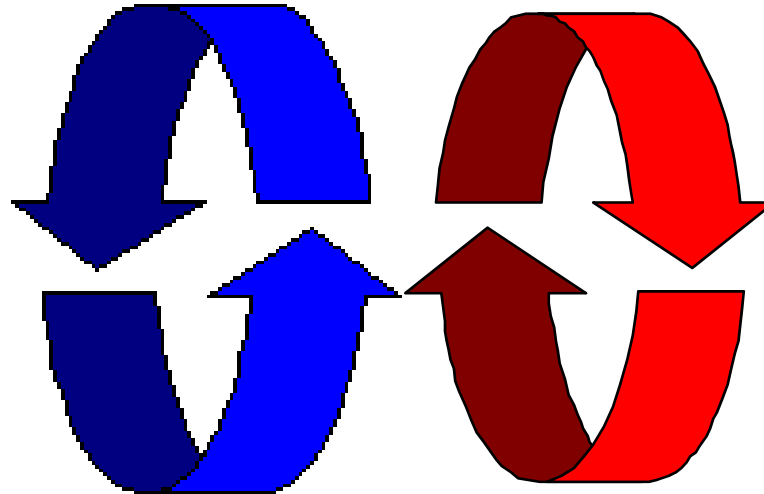


# Concurrency

---

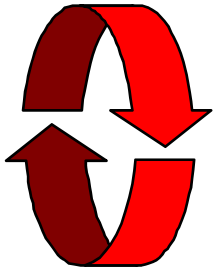
## *State Models and Java Programs*



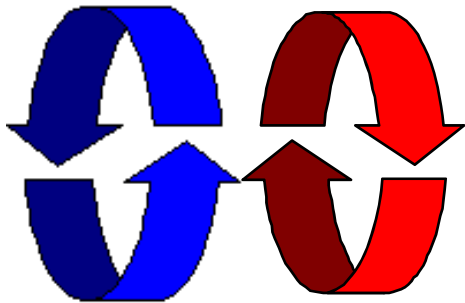
**Jeff Magee** and **Jeff Kramer**

# 並行プログラムとは何か

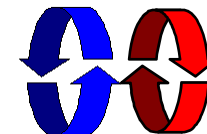
---



**逐次**プログラムは制御の糸が1本である。



**並行**プログラムは複数の制御の糸を持ち、並列に複数の計算ができるようにし、また同時に起こる複数の外部の活動を制御できるようにする。



## なぜ並行プログラミングが必要か？

---

- ◆ ハードウェアの多重化処理による性能向上
  - 並列性
- ◆ 適用ソフトウェアのスループットの向上
  - I/Oは1つのスレッドをブロックするだけ
- ◆ ソフトウェアの応答性の向上
  - 利用者の要求に対して優先度の高いスレッドを割り当てる.
- ◆ より適切な構造
  - 環境と相互作用し, 多重な動作を制御し, 多重な事象を操作するプログラムに適した構造

