Java is a trademark of Sun Microsystems, Inc.

Java

JavaOne℠

Fun with Java™ Technology on Lego® Mindstorms®

Roger Glassey
University of California, Berkeley

Andy Shaw
Sun Microsystems

LEJOS
Java for LEGO Mindstorms

Lego Mindstorms – A little history

- Originally launched 1998
  - The Lego Mindstorms Robot Invention System (RCX “Brick”)
  - Simple visual programming system
  - Reverse engineered
- Major update 2006
  - Lego Mindstorms NXT
  - Open source hardware & firmware
  - Lego Mindstorms NXT V2.0 Aug 2009

Lego NXT

- Atmel Arm7TDMI + Atmel ATMega 48
  - 64Kb Ram
  - 256Kb Flash
- 3 Motor ports, PWM + Tachometers
- 4 Sensor ports
  - Analogue/Digital I2C
  - Standard: Ultrasonic, Touch, Light, Sound
  - Additional: Lots!
- Bluetooth, USB, RS485
- LCD 100x64 display/PDM sound output

Java on the NXT - leJOS

- Evolved from TinyVM/leJOS for the RCX: 10Kb JVM!
- Firmware 53Kb
  - Threads: pre-emptive, priority based, 2ms time slice
  - Interpreter: Fast table based dispatch
  - GC: concurrent, very low pause time
  - Native methods: I2C, Bluetooth, USB, Motor, LCD, Sound, Flash, RS485
- Language
  - Most 1.5+ features iterators, generics, enums, nested classes, etc.
  - longs, floats, doubles, arrays, strings etc.

