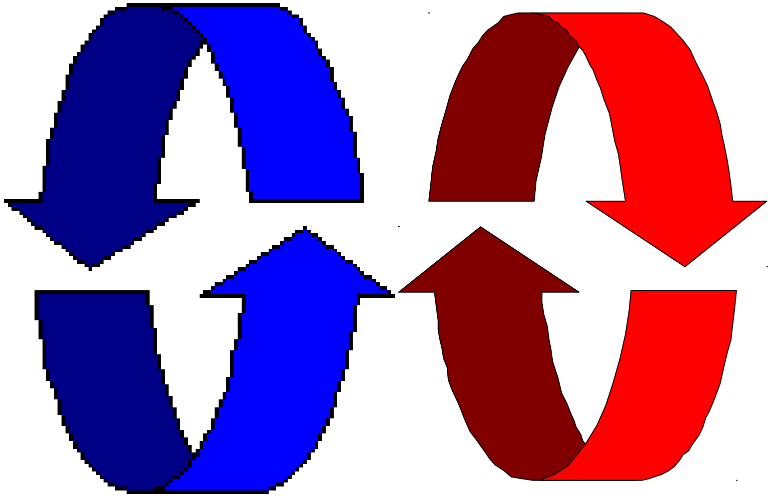
Concurrency

State Models and Java Programs



Nobuko Yoshida,

Raymond Hu, Julien Lange

n.yoshida raymond.hu05 j.lange

@ imperial.ac.uk

Logistics

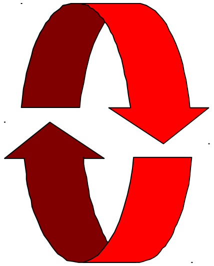
- Lectures
  - Mondays 11am-12pm @342
  - Fridays 11am-1pm @144
- Labs/Tutorials Mondays 12pm-1pm (202)
- TAs:
  - Julien Lange, Nicholas Ng
- Software:
  - LTSA Analyzer (available from the website)
  - Java

Note

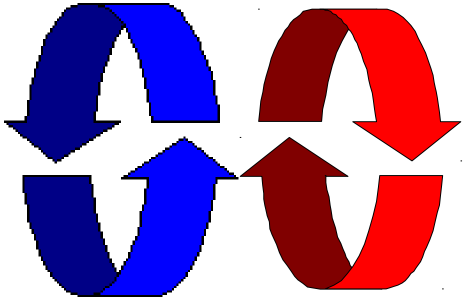
•Please print the lecture notes available from CATE from the next lecture. We will not provide copies.

- Tutorial Answers are not attached in CATE. We shall give the next tutorial.
- Feedbacks and Revisions (Cws, Java, LTSA, Examination)
- Examination is Week 11 (Answer 3 out of 4 Questions)
- Coursework 1 (Publish 2 Feb and Deadline 16 Feb: Elect)
  Coursework 2 (Publish 23 Feb and Deadline 09 Mar: Elect)

What is a Concurrent Program?



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



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

Why Concurrent Programming?

- Performance gain from multiprocessing hardware
  - parallelism.
- Increased application throughput
  - an I/O call need only block one thread.
- Increased application responsiveness
  - high priority thread for user requests.
- More appropriate structure
  - for programs which interact with the environment, control multiple activities and handle multiple events.

Do I need to know about concurrent programming?

Concurrency is widespread but error prone.

- Therac - 25 computerised radiation therapy machine
  - **Concurrent programming errors contributed to accidents causing deaths and serious injuries.**
  - The article which reports Therac 25 accidents is available from CATE.

