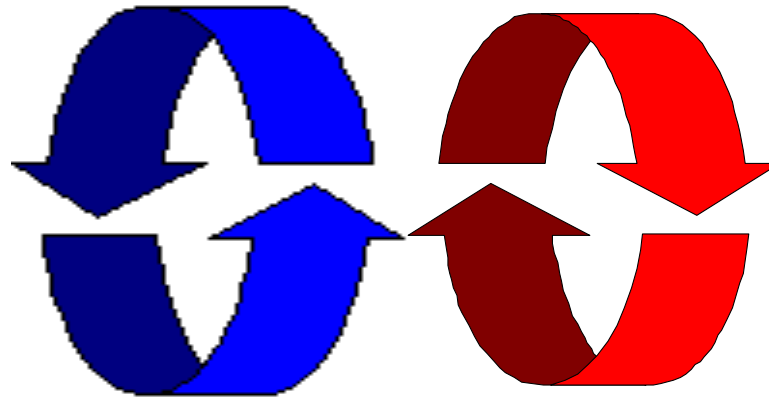


Concurrency

Concepts, Models and Programs



(Most of the) Material Freely Online!

staff.city.ac.uk/c.kloukinas/concurrency

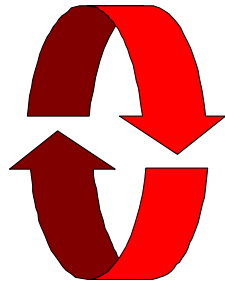
Book by:

Jeff Kramer and Jeff Magee

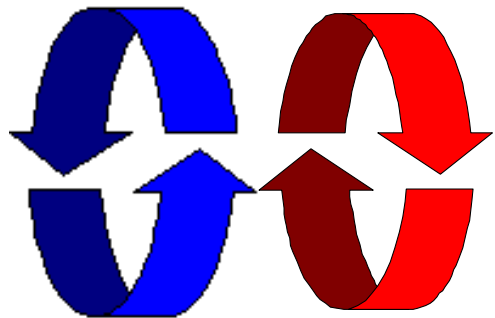
What Shall we Learn?

- The *theoretical foundations* of concurrency
- Its *main pitfalls* (race conditions, deadlocks, resource starvation, etc.)
- The *solutions* to these
- How to *automatically verify* your designs
- How to *develop correct multi-threaded systems*.

What is a Concurrent Program?



A **sequential** program has a single thread of control.



A **concurrent** program has multiple threads of control allowing it perform multiple computations in parallel and to control multiple external activities which occur at the same time.

Can be done on a single core! (*multi-tasking*)

You've done it already! (*event handlers*)
(*most probably wrong...* 🙄)

Concurrency is widespread but error prone!

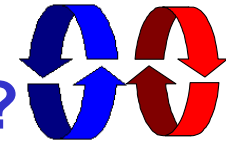
- ◆ Therac - 25 computerised radiation therapy machine

Concurrent programming errors contributed to accidents causing deaths and serious injuries.

- ◆ Mars Rover

Problems with interaction between concurrent tasks caused periodic software resets reducing availability for exploration.

So, Why Bother with Concurrent Programming?



- ◆ Performance gain from multiprocessing hardware
 - e.g. fine grain parallelism on multicore hardware : low level memory models
 - e.g. coarse grain parallelism for partitioned scientific calculations : processes
- ◆ Increased application throughput : *avoid polling (busy waiting)!*
 - e.g. an I/O call need only block one thread
- ◆ Increased application responsiveness
 - e.g. high priority thread for user requests.
- ◆ **More appropriate structure**
 - for programs which interact with the environment, control multiple activities and handle multiple events – coarse grain parallelism.

Module Approach

◆ Concepts

- **we adopt a model-based approach for the design, analysis and construction of concurrent programs**

◆ Models

- **we use finite state models to represent concurrent behaviour.**

◆ Practice

- **we use Java for constructing concurrent programs.**

Examples are used to illustrate the concepts, models and demonstration programs.

