**EDUCATION PROGRAM FOR TOP SOFTWARE ENGINEERS** 

2023.11.24 講座説明会

## テストと検証シリーズご紹介

### 担当講師:

- 石川 冬樹 (NII)
- 宇佐美 雅紀 (Amazon Japan)
- 粂野 文洋(日本工業大学)
- 久連石 圭(東芝)
- 田辺 良則(鶴見大学·NII)
- 長谷川哲夫(東芝)

- 初谷 久史 (PRINCIPIA)
- 早水 公二(フォーマルテック)
- 藤本 洋 (イーソル)
- 前田 直人(NEC)
- 松崎 和賢(中央大学)
- 吉岡 信和(早稲田大学)



## テストと検証シリーズ

ソフトウェアの品質確保に関する手法

- テスティング
  - テスティング (基礎)
- ■静的解析
  - ■プログラム解析
- ■モデル検査
  - ■モデル検査入門I, II
  - ■設計モデル検証
  - ■モデル検査特論
  - ■モデル検査事例演習



## テストと検証シリーズ

## ソフトウェアの品質確保に関する手法

- テスティング
  - テスティング (基礎)
- ■静的解析
  - ■プログラム解析
- ■モデル検査
  - ■モデル検査入門I, II
  - ■設計モデル検証
  - ■モデル検査特論
  - ■モデル検査事例演習



## テスティング (基礎)

- テストツール
- ■テスト駆動開発
- ブラックボックステスト
  - ■直交表
- ■ホワイトボックステスト
  - ■パスカバレッジ
- ■テスト自動化
- ■テスト計画
- テストの管理と評価・改善



## テストと検証シリーズ

ソフトウェアの品質確保に関する手法

- テスティング
  - ■テスティング (基礎)
- ■静的解析
  - ■プログラム解析
- ■モデル検査
  - ■モデル検査入門I, II
  - ■設計モデル検証
  - ■モデル検査特論
  - ■モデル検査事例演習



## プログラム解析

- ■オムニバス形式で、いろいろな静的解析の手法・ ツールを紹介する.
  - ■有界モデル検査 / CBMC
  - ■データフロー解析 / SOBA
  - ■記号実行 / Triton
  - ■契約による設計 / JML



## テストと検証シリーズ

ソフトウェアの品質確保に関する手法

- テスティング
  - ■テスティング(基礎)
- ■静的解析
  - ■プログラム解析
- ■モデル検査
  - ■モデル検査入門I, II
  - ■設計モデル検証
  - ■モデル検査特論
  - ■モデル検査事例演習



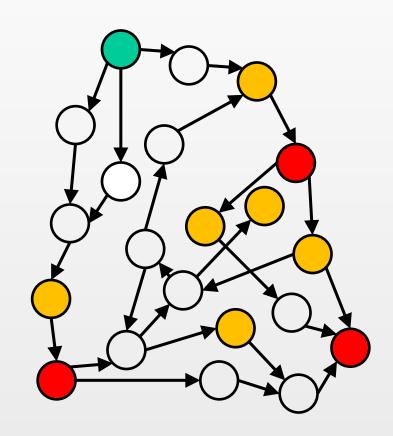
## モデル検査: ターゲット

- ■再現が難しい不具合
  - ■並行実行に起因するもの
- ■(例) 以下の条件が揃ったときだけ、ファイルをクローズしそこねる:
  - ■スレッド1 が メソッドa の20行目を実行したが, まだ21行目は実行していない
  - ■スレッド2 が メソッドb の53行目を実行した



## モデル検査:考え方

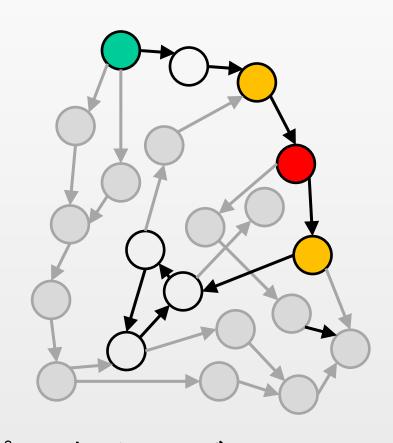
- ■システムが取り得る あらゆる状態を含む 状態遷移系を構築 する.
- 状態遷移系を効率 よく探索して、不具 合を起こす経路の 有無を判断.



緑: スタート, 黄色: オープン, 赤: クローズ オープンしたら必ずクローズされるか?

## モデル検査:考え方

- ■システムが取り得る あらゆる状態を含む 状態遷移系を構築 する.
- 状態遷移系を効率 よく探索して、不具 合を起こす経路の 有無を判断.



緑: スタート, 黄色: オープン, 赤: クローズ オープンしたら必ずクローズされるか?

## モデル検査:学ぶ意義

- ■並行実行に起因する,再現が難しい不具合を追求する手段を獲得できる
- モデリング技術・抽象化技術の適用を学べる

EDUCATION PROGRAM FOR TOP SOFTWARE ENGINEERS

## モデル検査:手順





仕様書だけ? ソースコードは?