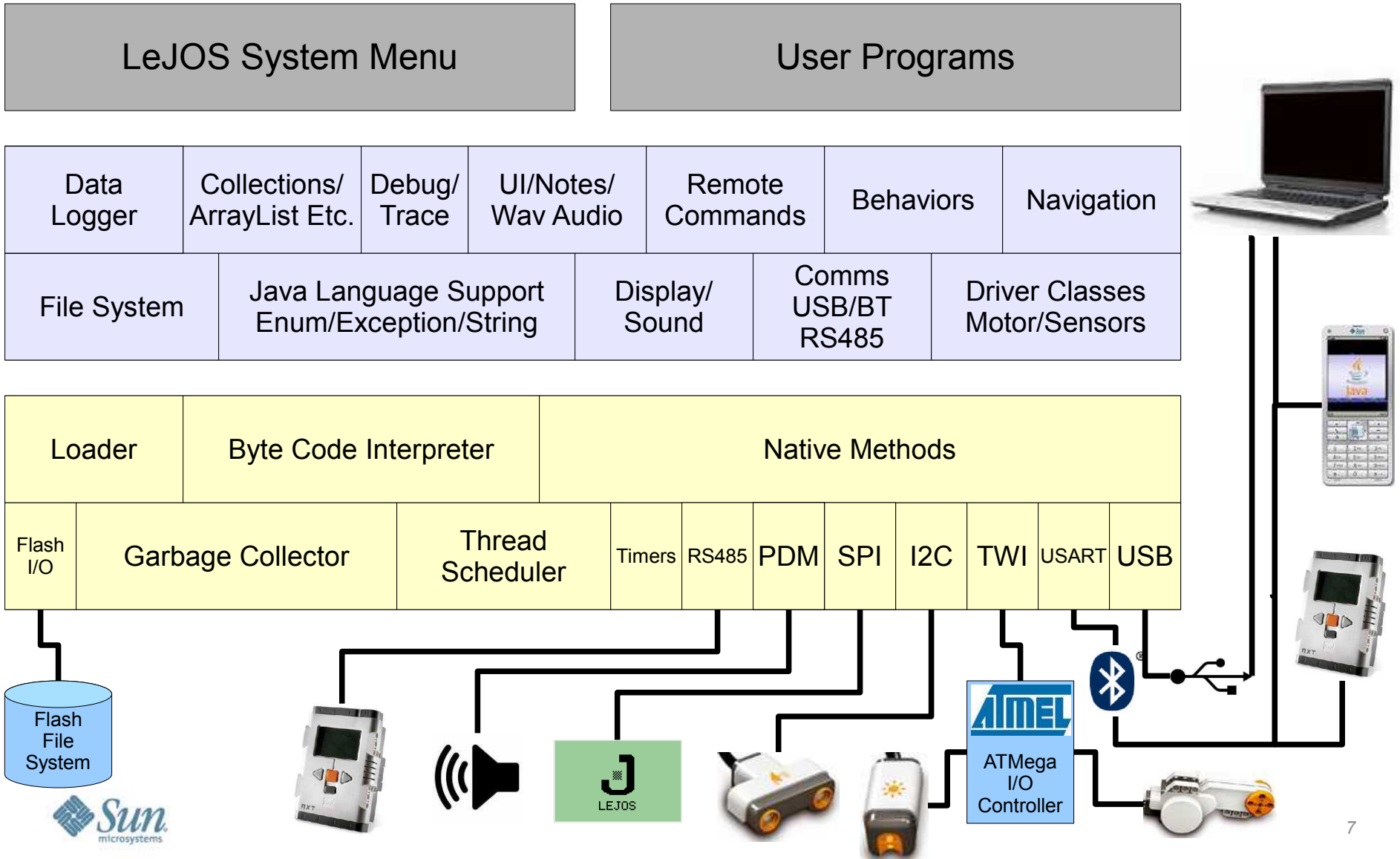
Java on the NXT - leJOS NXJ

- Classes
  - classes.jar: 300+ classes
  - Support for more than 36 sensor types.
  - Unified communications for Bluetooth/USB/RS485
  - Sophisticated motor control
  - Robotics: Subsumption, navigation, localization
  - Flash based file system
  - Data logging
  - Remote control - Lego command protocol
  - Remote console, basic debugging

Java on the NXT – Host tools/classes

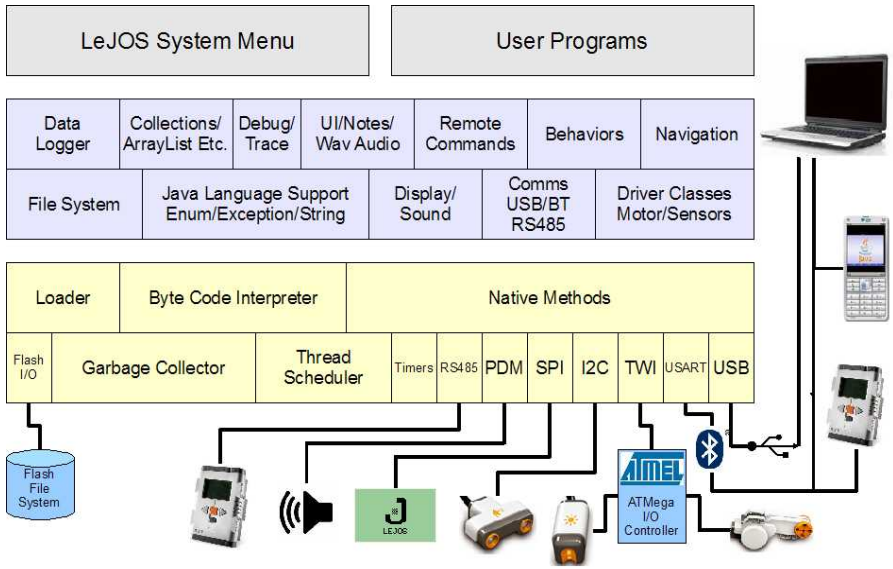
- Support for Windows/Linux/Mac
- Tools
  - Firmware upload, program upload, linker
  - Browsers: files, data logging, remote console
- Plugins for NetBeans/Eclipse
- Classes
  - Communications: Bluetooth, USB
  - Remote control: Lego Control Protocol
  - Over 100 classes
- Mobile devices, Java ME

LeJOS Architecture

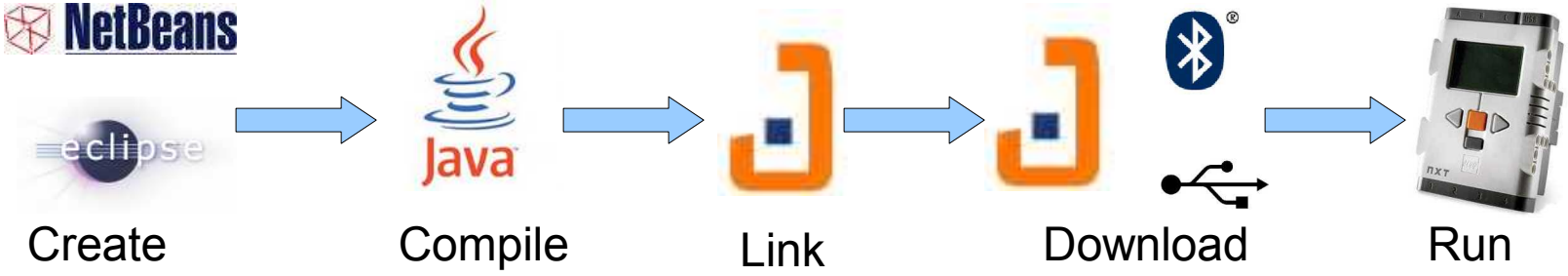


Architecture

- Firmware v Java
  - Java whenever possible
  - Easier to change
  - More team members
- Enabled by firmware
  - Priority based threading
  - GC low pause time (< 1ms)
  - Allows many real time functions in Java
    - Motor control PID tasks
    - Sensor processing
    - Communications



