

Tutorial Summary

Communicating Objects: Concurrent and Parallel Programming in Java with CSP

Christopher H. Nevison chris@cs.colgate.edu
Computer Science Dept. (315) 228-7589
Colgate University
Hamilton, NY 13346

Half-day tutorial

topic area: concurrent computing, parallel computing

prerequisites: basic knowledge of Java

Objective: This tutorial is intended to prepare the participant to develop concurrent and parallel programs in Java using the CSP model as instantiated in the JCSP library and to use these tools in teaching concurrent and parallel computing.

Presentation Format: This tutorial will be presented with about 60% demonstration of the JCSP tools and about 40% work in small groups designing example programs. There will be opportunity for discussion of the tools and comparison of them with the native control structures provided in Java. We will also have some discussion on how these tools can be used in undergraduate courses on distributed and/or parallel computing.

Biography: Chris Nevison is a professor of Computer Science at Colgate University. He has worked with parallel and concurrent programming for the past fifteen years, teaching courses at the undergraduate level, teaching several workshops for college and university faculty, editing and contributing to the book *Laboratories for Parallel Computing*, and doing research on parallel discrete event simulation. Nevison has worked with the CSP model for concurrent programs for most of this time, first using the language Occam 2, a parallel programming language based on CSP, and more recently using and extending the JCSP library for Java. Nevison is currently working on developing materials for teaching concurrent and parallel programming using Java with CSP, supported by several NSF grants (most recently DUE-9952260). Nevison is also a member of ACM SIGCSE and has worked extensively with the Advanced Placement Program in Computer Science.

Tutorial Outline

Communicating Objects: Concurrent and Parallel Programming in Java with CSP

The tutorial will include presentation of concepts and constructs of Java with CSP. Each presentation will be followed by an opportunity for participants to carry out exercises that demonstrate the use of the ideas covered in the presentation. Participant solutions will be discussed and compared to the instructor's solutions in order to develop a fuller understanding of the use of Java with CSP.

1. Introduction

Motivation

- Concurrency of real-world phenomena
- Difficulties with Java primitives for concurrency

- Advantages of CSP model
- Basic objects used for constructing CSP programs
- channels, processes, parallel
- Simple examples demonstrating these constructs
- JCSP “plugNplay” tools
- Participant exercises
- Building simple computations.
- Fibonacci generator, Series summation
2. Selection in CSP
- Selecting among concurrent alternatives
- issues
 - JCSP construct, Alternative
- Participant exercise
- a simple simulation
3. JCSP AWT library
- Implementation
- replace Java listeners and configuration with CSP message model
 - graphics/window library meshed with CSP model
4. Concurrent O-O design problem
- Example: Pong
- Participant work: creating concurrent O-O design for game
- Discussion
- Participant work: implementing components ball, paddle
5. Channels over network links
- Channel links
- Example: Pong on two machines
- Participant Exercises
- A simple Chat room
 - Making the Chat room robust
6. Parallel Computation
- Parallel algorithms
- CSP implementations
- Parallel computation
- Multi-processor
 - Cluster

Description:

We will learn how concurrent and parallel programs in Java can be designed and implemented using the JCSP library from the University of Kent. JCSP enables us to develop robust concurrent programs that preserve the integrity of the O-O model of encapsulation. We will also learn to develop parallel programs that run over a cluster of workstations.