、状態遷移系 (数学的モデル)

遷移系の 作り方は?

成り立つべき性質

性質って どう書くの?

時間制約も 扱える?

モナル

モデル検査器の 使い方は?

他の開発工程 との関係は?

または

モデル検査が 有効な領域は?

検査結果を説 明してわかって もらえるかな?



性質って どう書くの?



## モデル検査入門」、II

- ■モデル検査の基礎理論
  - ■検証する性質の表現 = 時相論理
    - Linear Temporal Logic
    - Computational Tree Logic
  - ■状態遷移系
- 代表的なモデル検査器に習熟
  - ■SMV 状態遷移系を直接記述
  - SPIN プロセスをプログラミング言語風に記述
- ■ロボットを利用した実習

遷移系の 作り方は?

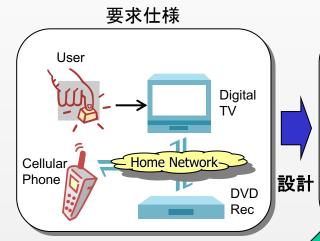
モデル検査器 の使い方は? EDUCATION PROGRAM FOR TOP SOFTWARE ENGINEERS

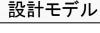
## 他の開発工程 との関係は?

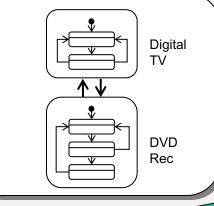
実装



## 設計モデル検証



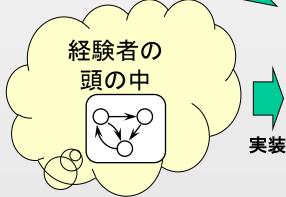




#### ソースコード

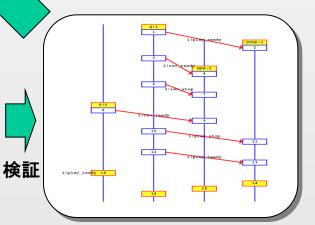
```
public class Client extends Thread {
 public Object lock1;
 public Object lock2;
 public Client(Object lock1, Object lock2) {
  this.lock1 = lock1;
  this.lock2 = lock2;
 public void run() {
  synchronized(lock1) {
   synchronized(lock2) {}
```

検証要求分析 検証モデル設計



```
mtype DVD_recorder_state = waiting;
active proctype HD_recorder() {
 mtype event;
  :: HD_ch?event ->
     :: (HD_recorder_state == waiting && event == req_reserve_HD) ->
      HD_recorder_state = ready;
DVD_ch!ack_reserve_HD
     :: (HD_recorder_state == ready && event == req_play_HD) ->
      progress1: HD_recorder_state = playing;
      DVD_ch!ack_play_HD
     :: handle_playing_terminated_HD()
  :: handle_playing_terminated_HD()
```

#### 設計誤り発見



検証モデル

モデル検査記述

検証結果/反例

EDUCATION PROGRAM FOR TOP SOFTWARE ENGINEERS

モデル検査が 有効な領域は?

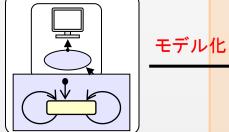


モデル検査特論 (その1)

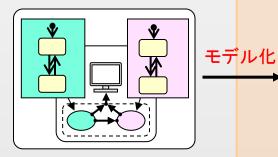
モデル検査器 FDRによる検証 プロセス代数CSP

検証

逐次システムや仕様



並行システムや仕様



によるモデル化

#### CSPモデル

SRseq(K) = SendRec(0,0,0,K)SendRec(n,m,i,K)

- = i<K & disp!send.n
- $\rightarrow$  SendRec((n+1)%N,m,i+1,K)  $\square$ i>0 & disp!receive.m
- $\rightarrow$  SendRec(n,(m+1)%N,i-1,K)

CSPモデル

#### SRconc

= (Sender(0) [| {| chan |} |] Receiver(0))\{| chan |}

#### Sender(n)

- = disp!send.n  $\rightarrow$  Sender'(n) Sender'(n)
- = chan!n  $\rightarrow$  Sender((n+1)%N)

#### Receiver(m)

- = chan?m  $\rightarrow$  Receiver'(m)
- Receiver'(m)
- = disp!receive.m → Receiver(m)

並行システム

逐次システム

FDRスクリプト

検証項目

**FDR** 



JCSPプログラム

プロセス並行合成 Senderの動作

実装 Receiverの動作 Java

■ 選択 DOS - java srconc.SRconc

send.6 receive.6 send.7 send.8 receive.7 send.9 receive.8 receive.9 receive.0 send. 1 receive.1

Javaライブラリ JCSPによる実装 \_ 🗆 ×

## モデル検査特論 (その2)

時間制約も 扱える?