models

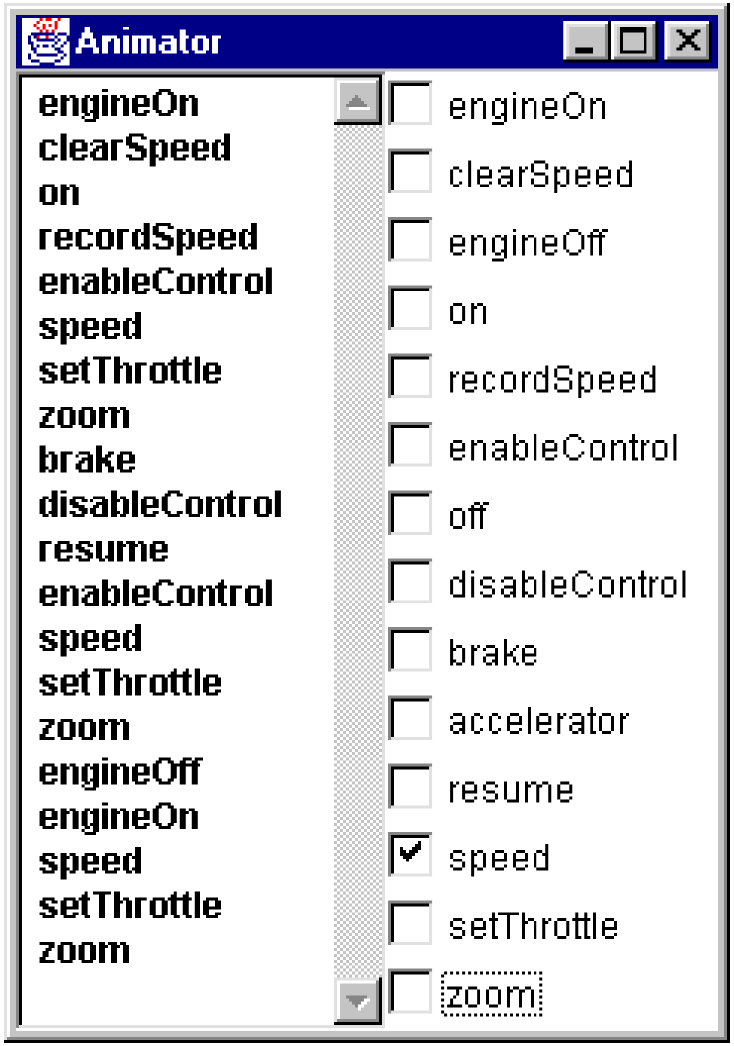
A model is a simplified representation of the real world.

Engineers use models to gain confidence in the adequacy and validity of a proposed design.

- focus on an aspect of interest - concurrency
- model animation to visualise a behaviour
- mechanical verification of properties (safety & progress)

Models are described using state machines, known as Labelled Transition Systems **LTS**. These are described textually as finite state processes (**FSP**) and displayed and analysed by the ***LTSA*** analysis tool.

modeling the Cruise Control System

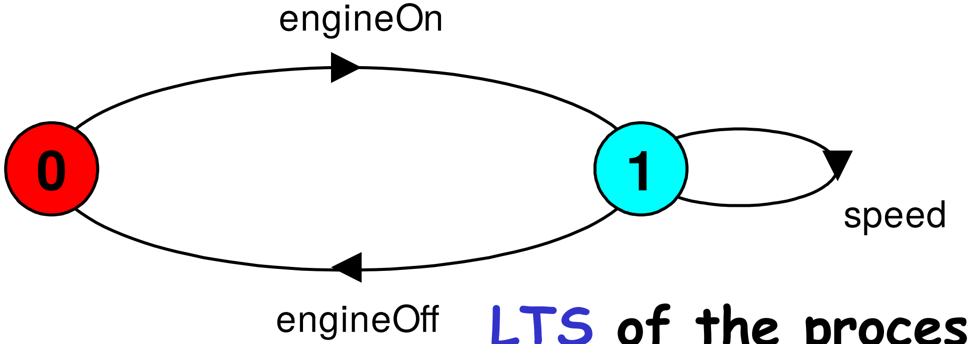


Animator

engineOn
clearSpeed
on
recordSpeed
enableControl
speed
setThrottle
zoom
brake
disableControl
resume
enableControl
speed
setThrottle
zoom
engineOff
engineOn
speed
setThrottle
zoom

- □ engineOn
- □ clearSpeed
- □ engineOff
- □ on
- □ recordSpeed
- □ enableControl
- □ off
- □ disableControl
- □ brake
- □ accelerator
- □ resume
- ☑ speed
- □ setThrottle
- □ zoom

LTSA **Animator to step through system actions and events.**



LTS **of the process that monitors speed.**

Later chapters will explain how to construct models such as this so as to perform animation and verification.

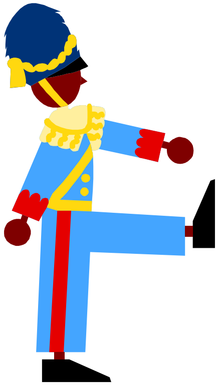
programming practice in Java

Java is

- widely available, generally accepted and portable
- provides sound set of concurrency features

Hence Java is used for all the illustrative examples, the demonstrations and the exercises. Later chapters will explain how to construct Java programs

"Toy" problems are also used as they exemplify particular aspects of concurrent programming problems!



course objective

This course is intended to provide a sound understanding of the ***concepts***, ***models*** *and* ***practice*** involved in designing concurrent software.

The emphasis on principles and **concepts** provides a thorough understanding of both the problems and the solution techniques. **Modeling** provides insight into concurrent behavior and aids reasoning about particular designs. Concurrent programming in **Java** provides the programming **practice** and experience.

