20144032)
Outline

- Introduction
  - Robotics & Terminology
  - Sensors
- Proposed Robot & Mission
- Current Accomplishments & Future Work
- Simulation and Performance
- Related Research/Common frameworks
- References
Introduction to Robotics

- Multidisciplinary field that lends itself readily to component based software systems (CBSS)
  - Rapid development
  - Low cost
  - Modularity

- Robots generally have the following three aspects:
  - Set of sensors
  - Program of robot behavior (state estimation and control)
  - Set of actuators and effectors
Robotic Fundamentals

Sensors

- Sensor classification [5]:
  - Proprioceptive sensors (Internal):
    - Wheel encoders
  - Exteroceptive sensors (External):
    - GPS
    - SONAR/Laser
    - Bumpers

- Sensors can be chosen based on:
  - Environmental constraints
  - Objective/purpose (ie: vacuum robots)
Robotic Fundamentals

Robot Taxonomy

- **Autonomous** vs. Non-autonomous
- **Mobile** vs. Immobile/Stationary
- **Physical** vs. **Virtual** (‘Soft-Bot’ a.k.a Intelligent Agent)
Robotic Fundamentals

Control Architectures [6]

■ NASREM architecture:
  ○ Sense, model, plan, act

■ Subsumption architecture:
  ○ Layered control system

■ TCA architecture:
  ○ Task-specific modules with centralized control module

■ LAAS architecture:
  ○ Decisional (mission supervisor & task supervisor) and functional levels
Robotic Fundamentals

Control Architecture Characteristics [6]

- Hierarchical vs. Centralized Control
- Functional vs. Behavioral
- Reactive vs. Deliberate
Why a ‘Soft-Bot’?

- No hardware needed (cheaper)
- Can exploit advantages of component based software systems
  - Reusability & Modularity
  - Extendibility
- Time and Cost efficient
Proposed Project

Goal:

- Build an centralized autonomous mobile soft-bot using concepts from component based software systems to perform a specific task/mission
Definition of Mission/Task

- Autonomously navigate the proposed circuit as fast as possible.
- Avoid any obstacles in the way of the robot.
- Comply with any traffic signals on the course.
System Model

High Level Architecture

High Level System Design of Mobile Autonomous Robot
System Model

High Level Architecture

Simulation/GUI

Controller

Robot Motor
Robot Steering
GPS
SONAR
Laser
Encoders
State Estimator

Effectors

Sensors

July 13, 2009
### System Model

#### Component Level Architecture

- **RobotSteering**
  - currentHeading : double
  - desiredHeading : double
  - steer() : void

- **RobotGPS**
  - currentLocation : double
  - desiredLocation : double
  - getLocation() : double

- **RobotMotor**
  - translationalVelocity : double
  - rotationalVelocity : double
  - CcwRotation : boolean
  - isRotating : boolean
  - rotationalDirection : boolean
  - MAXtransVel : double
  - MAXrotationalVel : double
  - MAXaccelTrans : double
  - MAXaccelRotational : double
  - accelerateTrans() : void
  - accelerateRotational() : void

- **Robot**
  - isRunning : boolean
  - motor
  - steering
  - encoders
  - gpsModule
  - laserModule
  - sonarModule
  - stateEstimator
  - currentLocation : int
  - desiredLocation : int
  - robotEnvironment
  - start() : void
  - stop() : void
  - executeBehavior() : void

- **RobotStateEstimator**
  - currentState
  - estimateNextState() : void

- **RobotBumper**
  - collision : boolean
  - RobotEnvironment
  - checkForCollision() : boolean

- **Notes**
  - Does not show get/set and constructor operations
  - Shows Current Accomplishments
Mission

- Currently, soft-bot mission (task) is to start and stop a random walk
- In the future, mission of soft-bot can be adapted by simply changing the `executeBehavior()` method in the Robot class and environment.
Missions
State Diagrams

Random Walk in 4-walled environment
Missions

State Diagrams – Cont’d

Figure 8 track mission with traffic light and stop signs
Performance & Simulation

Simulation (Environment, Sensor Data)

Performance metrics:
- Time to complete mission
- Smoothness of trajectory/path
- Robustness to noise in sensor data
Analysis of System

Advantages:
- Reusability of code
- Extendibility (add components as needed)
- Behavior can be easily adapted
- No hardware costs

Disadvantages:
- Need to simulate hardware components (can be complex)
- Need sensor data to be realistic and coherent with one another
- May not necessarily match up with the real world
Current and Future work

- **Current Accomplishments:**
  - Soft-bot with random walk mobility

- **Future Work:**
  - Add additional sensor capability (eg: Sonar, Compass, etc)
  - Develop more complex environments
    - Obstacles
    - Figure 8 Track
    - Noise
  - GUI development (envision the robot in its environment with a mission)
Related Research

- Component-based robotics frameworks are available:
  - Maintain repository of components
  - Emphasize extensibility
  - Allow for development of single vehicle to distributed and/or decentralized robots

- Advantages of frameworks [7]:
  - Formal organization of components
  - Uniform communication mechanism between components
  - Component reusability
  - Starting point for amateurs

- Example Application
  - DARPA Grand Challenge
## Related Research

### Available Component Based Frameworks

<table>
<thead>
<tr>
<th></th>
<th>CARMEN</th>
<th>Webots</th>
<th>Player/Gazebo</th>
<th>Simbad</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Platform</strong></td>
<td>Linux</td>
<td>Windows/Linux/MAC</td>
<td>Windows/Linux/MAC</td>
<td>Windows/Linux/MAC</td>
</tr>
<tr>
<td><strong>Open Source</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Built-in visualization (GUI)</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Built-in Localization</strong></td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Build-in 3D vision</strong></td>
<td></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Sensor Simulator</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Ease of use</strong></td>
<td>Moderate</td>
<td>Easy-Moderate</td>
<td>Moderate</td>
<td>Easy</td>
</tr>
</tbody>
</table>
Summary

- Since Robotics is such a module-based field, it can take advantage of organizational aspects of component based software systems.
- A number of available software frameworks for component-based soft-bots.
- Current OO implementation of soft-bot capable of a random walk task.


Questions?