- 6 CHEERS EDUCATION OF THE PROPERTY OF THE PROP
- 組込みソフトウェアの時間制約に関する設計検証
- 講座で取り扱う課題
  - ■いろいろな時間要素の検証モデルへの反映方法と 検証結果分析の難しさ
  - ■設計プロセスと検証プロセスのシームレスな連携
- 利用するツール
  - ■時間制約を扱えるモデル検査ツールUPPAAL
    - ■スウェーデンのUppsala大学とデンマークのAalborg 大学により開発



## モデル検査事例演習

【目的】企業におけるモデル検査推進者の育成

■ モデル検査の実務(組織)を想定し、その開始から終了までの 全プロセスを体験する演習

### 【習得できる知識】

- 組織におけるモデル検査の適用プロセス
  - 1. 開発者からのヒアリング
  - 2. ソフトウェアのモデル化
  - 3. モデル検査
  - 4. 結果報告
  - 5. 最終報告書
- モデル検査報告書の作成方法
- モデル検査の導入を推進するためのノウハウ

【検証対象】実例を基にしたソースコード(C言語)、もしくは仕様書

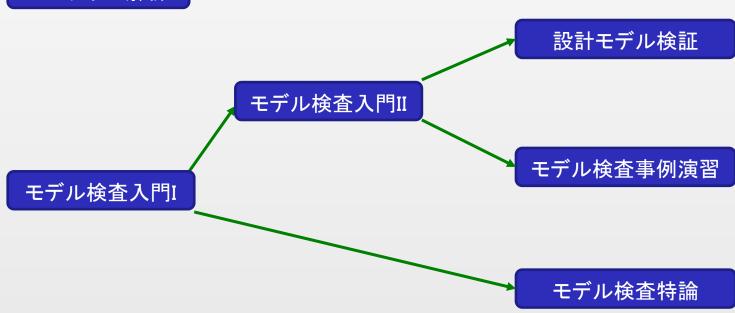
検査結果を説 明してわかって もらえるかな?

## B CHEERS EDUCATION PRODUCTION OF THE NGINEERS OF THE NGINEERS

## 科目間依存関係

テスティング基礎

プログラム解析



# 9 CHERS EDUCATION DA PROPERTIES OF THE PROPERTY OF THE PROPERT

## 目的別科目選択例 -1-

ソフトウェア工学の基礎科目を幅広く受講したい

- ■テスティング基礎
- ■モデル検査入門I

# O CHERS EDUCATION DE PROPERTIES DE LA CONTROL DE LA CONTRO

## 目的別科目選択例 -2-

## モデル検査技術をマスターしたい

- ■モデル検査入門I・II
- ■設計モデル検証
- ■モデル検査特論
- ■モデル検査事例演習



## 目的別科目選択例 -3-

テスティング技術を基礎から応用まで学習したい

- ■テスティング基礎
- ■プログラム解析

## 目的別科目選択例 -4-

ソフトウェアの堅牢な構築についての技術全般に 興味があり、マスターしたい

- ■テスティング基礎
- ■プログラム解析
- ■モデル検査入門Ⅰ・Ⅱ
- ■設計モデル検証
- ■モデル検査特論
- ■モデル検査事例演習

23

EDUCATION PROGRAM FOR TOP SOFTWARE ENGINEERS

## [補足] モデル検査系科目群のまとめ

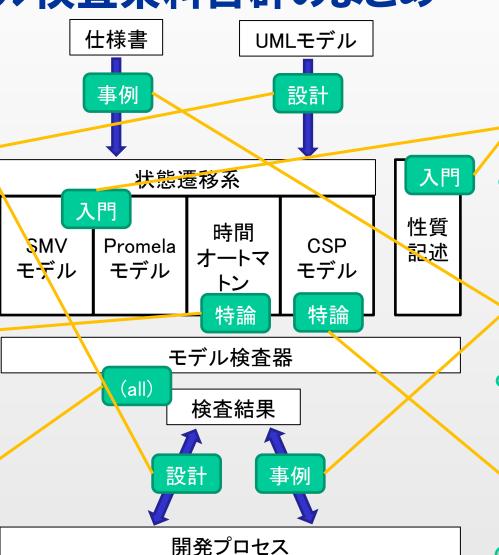


設計モデル検証では.

UMLからの状態遷移系 の作り方を学びます. 開 発プロセス中にモデル検 査をどう組み込むかを検 討します.

モデル検査特論では、 時間制約をモデル検査 する方法を学習します.

すべての講義で、モデ ル検査器を用いた実 習を行います.



モデル検査入門では.

基本的な状態遷移系と 性質記述を学びます.

#### モデル検査事例演習で

は、モデル検査の専門 家と開発者の協働方法 を検討します. 仕様書か らの状態遷移系作成も 扱います.

#### モデル検査特論では.

CSPモデルから状態遷 移系を作りモデル検査 し、同時に実装系も生 成する手法を学習しま す.