Teaching Distributed Systems to Undergraduates: An Experience Report

Gregory M. Kapfhammer
(Geoffrey Arnold and Brian Zorman)

Sixth Annual Jini Community Meeting
Boston • June 17-20, 2002

Allegheny COLLEGE
Department of Computer Science

Presentation Outline

- Introduction
- Instructor to Students: "Building distributed systems is hard!"
- Course and Performance Objectives
- A Sampling of Covered Material
- Homework Assignments: Read the literature!
- Laboratory Assignments: Jini and JavaSpaces!
- "Lessons Learned" from the Laboratories
- A Survey of Submitted Final Projects
- "The Next Time Around": Teaching Distributed Systems Again!
- Concluding Remarks

Allegheny College
Department of Computer Science

Introduction

- He turned to the flyleaf of the geography and read what he had written there: himself, his name, and where he was. *Stephen Dedalus, Class of Elements, Clongowes Wood College, Sallins, Country Kildare, Ireland, Europe, the World, the Universe.*

  James Joyce, *A Portrait of the Artist as a Young Man*

- He turned to the PowerPoint presentation and read what he had written there: himself, his name, and where he was. *Gregory M. Kapfhammer, Department of Computer Science, Allegheny College, software testing and engineering community, Jini Community, Meadville, Pennsylvania, United States, the World, the Universe.*

Allegheny College
Department of Computer Science

The First Day of Class

" … When I think of formal scientific method an image comes to mind of an enormous juggernaut, a huge bulldozer -- slow, tedious, lumbering, laborious, but invincible. It takes twice as long, five times as long, maybe a dozen times as long as informal mechanic's techniques, but you know in the end you're going to *get* it. … When you've hit a really tough one, tried everything, racked your brain and nothing works, you know that this time Nature has really decided to be difficult, you say, 'Okay, Nature, that's the end of the *nice* guy,' and you crank up the formal scientific method …"

Robert M. Pirsig, *Zen and the Art of Motorcycle Maintenance*

Distributed System Development Challenges

- **Background Knowledge Required:** potentially, you will already need to be familiar with programming languages, operating systems, networks, theory, and algorithms!
- **It's Hard to Learn How to Fly:** you must overcome the many "accidents" initially associated with configuring your development environment, designing your first distributed system, implementing a few services, and configuring the runtime environment!
- **Fault Isolation Difficulties:** finding simple bugs is hard for novitiates because there are so many new factors to consider!
- **Does this Ever Get Easier?:** simple command-line errors can cause strange behavior and even confuse the Instructor (momentarily, of course)!

Allegheny College
Department of Computer Science

Course and Performance Objectives

- **Course Objectives:**
  - Explore the principles and paradigms associated with the discipline of distributed systems
  - **Principles:** understand the basics of naming, synchronization, consistency, replication, fault tolerance and other topics!
  - **Paradigms:** become very familiar with object-based distributed systems. Specifically, become comfortable with the Jini network technology and the JavaSpaces object repository
- **Student Performance Objectives:**
  - Be aware of the challenges and complexities associated with the design, testing, and implementation of distributed systems
  - Be familiar with the fundamental concepts
  - Develop a toolkit that can be applied to the creation of distributed systems
  - Become aware of current research and the open research questions

Allegheny College
Department of Computer Science

Instructional Objectives

- Make course time much like a Socratic dialogue!
- Create exciting laboratories that enable students to explore the Jini network technology and the JavaSpaces object repository
- Take a "gloves off" approach: allow students to grapple with all of the challenges that initially face most Jini developers!
- Encourage students to keep "laboratory notebooks" that describe the challenges, pitfalls, and solutions that they discover during the completion of a laboratory
- Require that students read scholarly and "popular press" articles that talk about the design, implementation, and testing of distributed systems

Allegheny College
Department of Computer Science

Selected Reading Material

- **Theory and Principles:**
  - Tanenbaum et al. *Distributed Systems: Principles and Paradigms*
  - Coulouris et al. *Distributed Systems: Concepts and Design*
  - Doreen L. Galli *Distributed Operating Systems*
  - Pankaj Jalote *Fault Tolerance in Distributed Systems*
  - Claudia Leopold *Parallel and Distributed Computing*
  - Nancy A. Lynch *Distributed Algorithms*
  - Tari and Bukhres *Fundamentals of Distributed Object Systems*
  - Albert Y.H. Zomaya (ed.) *Parallel and Distributed Computing Handbook*
- **Java, Jini, and JavaSpaces:**
  - David Flanagan *Java in a Nutshell*
  - W. Keith Edwards *Core Jini*
  - Freeman, Hupfer, Arnold *JavaSpaces Principles, Patterns, and Practice*
  - Scott Oaks and Henry Wong *Jini in a Nutshell*
  - Jan Newmarch**,** *Guide to JINI Technologies*

Allegheny College
Department of Computer Science

A Sampling of Covered Material

- **Principles:**
  - **Basics of Distributed Systems:** communication protocols (FTP, HTTP, and RMI), objects in distributed systems, message-oriented and loosely-coupled communication
  - **Processes:** green and native threads, light-weight processes, client and server organization, process migration and agents
  - **Names:** name resolution and closure, name space implementation, removing unreferenced entities
  - **Synchronization:** time-based synchronization, Lamport timestamps and logical clocks, global state and distributed snapshots, leader election, distributed transactions
  - **Consistency, Replication, Fault Tolerance:** availability and reliability, Byzantine failures, fail-stop and fail-silent failures, TMR and N-version programming, two-army problem and Byzantine generals problem, failure recovery techniques

Allegheny College
Department of Computer Science

Homework Assignments

- Waldo et al. *A Note on Distributed Computing*.
  - How do you define a distributed system?
  - What are the *really hard problems* that face this discipline?
  - Is it a "good thing" to always be aware of the differences between local and distributed computing?
- RMI Documentation and Waldo *The End of Protocols.*
  - What are the differences between local and remote objects in Java?
  - Is language independence an important facet of a framework that supports the implementation of distributed systems?

