Lego Robots and Software Design

CITS1220 Software Engineering
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:
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 CITS12220 Resources web 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

see handout: Max Lego NXT, Chapter 18
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

```java
public Arbitrator(Behaviour [] behaviours);
```

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

```java
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
package lejos.subsumption;

public class DriFord implements Behavior {

    public boolean takeControl() {
        return true;
    }

    public void action();
    {
        Motor.A.forward();
        Motor.C.forward();
    }

    public void suppress();
    {
        Motor.A.stop();
        Motor.C.stop();
    }
}
package lejos.subsumption;

public class HitWall implements Behavior {
    public TouchSensor touch = new TouchSensor(SensorPort.S!);

    public boolean takeControl() {
        return touch.isPressed();
    }

    public void suppress() {
        Motor.A.stop();
        Motor.C.stop();
    }
}
public void action();
{
    Motor.A.backward();
    Motor.C.backward();
    try{Thread.sleep(1000);}catch(Exception e) {}
    // Rotate by causing only one wheel to stop:
    Motor.A.stop();
    try{Thread.sleep(300);}catch(Exception e) {}
    Motor.C.stop();
}
```java
import lejos.subsumption.*;
public class BumperCar {
    public static void main(String[] args) {
        Behavior b1 = new DriFord();
        Behavior b2 = new HitWall();
        Behavior[] bArray = {b1, b2};
        Arbitrator arby = new Arbitrator(bArray);
        arby.start();
    }
}
```
Subsumption
Learn More …

- CITS1220 Resources web page
  - Lejos NXT API
  - Sample Programs
    - See BumperCar example for behaviours
  - Lego RCX tutorial (for previous generation hardware, but many parts still relevant)

- http://lejos.sourceforge.net/