# 並行プログラミングを学ぶ必要性

---

*並行性はよく見られるが誤りやすい。*

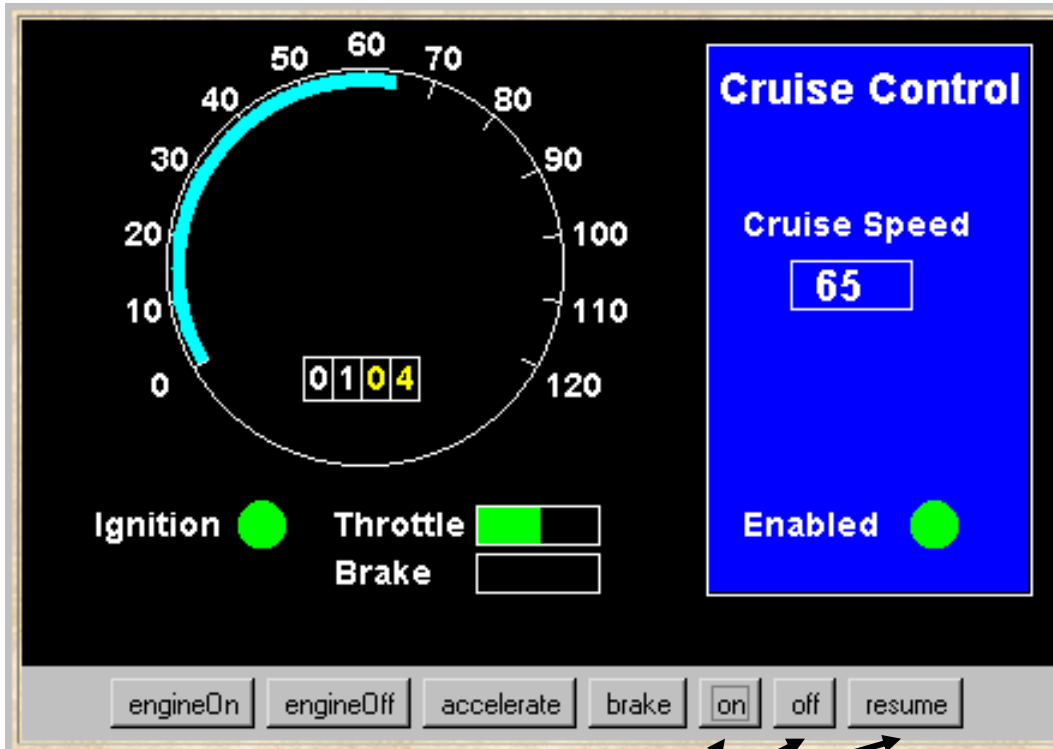
- ◆ テラック-25 計算機制御の放射線治療機械

並行プログラミングのエラーで死者, 重傷者を出す事故が発生

- ◆ 火星ローバー

並行タスク間の相互作用の問題で, 周期的にソフトウェアのリセットが起こり, 探査のための使用可能時間が減少

# 巡航制御システム



ボタン

- ◆ このシステムは安全か?
- ◆ テストですべてのエラーが見つかるか?