Allegheny College
Department of Computer Science

Homework Assignments (continued)

- Cukier et al. *Fault Injection Based on a Partial View of the Global State of a Distributed System.*
  - What is a fault injector? Why is it difficult to build a fault injector for distributed systems?
  - How can you make a fault injector that is as non-intrusive as possible?
- Avizienis. *The Methodology of N-version Programming*.
  - How do you develop a software system using an N-version programming approach?
  - What are the differences between an N-version approach and a recovery block approach?
  - How is N-version programming related to software fault tolerance?

Allegheny College
Department of Computer Science

Laboratory Assignments

- **Java Development Refresher:**
  - Re-familiarize yourself with Java compiler, virtual machine, `CLASSPATH` settings, etc.
  - Learn how to use CVS, our version control system, `jwhich`, and the Log4j logging package
- **Building Simple Jini-based Distributed Systems:**
  - Collaboratively create a simple "Hello World!" distributed system in Jini
  - Understand the hazards associated with `CLASSPATH` contamination
  - Build a distributed `StackMachine` interpreter
- **Using JavaSpaces for Distributed Service Communication:**
  - Collaboratively create a simple "Hello World!" distributed system that uses JavaSpaces
  - Refine the `StackMachine` so that it uses JavaSpaces
- **Benchmarking JavaSpaces Implementations:**
  - Conduct experiments, using Tonic, to evaluate the strengths and weaknesses of JavaSpaces

Allegheny College
Department of Computer Science

Laboratory Assignments (continued)

- **Using Federated Name Spaces:**
  - Use the `Federate` class written by Edwards to create an association between two name spaces
  - Write a Jini client that is able to use federated name spaces to search for a desired Jini service
- **Using Lookup Service Tunnels:**
  - Use the `TunnelService` class written by Edwards to create a tunnel between two name spaces
  - Write a Jini client that is able to use a name space that is connected to a tunnel
  - Experiment with the types of services that can be "pushed" through the tunnel

Allegheny COLLEGE
Department of Computer Science

Lessons Learned from the Laboratories

- Some students never hurdled the accidental difficulties associated with Jini-based distributed system development
  - For example, some students always struggled with `CLASSPATH` settings, HTTP servers and codebase settings, and RMI activation
- Once students had experienced the challenges associated with getting Jini "out of the box," they requested the provision of tools to automate these tasks
- Some students forgot material from past laboratories and found it difficult to transfer classroom "head knowledge" into laboratory "development skills"
- The laboratory notebook was not mandatory. Thus, the students that needed this tool the most often neglected it!

Allegheny College
Department of Computer Science

Submitted Final Projects

- **Implementation and Analysis:**
  - **Tuplespaces:**
    - Jini Transactions with JavaSpaces
    - Jini-Enabling TSpaces
    - "Space Off": Comparing TSpaces and JavaSpaces
    - Performance Analysis of Outrigger and GigaSpaces
  - **Java Agents:**
    - Implementing Mobile Agents with the **J**ava **A**gent **DE**velopment Framework (JADE)
  - **Metacomputing:**
    - The Frugal Metacomputing Environment
    - Distributed Regression Testing using the Frugal Metacomputer
- **Literature Review and Synthesis:**
  - High Throughput Computing with Condor
  - Loki: A Fault Injection System

Allegheny College
Department of Computer Science

Thoughts About "The Next Time Around"

- Ground more of the concepts in actual Java and Jini-based implementations
  - When talking about leader election, demo a working example of the bully algorithm; when talking about agents, use an agent-based program developed in JADE
- Make the laboratory notebook mandatory
  - Using a Wiki Web (specifically, the Sqeak Wiki, or Swiki), it is very simple to build useful laboratory notebooks
  - The best students might find this requirement to be "busy work", but they will probably still benefit from recording their experiences
  - Students that find laboratory sessions very challenging will be able to use past laboratory notebooks as a resource
- After the first or second laboratory, provide tools that automatically create a functional Jini runtime environment

Allegheny COLLEGE
Department of Computer Science

Concluding Remarks

- Current distributed systems textbooks are generally very good
- An undergraduate curriculum must balance the treatment of principles and paradigms
  - An introduction to the basic concepts of distributed systems often takes several weeks at the beginning of the semester
  - Students normally struggle with finishing the laboratory and homework assignments while still keeping up with reading
  - Appealing to a student's intuition seemed profitable: a formal treatment of some algorithms and proof techniques was too time consuming
- Most students seemed to enjoy the usage of the Jini network technology and the JavaSpaces object repository
- Jini network technology was perceived by some students as being a "difficult" framework for creating distributed systems -- but none had ever used CORBA, DCOM, or COM+ before!

Allegheny College
Department of Computer Science

Further Resources

- G. Kapfhammer Internet Site:
  - `http://cs.allegheny.edu/~gkapfham/`
- G. Kapfhammer's Current Courses, including Computer Science 490, Principles of Distributed Systems:
  - `http://cs.allegheny.edu:8080/gkapfham/8`
- Jini in Academia Internet Site:
  - `http://www.ecs.soton.ac.uk/~ra00r/jinidemia/`