Lego Robots and Software Design

CITS1001 extension notes
Rachel Cardell-Oliver
Lecture Overview

1. Designing NXT Robots
2. NXT Hardware
3. Sensors and Actuator API
4. The Behaviour Interface
Software Design

is a problem-solving process whose objective is to find and describe a way

- To implement the system’s *functional requirements*...
- While respecting the constraints imposed by the *quality, platform and process requirements*, including the budget
- And while adhering to general principles of *good quality*
Lecture Overview

1. Designing NXT Robots
2. NXT Hardware
3. Lejos Java API
4. The Behaviour Interface
Lego NXT: by Ro Mathew

- From the Czech word: robota

- Slavery, Drugery, Servitude – Forced Labour
Robot **Sensors:**

- Provide information for the robot to examine its environment.
Robot Sound:
Robot Touch:
Robot Vision - Colour:

This is what your eyes see

This is what your robot will see, using the light sensor.
Robot Vision – Distance:
Robot Peripherals:

- Allows the robot to change its environment.
Robot Peripherals:

screen

speakers
Robot Controller:

- Acts as the brain for the robot: Coordinates Devices
ROBOT CONTROLLER

Connect to our computer
Store programs and data
Connect to our sensors and motors
Retrieve information and provide power
Run programs and display information
Robot Motion System:

- Acts as the muscles for the robot allows physical movement
Robot Motion:
The more sensors, the better the robot is able to interact with the environment.

- Motors
- Gears and Axles
Robot System:

- Incorporates sensors, peripherals, motion and power systems
Robot System:
For Building Instructions see CITS1001 extensions page

Components
- Basic NXT Driving Base
- Motor Module
- Sound Sensor Module
- Touch Sensor Module
- Ultrasonic Sensor Module

Models
- Scorpion
- TriBot
- Alpha Rex humanoid
Lego RCX
Lecture Overview

1. Designing NXT Robots
2. NXT Hardware
3. Lejos Java API
4. The Behaviour Interface
import lejos.nxt.*;
class Cockroach {
    public static void main(String [] args) {
        LightSensor ls = new LightSensor(SensorPort.S2);
        Motor.B.forward();
        Motor.C.forward();
        LCD.drawString("Too much light",3,4);
        LCD.refresh();
        while (ls.readValue() > 55) {
            //keep moving forward until dark is found
        }
        LCD.drawString("That's better",3,4);
        LCD.refresh();
        Motor.B.stop();
        Motor.C.stop();
    }
}
Lecture Overview

1. Designing NXT Robots
2. NXT Hardware
3. Lejos Java API
4. **Lejos Behaviour Interface**
   *(from the book Max Lego NXT, Chapter 18)*
Subsumption
package lejos.subsumption;

public interface Behavior {

    public boolean takeControl();
    // Trigger condition for invoking this behaviour

    public void action();
    // Start a behaviour (eg. move forward)
    // Actions must return quickly (so that Arbitrator can continue checking takeControl)

    public void suppress();
    // Terminate this behaviour (eg. Stop a motor)
    // Also update any data if needed
}
Arbitrator

public Arbitrator(Behaviour [] behaviours);

- Create an arbitrator with an array of behaviours: highest array index has highest priority

public void start()

- Starts the arbitration system:
  - call takeControl() for each behaviour starting with the highest priority behaviour, until true
  - Execute the suppress() method of the current (lower priority) behaviour then
  - Execute the action() method of the chosen behaviour
Learn More …

- Lejos project home page
- http://lejos.sourceforge.net/
  - Lejos NXT API
  - Sample Programs
    - See BumperCar example for behaviours
  - Lego RCX tutorial (for previous generation hardware, but many parts still relevant)