Building applications



- Standard tools to create and compile
  - NetBeans, Eclipse, vi, emacs!
  - Javac
- Linker
  - No class loader
  - Combines application classes with classes.jar
  - Eliminates unused code (50%-60% reduction)
  - Creates image that can be executed from flash

Alpha Rex

- The Mindstorms “poster robot”
- Rex is “helping out” by greeting delegates at JavaOne
  - Waits looking around
  - Pays attention when approached
  - Walks forwards to meet
  - Reads the badge
  - Greets the delegate

Demonstration

- Let's see Rex in action...
  - View a video by clicking here

Main Components

Head

- Ultrasonic sensor “eyes”: Distance

Body

- Single motor moves head/arms
- Right Arm – RFID Sensor
- Left Arm – Light Sensor “torch”

Legs

- Hip Motor – rocks side to side
- Leg Motor – moves legs

Additional Components

On/Off button

- Touch sensor – starts/stops Rex

Sound

- Plays wav files - greeting

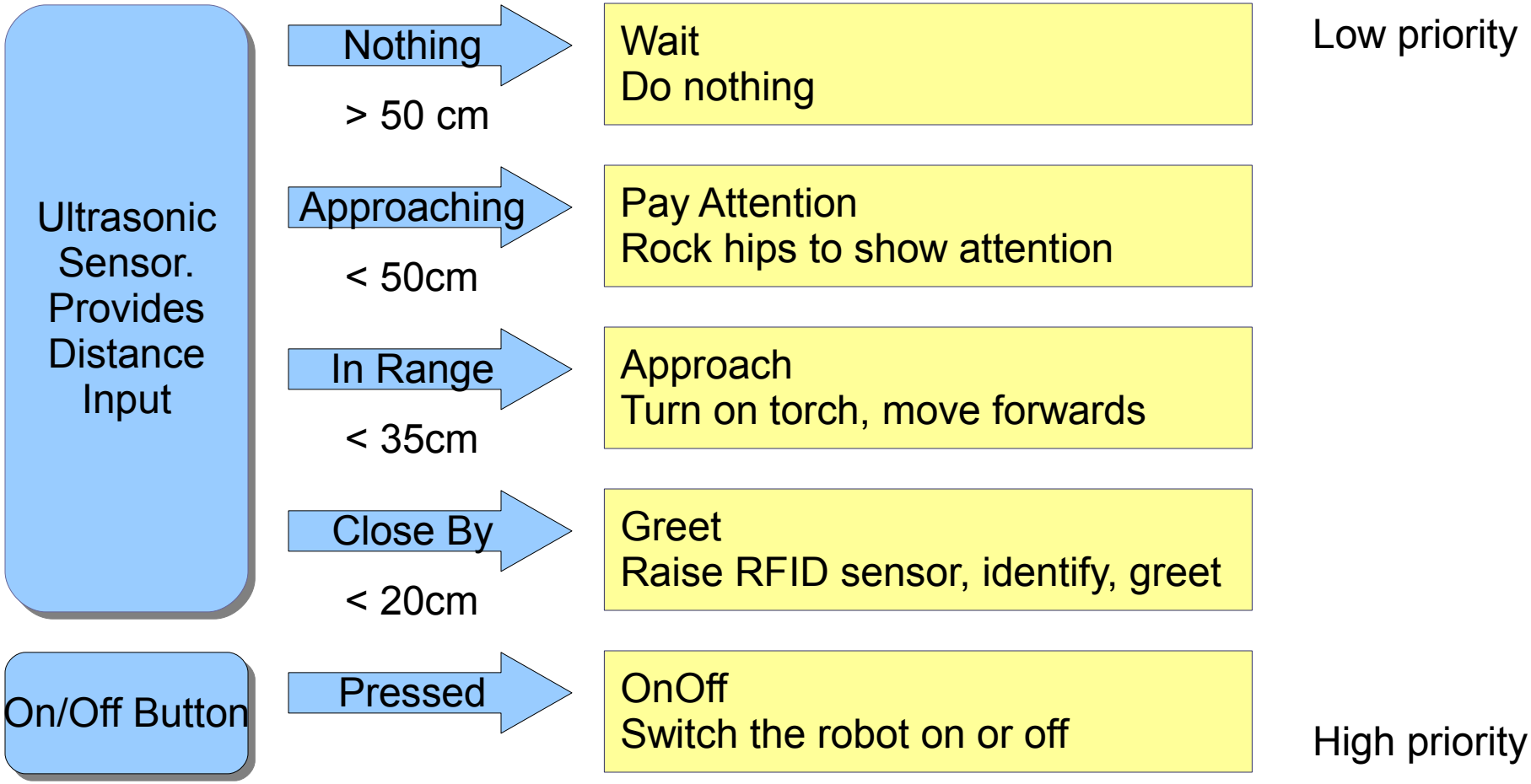
LCD

- Displays status

Bluetooth

- Connects to remote console
  - Displays detailed status
  - Shows LCD screen

Behavior model



Let's look a little deeper

- So what is actually happening...
  - Look at the Remote Console
    - See a video by clicking here
  - Look at the code
    - The code is available here

LejosNXJ

