

# CS3773 Software Engineering

## Lecture 20 Software Architecture Case Study

### Mobile Robotics

A mobile robotics system is one that controls a manned or partially manned vehicle, such as a car, a submarine, or a space vehicle. Such systems are finding many new uses in areas such as space exploration, hazardous waste disposal, and underwater exploration. A robotics system must deal with external sensors and actuators, and they must respond in real time at rates commensurate with the activities of the system in its environment. In particular, the software functions of a mobile robot typically include acquiring input provided by its sensors, controlling the motion of its wheels and other moveable parts, and planning its future path.

2

UTSA CS3773

### Mobile Robotics Functional Requirements

- Typical software functions
  - Acquiring and interpreting input from sensors
  - Controlling the motion of all moving parts
  - Planning future activities
  - Responding to current difficulties

3

UTSA CS3773

### Mobile Robotics Complications

- Several factors complicate the tasks
  - Obstacles may block the robot's path
  - Sensor input may be imperfect or fail
  - Robot may run out of power
  - Mechanical limitations may restrict the accuracy with which it moves (movement may diverge from plans)
  - Robot may manipulate hazardous materials (water)
  - Unpredictable events may demand a rapid response

4

UTSA CS3773

## Mobile Robotics Design Constraints

### ➤ Evaluation criteria

- Accommodation of deliberate and reactive behavior: robot must coordinate actions to achieve assigned objectives with the reactions imposed by the environment
- Allowance for uncertainty: robot must function in the context of incomplete, unreliable and contradictory information
- Accounting of dangers in the robot's operations and its environment: relating to fault tolerance, safety and performance, problems like reduced power supply, unexpectedly open doors, etc., should not lead to disaster
- Flexibility: support for experimentation and reconfiguration

5

UTSA CS3773

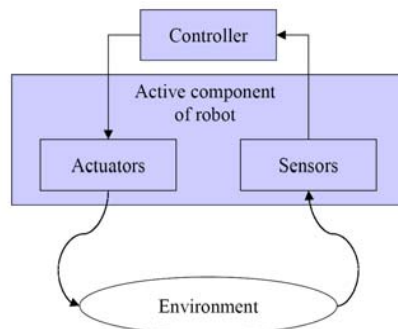
## Mobile Robotics Architecture Designs

- Closed loop control architecture
- Layered architecture
- Repository/Blackboard architecture
- Implicit invocation architecture

6

UTSA CS3773

## Control Loop Architecture Style



7

CS3773

## Control Loop Architecture Style Evaluation

- Accommodation of deliberate and reactive behavior: robot must coordinate actions to achieve assigned objectives with the reactions imposed by the environment

Simplicity: problem in more unpredictable environments since there is basic assumption that changes in environment are continuous and require continuous reactions. Basic changes in behavior may be needed when confronted with disparate discrete events.

8

UTSA CS3773

## Control Loop Architecture Style Evaluation

- Allowance for uncertainty: robot must function in the context of incomplete, unreliable and contradictory information

Uncertainty is resolved by reducing unknowns through iteration: problem if more subtle steps are needed.

9

UTSA CS3773

## Control Loop Architecture Style Evaluation

- Accounting of dangers in the robot's operations and its environment: relating to fault tolerance, safety and performance, problems like reduced power supply, unexpectedly open doors, etc., should not lead to disaster

Fault tolerance and safety are enhanced by the simplicity of the architecture.

10

UTSA CS3773

## Control Loop Architecture Style Evaluation

- Flexibility: support for experimentation and reconfiguration

Major components (supervisor, sensors, motors) can be easily replaced; more refined tuning must take place inside the modules.

11

UTSA CS3773

## Control Loop Architecture Style Overall Evaluation

- Appropriate for simple robotic systems that must handle only a small number of external events and whose tasks do not require complex decompositions

12

UTSA CS3773

## Layered Architecture Style

1. Supervisor
2. Global planning
3. Control
4. Navigation
5. Real-world modeling
6. Sensor integration
7. Sensor interpretation
8. Robot control

13

UTSA CS3773

## Layered Architecture Style Evaluation

- Accommodation of deliberate and reactive behavior: robot must coordinate actions to achieve assigned objectives with the reactions imposed by the environment

Nicely organizes components needed to coordinate operation. However, does not fit the actual data and control-flow patterns. Information exchange is less straightforward: exceptional events may force direct communication between levels 2 and 8, for example. Also, there are really two abstraction hierarchies that actually exist: a data hierarchy and a control hierarchy.

14

UTSA CS3773

## Layered Architecture Style Evaluation

- Allowance for uncertainty: robot must function in the context of incomplete, unreliable and contradictory information

Existence of abstraction layers nicely addresses need for managing uncertainty: things get more certain the higher one gets.

15

UTSA CS3773

## Layered Architecture Style Evaluation

- Accounting of dangers in the robot's operations and its environment: relating to fault tolerance, safety and performance, problems like reduced power supply, unexpectedly open doors, etc., should not lead to disaster

Fault tolerance and passive safety are also served by abstraction mechanism. Performance and active safety issues may force the communication pattern to be short-circuited.

16

UTSA CS3773

## Layered Architecture Style Evaluation

- Flexibility: support for experimentation and reconfiguration

Interlayer dependencies are an obstacle to easy replacement and addition of components.

## Layered Architecture Style Overall Evaluation

- Nice, high-level view of robot control, but breaks down as an implementation view since the communication patterns are not likely to follow the orderly scheme implied by the architecture.