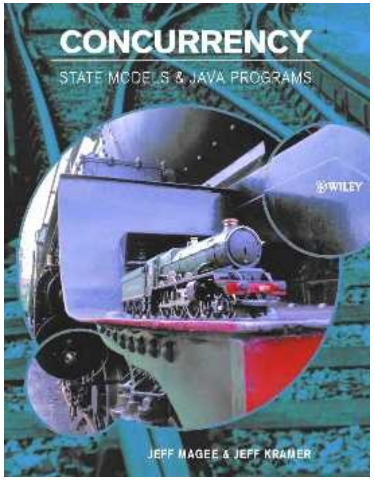
Book (http://www.doc.ic.ac.uk/~jnm/book/)

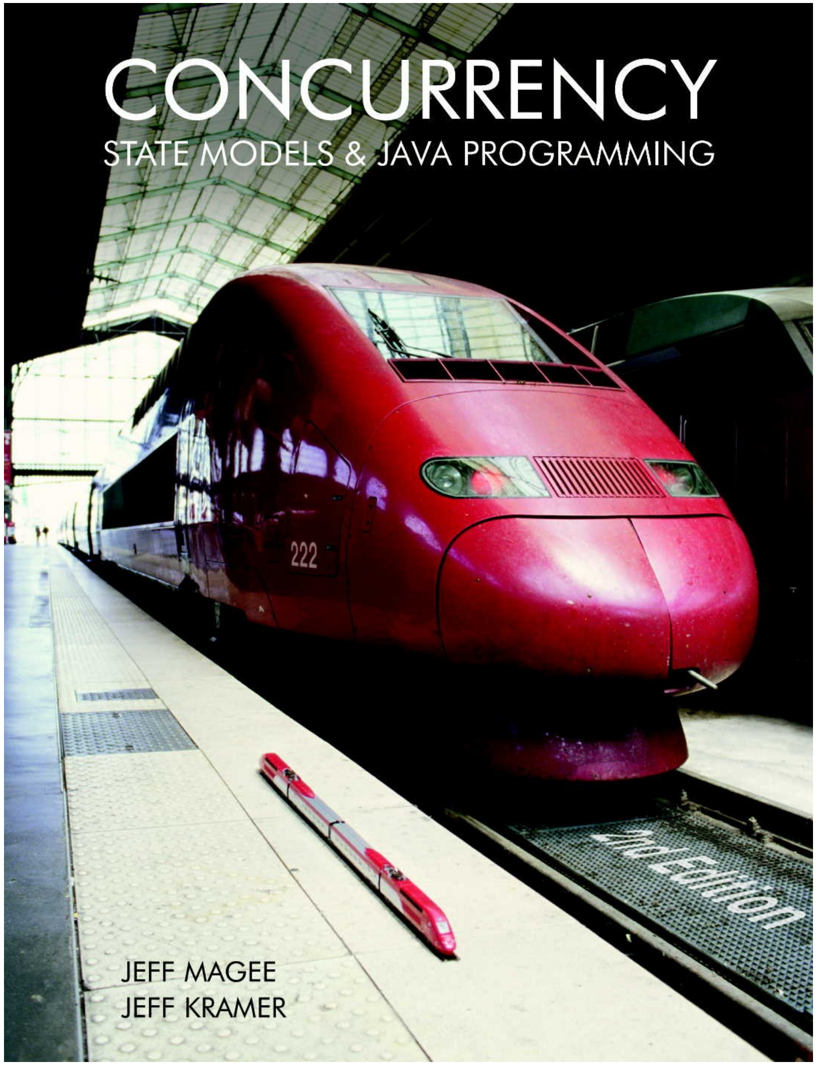
Concurrency: State Models & Java Programs,
2nd Edition

Jeff Magee &
Jeff Kramer

WILEY

1st
edition





Course Outline

1. Processes and Threads
2. Concurrent Execution
3. Shared Objects & Interference
4. Monitors & Condition Synchronization
5. Deadlock
6. Safety and Liveness Properties
7. Model-based Design

The main basic

Concepts

Models

Practice

Advanced topics …

1. Dynamic systems
2. Message Passing
3. Concurrent Software Architectures

1. Timed Systems
2. Program Verification
3. Logical Properties

Web based course material

www.wileyeurope.com/college/magee

- Java examples and demonstration programs
- State models for the examples
- Labelled Transition System Analyser (***LTSA***) for modeling concurrency, model animation and model property checking.
- *Jeff Kramer's article (CACM) which explains an importance of Abstraction for Computing is available from CATE*

Summary

- Concepts
  - **we adopt a model-based approach for the design and construction of concurrent programs**
- Models
  - **we use finite state models to represent concurrent behavior.**
- Practice
  - **we use Java for constructing concurrent programs.**

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