in Engineering Education

Roger Glassey
Professor, IE & OR

IEOR140: Introduction to mobile robots

- Course Objectives: Develop engineering skills
  - Analysis and design of moderately complex systems
    - interactions between the hardware and software design.
  - Effective project team membership
  - Report writing

Course structure

- Teams of 2 or 3
- Weekly project
- Translate performance specifications into hardware and software design.
- Project based learning plus interactive web based individual Java instruction.
- Class size: 20 – 30
- 3rd or 4th year engineering students

Projects

- Start simply
- Build a robot to trace a simple shape
- Hardware: robot with 2 wheels, independently controlled
- Square:
  - Repeat 4 times:
  - travel in straight line
  - 90 degree turn

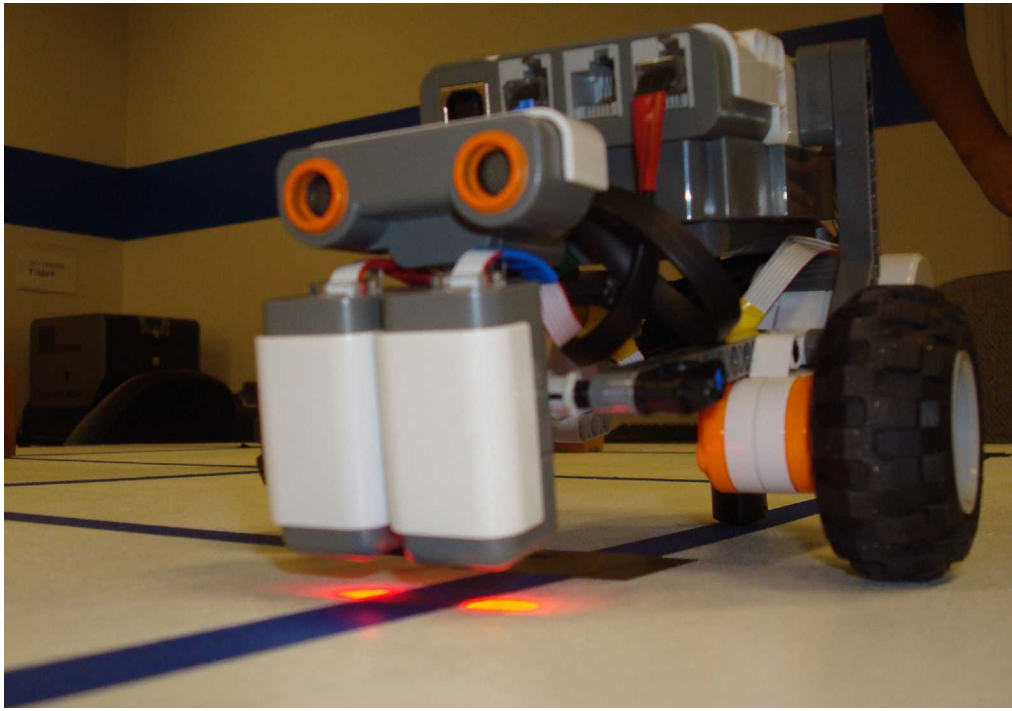

```
import lejos.nxt.*;

public class SquareBot
{
  private void square(int length)
  {
    for (int i = 0; i < 4; i++)
    {
      Motor.A.rotate(degPerDistance * length, true);
      Motor.C.rotate(degPerDistance * length);
      Motor.A.rotate(-90 * turnRatio, true);
      Motor.C.rotate(90 * turnRatio);
    }
  }
```

Demonstration

- Let's see the the robot in action
  - Sorry we don't have a video of this part of the demo. But I'm sure you can work it out!

