Smart Java^{TM} Technology Lego^{TM} Robots Invade the University BOF 3044

Autonomous Embedded Java Robots in University of Utah CE/CS Senior Projects

Ben Holt
Bruce Boyes

University of Utah
and Systronix Inc
www.jcx.systronix.com

THE UNIVERSITY OF UTAH

SYSTRONIX
Embedded Java Spoken Here

JCX at the University of Utah

A new hardware and software paradigm for teaching at the University level

Using real-time, native-execution Java and the JCX system architecture in the CE/CS 4710 Senior Project class.

JCX in Univ of Utah Senior Projects

CE4710 class overview

Raw materials

Two phases

New topics in 2003

Smooth sailing

Land mines

Summary

CE4710 Class Overview

Past History

- Required of all senior CE students
  - Necessary for graduation
  - Senior “Capstone” project class
  - 3 credits, taught in fall semester
- Prior to 2001 was using 6811 Handyboards
  - This is fine technology but hasn't the state of the art advanced in 15-20 years??
  - Most of class could not complete in fall semester, drew incompletes and finished in spring. This is why it is taught in the fall...
  - No machine shop facilities for CE4710 students, yet students were “encouraged” to build hardware
- Vital to new ABET accreditation

CE4710 Class Overview

Big changes in 2001 – 2002 - 2003

- Embedded Java
  - “JCX” hardware architecture
  - Standardized high level language
  - Rich, standardized I/O
  - Decent development tools
- Lego Technics for mechanical chassis
  - “What? I'm a college senior using 'Legos'??”
  - Lab administrator and some faculty were highly skeptical, not to mention the students
- You *will* complete in one semester
  - Tight management, milestones, status meetings

JCX in Univ of Utah Senior Projects

CE4710 class overview

Raw materials

Two phases

New topics in 2003

Smooth sailing

Land mines

Summary

Raw Materials

Facilities and students

- New senior hardware lab
  - Access 24/7 with badge reader
  - This year:
    - Newly remodeled lab, ready first day
    - Computers ready to use
    - No Olympics or other distractions
- 8 teams of two to four students
  - Each with own workstation
  - Local subnet, each team has admin rights
  - Each with own JCX and other hardware

Raw Materials - Old

20-year old technology was good 20 years ago...

- 68HC11 Handyboard
  - 32 Kbytes NVRAM
  - Four motor outputs
  - 16 sensor inputs, 8-bit sampling
  - 16 x 2 LCD
  - TV/VCR IR Remote
  - $300 for the Handyboard w/battery
- C and assembly
  - No IDE, doc tools, build control
  - proprietary/arbitrary libraries
  - Students asked “why are we learning obsolete things?”

Raw Materials - New

Software

- Java
  - Javaxcomm serial I/O
  - Javadoc with options – tags, packages, names
  - Code conventions
  - Eclipse IDE with JDK 1.4.03
  - Ajile runtime – J2ME/CLDC 1.0
- JCX API
- Ant
  - Project build automation
  - Uses XML build dependency files

JCX

Architecture - Signals

- SPI is the main expansion interface
  - 4 to 6 Mbits/sec on JStamp and JStik
  - Typically master/slave, could be P2P
- Asynch serial
  - Serial FIFOs, hardware UARTs x 2, 115 kbaud
  - Standard javaxcomm package for all
  - Up to 14 UARTs on JStik with JSQS
- 1-Wire
- I2C
- HSIO on JStik
- Ethernet

JCX

Architecture – Tagging

- Tagging memory – why?
  - One code base for N systems (swarms)
  - Multiple instances of same hardware board
    - Possible 20 motors or 32 sensor inputs per system
    - 120 DIO or 32 Analog ins and Analog outs
    - How do you maintain sanity
    - Plug and play
    - Easy board swapping for industrial use
  - Auto configuration of hardware to controller
  - Enumeration of all available devices
  - Label how I/O instances are being used
  - Runtime class binding with Class.forName
  - Versioning information

JCX

Architecture – Tagging

- Tagging - what
  - eeprom in SPI address space
  - XML data, parser in API
  - Sacred and user spaces
  - Generally a *device*, not a *board* perspective
    - You instantiate a *motor*, not a motor board
  - Makes SPI address use and conflicts obvious

JCX

System Hardware

- JCX System
  - Rev 3, almost indestructible
  - JStamp on JSimm carrier board
  - Six-slot backplane
  - Quad motor driver
  - Octal sensor inputs
  - RF Modem
  - Prototyping board

JCX Photos

Photos

- JSimm.JStamp
- Six-slot backplane
- Quad motor driver
- Octal sensor inputs
- RF Modem
- Prototyping board

Using JCX

Two basic ways

- Instantiation via Enumeration
- Direct Addressing

I/O Devices Used in 4710

Hardware

- Sensors
  - Lego touch, light, rotation
  - CMUcam color vision with primitives
  - Sonar modules SRF04
- Motors
  - Lego DC motors with PWM control in JCX
  - R/C servo motors
  - Heavy duty gearhead motors
- UI
  - Amulet ¼ VGA with touchscreen
  - Smart local controller and page memory
  - Uses HTML and OO paradigm