クルマのエンジンがかかっていて、オン・ボタンが押されると、現在の速度が記録され、巡航制御システムが稼動状態となる: クルマの速度は記録された速度に保たれる。

ブレーキ, アクセルを踏むか, オフ・ボタンを押すと, システムは稼動しなくなる. レジューム・ボタンを押すと, システムが再稼動する。

# モデル

---

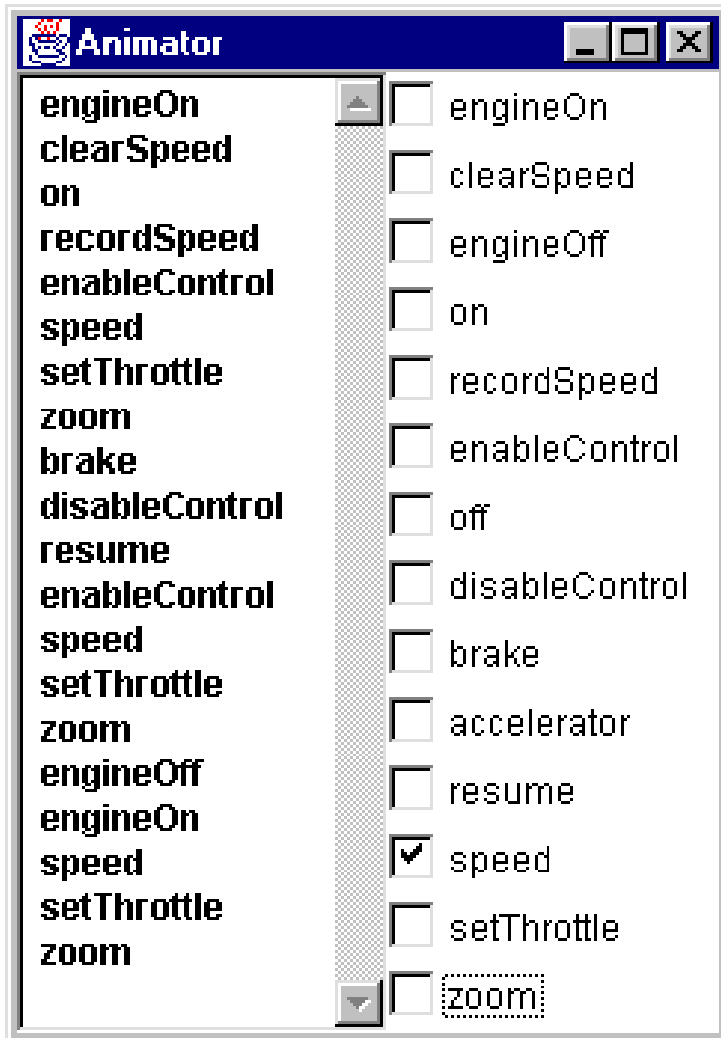
モデルとは実世界を簡潔に表現したものである。

エンジニアは自分の設計が充分で正しいと自信が持てるように、モデルを用いる。

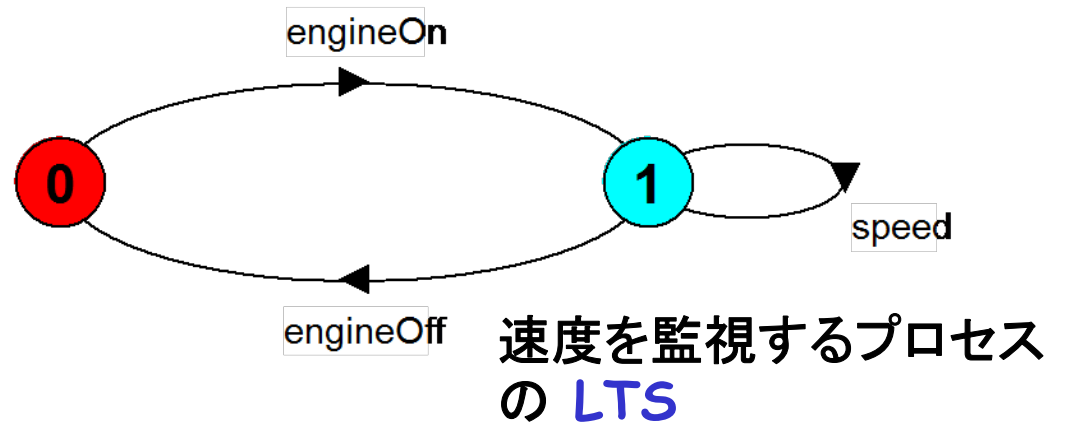
- ◆ 関心のある側面—並行性に焦点を当てる。
- ◆ 振舞いを視覚化するために動くモデルを作る。
- ◆ 性質(安全性や進行性)の機械的な証明をモデル上で行う。

ラベル付き遷移システム (Labelled Transition Systems **LTS**) という状態機械を用いてモデルを記述する。有限状態プロセス (finite state processes **FSP**) としてテキスト表現し、**LTSA** という分析ツールで表示や分析を行う。

# 巡航制御システムのモデル化



*LTSA* Animator でシステムの動作と事象をステップごとに追う。



後の章で、このようなモデルをどう構築し、動画や検証を実現できるかについて述べる。

# Javaによるプログラミングの実践

---

Java は

- ◆ 広く使われており, 移植性が高い
- ◆ 健全な並行処理機能を備えている

そこで, 例題, デモ, 演習にはすべてJavaを用いる. 後の章で, たとえば巡航制御システムのような Javaプログラムをどのように作成するかを述べる.

*“玩具”問題は, 並行プログラミング問題の特定な側面を明確にするためにも用いる.*



## 本講の目標

---

本講は並行ソフトウェアを設計する際に必要なに関する**概念**、**モデル**、**実践**を正しく理解することをねらいとする。

原理と**概念**に重点をおくことにより、問題と解法の両者について十分な理解が得られるようにする。また**モデル化**により並行動作についての洞察が与えられ、特定の設計について考察することの助けとなる。**Java**によるプログラミングで、プログラミングの**実践**と経験が得られる。

## 講義の概要

- ◆ プロセスとスレッド
- ◆ 並行実行
- ◆ 共有オブジェクトとインターフェース
- ◆ モニタと条件同期
- ◆ デッドロック
- ◆ 安全性と活性
- ◆ モデルベースの設計