Grid Navigation

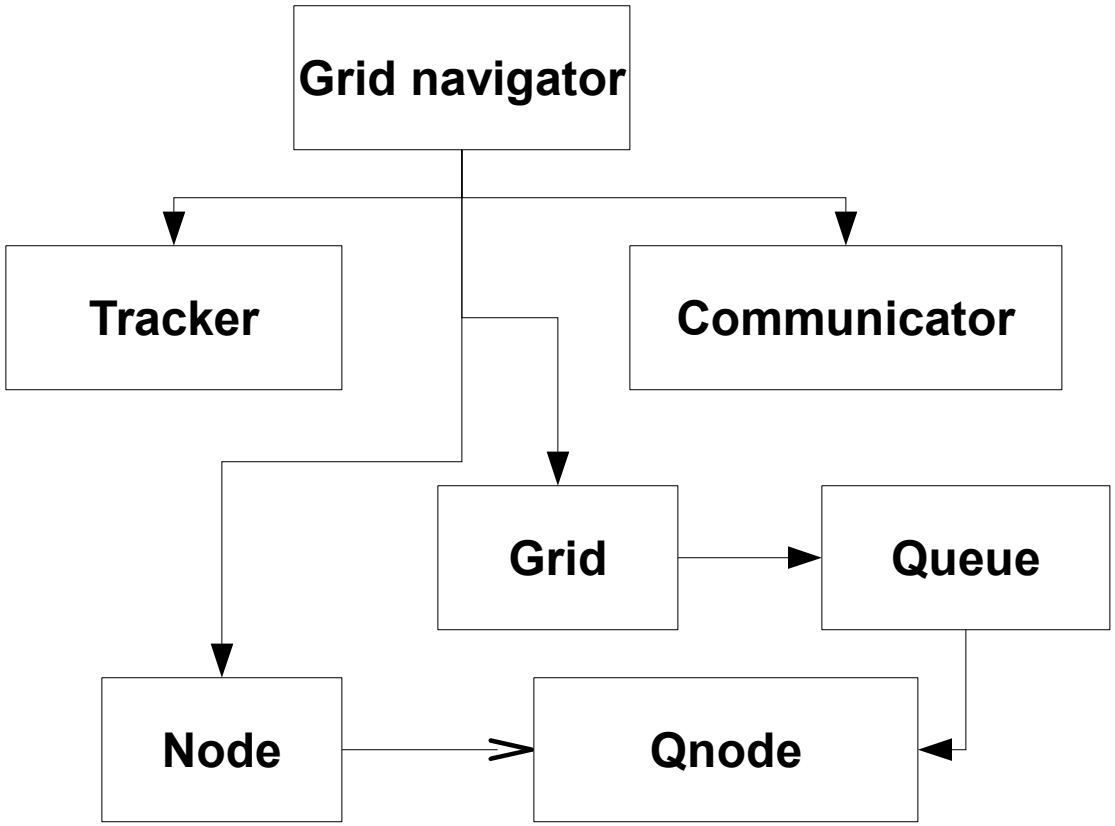
- Prototype of materials handling robot
- Navigate a rectangular grid
  - Line follower
- Detect obstacles - adaptive path planning
- Remote control and mapping via Bluetooth
- Sensors:
  - 2 Light sensors
  - Ultrasonic sensor



Grid navigation tasks

- Accept destination
- Go to destination
  1. Detect obstacle
  2. Decide on direction of turn - Dijkstra's algorithm
  3. Execute turn
  4. Travel to next intersection
  5. Repeat 1 – 4 until at destination

Grid Navigation
Class Dependencies



Demonstration

- Let's see the Grid Navigator in action
  - You can view a video by clicking here

Follow me + Voice control

- Simple co-operation between robots.
- First robot has sensors
  - Obstacle detection
  - Audio Input
- Linked to second via Bluetooth
- Second robot follows commands from first

Demonstration

- Let's see the two robots in action
  - See a video by clicking here

Conclusions

- Decomposing robot performance specifications into tasks and sub-tasks can result in a clean, modular design of classes and methods.
- Many students report that this course, while requiring more work than most, is also more fun than most.

More information

- http://lejos.sourceforge.net/
- http://mindstorms.lego.com/

JavaOne℠

Thank You