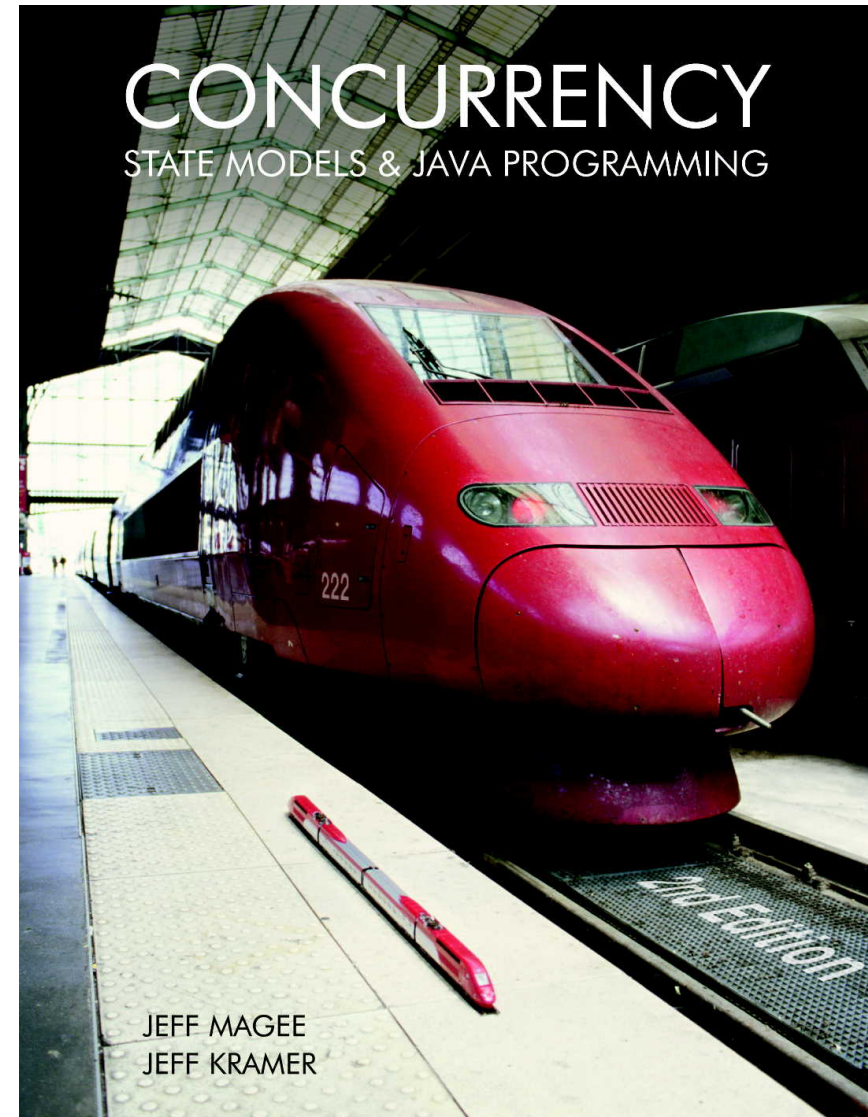
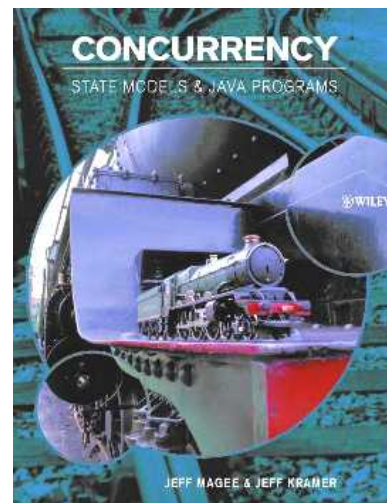
Book

Concurrency: State Models & Java Programs, 2nd Edition

Jeff Magee &
Jeff Kramer

WILEY

1st
edition



Course Outline

- 2. Processes and Threads**
- 3. Concurrent Execution**
- 4. Shared Objects & Interference**
- 5. Monitors & Condition Synchronization**
- 6. Deadlock**
- 7. Safety and Liveness Properties**
- 8. Model-based Design (*underlying theme*)**

The main basic
Concepts
Models
Practice

Advanced topics ...

- 9. Dynamic systems
- 10. Message Passing
- 11. Concurrent Software Architectures
- 12. Timed Systems
- 13. Program Verification
- 14. Logical Properties