JCX in Univ of Utah Senior Projects

CE4710 class overview

Raw materials

Two phases

New topics in 2003

Smooth sailing

Land mines

Summary

Two phases

Get a toolkit then apply it

- Lecture phase
  - Individual assignments, quizzes
  - Create a toolkit for the project phase
  - Create project definition, budget, plan, IP
  - Fifteen lectures, 2 per week, Aug 21 - Oct 23
- Project phase
  - Milestones for each week
  - Weekly status meetings
  - Final presentation
    - 10 minutes per team
    - “Justify your team's existence to management”
  - Six weeks, Oct 27 – Dec 05

JCX in Univ of Utah Senior Projects

CE4710 class overview

Raw materials

Two phases

New topics this year

Smooth sailing

Land mines

Summary

Special Topics

New in 2003

- IP issues
  - Visit from practicing attorney
- Threading, esp Periodic Threads
  - One team persisted then abandoned in favor of simpler cooperative threading
- RF Modems
  - True peer to peer link layer, media layer completely abstracted away
  - Same javaxcomm stream as LCD GUI
  - One team used this in a sonar mapping project
  - Data streamed back to PC for display

Special Topics

Approach to rapid development

- Top down spec
  - Functional partition
  - Simple state diagrams & flowcharts
- Use efficient tools
  - Java, eclipse, Ant, javadoc, team website, email
- XP Techniques
  - Team programming
  - Documentation first! Use of javadoc
  - Start simple, iterate, test, save versions as you go
- Frequent reviews
  - Weekly team/prof/TA meetings, part of grade

JCX in Univ of Utah Senior Projects

CE4710 class overview

Raw materials

Two phases

New topics this year

Smooth sailing & land mines

Summary

Outcome of 2003 Class

Successes

- Lego mechanical bits for fast prototyping
  - Most faculty were impressed
  - Most students quickly embraced them
- Java
  - After some learning curve most students were enthusiastic. Some had made it through school with no Java training and struggled a bit.
  - One team chose C and AVR assembly code for their project

Outcome of 2003 Class

Some landmines/speed bumps

- First order reasonableness test
  - Do this *before* cutting metal
- Task partitioning & team coordination
  - Rampant optimism
  - Reluctance to change plans once made, assuming you *have* a good contingency plan
- Documentation & communication
  - Yes, I *really* would rather have well-documented broken code than undocumented working code
- Threading tough to grasp, esp PT

Outcome of 2003 Class

More landmines

- Debugging
  - It's hard to teach a logical, hierarchical approach to hard- and soft- ware triage
  - Best learned on the job, and most students haven't had much of one yet
- Serial communication protocol
  - Example: ascii-encoded hex data to/from LCD
  - Students had to build a self-correcting parser and packet-handling code.
  - This proved much more problematic than expected, even given good packet documentation, examples, and a sniffing tool.

JCX in Univ of Utah Senior Projects

CE4710 class overview

Raw materials

Two phases

New topics this year

Smooth sailing & land mines

Summary

Summary

- No incompletes for third year
  - Every team completed the required course elements in one semester
  - Though not all teams reached all their goals
  - Learning occurs in striving for goals not met
- Real projects are hard
  - The difference between theory and practice, while small in theory, is large in practice.
- Basic engineering principles still count
  - We observe the normal distribution of performance
- One semester may not be enough
  - 2004-2005 year is using two semesters

JCX in Univ of Utah Senior Projects

CE4710 class overview

Raw materials

Two phases

New topics this year

Smooth sailing & land mines

Summary

Case history

Case History – Ben Holt

- Intro
- Team
- Project/Rules
- What went well
- Minor disasters
- What we learned

Future directions?

- One semester implies high-level, abstract tools
  - Is this really good? Where do they get the low-level view? “Brakes don't stop the car, tires do”
- Better threading tools are needed
  - Threads themselves are a Java tar pit, much is “implementation dependent”
- Better debugging tools
- We're still in the programming Stone Age, how do we get out?

Videos

Student Team Projects

For More Information

- Technologies
  - http://www.jcx.systronix.com
  - http://www.cs.utah.edu/classes/cs4710/
  - http://www.amulettechnologies.com/
  - http://www.java.sun.com
- Techniques
  - *Debugging: The Nine Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems,* David J. Agans, 2002
  - *The Java Language Specification,* http://java.sun.com/docs/books/jls/second_edition/html/j.title.doc.html
  - http://www.practicalembeddedjava.com/

Autonomous Embedded Java Robots in University of Utah CE/CS Senior Projects

Questions?

Bruce Boyes

University of Utah
and Systronix Inc
www.jcx.systronix.com

THE UNIVERSITY OF UTAH

SYSTRONIX
Embedded Java Spoken Here