概念  
モデル  
実践

- ◆ 動的システム
- ◆ 並行ソフトウェア・アーキテクチャ
- ◆ メッセージ送信
- ◆ 時間システム

<http://www-dse.doc.ic.ac.uk/concurrency/>

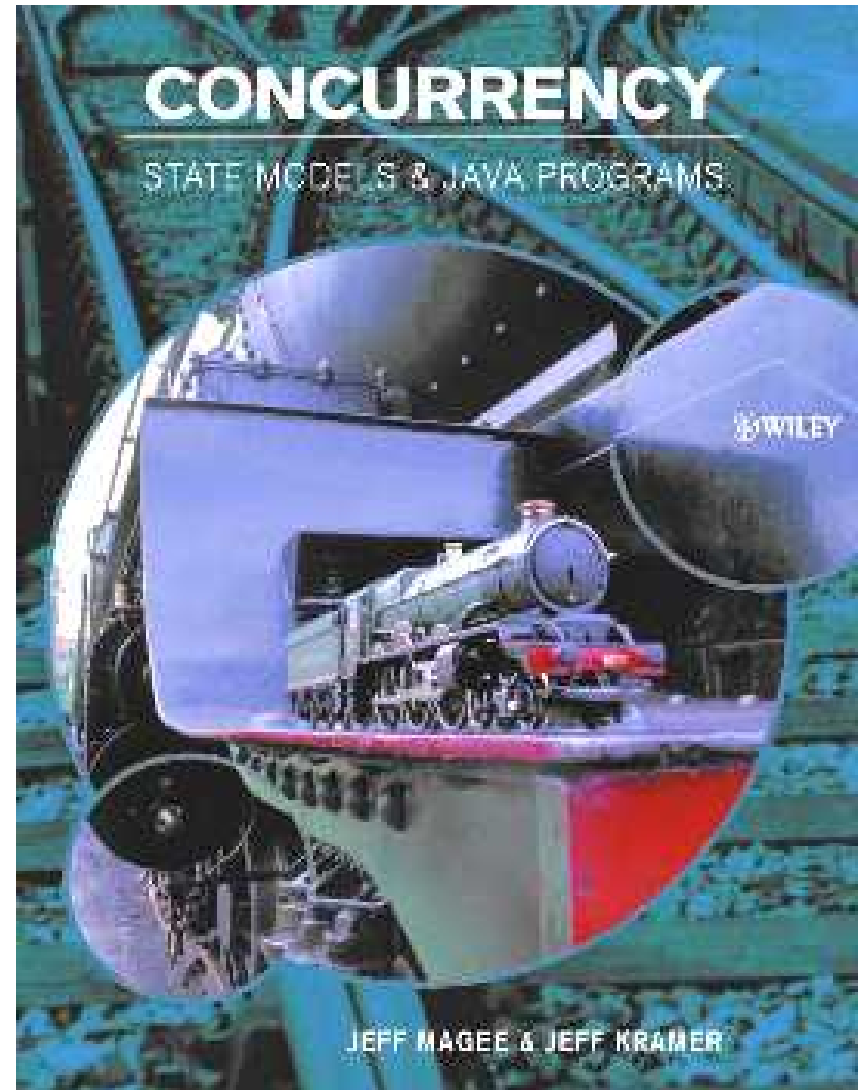
- ◆ Java の例題とデモ・プログラム
- ◆ 例題用の状態モデル
- ◆ 並行システムのモデル化, モデルの動画, モデル性質検査のためのラベル付き遷移システム分析系 (*LTSA*)

# 教科書

## Concurrency: State Models & Java Programs

Jeff Magee &  
Jeff Kramer

WILEY



## まとめ

---

### ◆ 概念

- 並行プログラムの設計と構築のためにモデルベースの方法を採用する.

### ◆ モデル

- 並行な振舞いを表現するのに有限状態モデルを用いる.

### ◆ 実践

- Javaを用いて並行プログラムを作成する.

*概念, モデル, デモ・プログラムを説明するために例題を用いる*