PETRI NET BASED
DEPENDABILITY ENGINEERING
OF REACTIVE SYSTEMS

MONIKA HEINER

monika.heiner@b-tu.de
http://www.informatik.tu-cottbus.de
EXAMPLE, PRODUCTION CELL:
- deposit belt (belt 2)
- elevating rotary table
- feed belt (belt 1)
- travelling crane
- robot
- arm 1
- arm 2
- press

EXAMPLE, CONCURRENT PUSHERS:
- Controller 1
- Controller 2
- Pos. 2
- Pos. 3
- Pusher 2
- Piece, Pos. 1
- M

14 sensors
34 commands
EXAMPLE, CRUISE CONTROL

- Pressing **on**, while car **ignition** is switched on
  -> Current speed is recorded and system is enabled

- Pressing **brake**, **accelerator** or **off**
  -> System is disabled

- Pressing **resume**
  -> Re-enables the system

MOTIVATION

CONCURRENCE IS WIDESPREAD, BUT ERROR PRONE

- Therac-25 computerized 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

. . .

OBVIOUS QUESTIONS

- Is a system safe?
- Is a system reliable?
- Would testing be sufficient to discover all errors?
**PRELIMINARIES**

- **DEPENDABILITY**
  - ability of a system to fulfill its predefined task (in spite of any hardware and/or software faults)
  - dependability modelling
    - Which kind of models?
    - Where do the models come from?
  - engineer’s basic principle:
    - **KEEP EVERYTHING AS SIMPLE AS POSSIBLE!**
    - dedicated models for different kinds of properties;

**METHODS**

**SOFTWARE DEPENDABILITY**

- **FAULT AVOIDANCE**  →  *development phase*
  - **FAULT PREVENTION**
  - **FAULT REMOVAL**
    - **MANUAL**
    - **COMPUTER-AIDED**  →  *VALIDATION*
      - animation / simulation / testing
      - context checking (static analysis)
      - consistency checking (verification)
  - **FAULT TOLERANCE**  →  *operation phase*
    - **FAULT MASKING**
      - DEFENSIVE
      - DIVERSITY
    - **FAULT RECOVERY**
MODEL BASED SYSTEM VALIDATION, GENERAL PRINCIPLE

What was in the beginning?

MODEL-BASED SYSTEM VALIDATION, TWO APPROACHES

problem

system
modelling
model
validation
implementation

system
model
validation
A POSTERIORI MODEL BASED SYSTEM VALIDATION, PROCESS AND TOOLS

- controller
- environment
- compiler
- modelling
- library
- control model
- environment model
- composition
- temporal logic
- system model
- set of temporal formulae
- verification methods
- safety requirements
- functional requirements
- errors / inconsistencies

OBJECTIVE - REUSE OF CERTIFIED COMPONENTS

REAL PROGRAM

SAFETY REQUIREMENTS

DREAM PROGRAM

FUNCTIONAL REQUIREMENTS
**ANOTHER APPLICATION: BIOCHEMICAL SYSTEMS, EXAMPLES**

- **metabolic pathways / networks**
  - stoichiometric relations known
  - concentrations of metabolites often known

- **signal transduction pathways / networks**
  - stoichiometric relations unknown
  - read arcs / test arcs
  - inhibitor arcs

- **gene regulatory networks**
  - stoichiometric relations unknown
  - mRNA concentrations often known
  - protein concentrations are hard to be measured
  - often a mixture of metabolic and signal transduction pathways

=>>> networks of elementary actions

**MODEL- BASED SYSTEM ENGINEERING**

- **biochemical system**
  - validation
  - known properties
  - unknown properties

- **behaviour prediction**
  - model
  - model properties

**GENERALIZATION TO BIOCHEMICAL SYSTEMS**
**BIONETWORK, EX 1**

**G-PP PATHWAYS**

**GLYCOLYSIS / PENTOSE PHOSPHATE PATHWAYS IN ERYTHROCYTES**

[Reddy 1996]

---

**BIONETWORK, EX 1**

**AS PETRI NET, VERSION 1**

**glucose1.speeded**
BIONETWORK, EX2, APOPTOSIS

APOPTOSIS IN MAMMALIAN CELLS

http://www.genomicObject.net
R1. SuSy: sucrose synthase
Sucre + UDP $\leftrightarrow$ UDPglc + Fructose

R2. UGPase: UDPglucose pyrophosphorylase
UDPglc + PP $\leftrightarrow$ G1P + UTP

R3. PGM: phosphoglucomutase
G6P $\leftrightarrow$ G1P

R4. FK: fructokinase
Fru + ATP $\rightarrow$ F6P + ADP

R5. PGI: phosphoglucone isomerase
G6P $\leftrightarrow$ F6P

R6. HK: hexokinase
Glc + ATP $\rightarrow$ G6P + ADP

R7. Inv: invertase
Suc $\rightarrow$ Glc + Fructose

R8. Glyc(b): glycolysis
F6P + 29 ADP + 28 Pi $\rightarrow$ 29 ATP

R9. SPS: sucrose phosphate synthase
F6P + UDPglc $\rightarrow$ S6P + UDP

R10. SPP: sucrose phosphate phosphatase
S6P $\rightarrow$ Suc + Pi

R11. NDPkin: NDP kinase
UDP + ATP $\leftrightarrow$ UTP + ADP

R12. SucTrans: sucrose transporter
eSuc $\rightarrow$ Suc

R13. ATPcons(b): ATP consumption
ATP $\rightarrow$ ADP + Pi

R14. StaSy(b): starch synthesis
G6P + ATP $\rightarrow$ starch + ADP + PP

R15. AdK: adenylate kinase
ATP + AMP $\leftrightarrow$ 2 ADP

R16: PPase: pyrophosphatase
PP $\rightarrow$ 2 Pi
WHY PETRI NETS?

- a suitable intermediate representation for different (specification/programming) languages, different phases of software development cycle, different validation methods;

- modelling power
  partial order (true concurrency) semantics applicable on any abstraction level specification of limited resources possible

- analyzing power
  not restricted to reachability graph

- BUT: modelling power <-> analyzing power

- integration of qualitative and quantitative analyses

INTEGRATION OF QUALITATIVE & QUANTITATIVE ANALYSES

net-based testing and monitoring

net-based qualitative analysis

net-based quantitative analysis
MODEL CLASSES

PETRI NETS

PLACE/TRANSITION

PETRI NET

(COLOURED PN)

context checking by Petri net theory

verification by temporal logics

TIME-DEPENDENT PN

NON-STOCHASTIC

PETRI NET

worst-case evaluation

STOCHASTIC

PETRI NET

performance prediction

reliability prediction

TOOL OVERVIEW

❑ Snoopy

design / animation / simulation of Petri nets, e.g.
QPN - XPN - SPN - XSPN - CPN - HPN,
and the coloured counterparts,
... and many more ...
special features
logical places / transitions
macro transition / places

❑ Charlie

standard Petri net analysis techniques, e.g.
structural properties
P/T-invariants
Siphon/Trap Property, rank theorem
reachability/coverability graph
(explicit) CTL model checking

❑ Marcie

QPN - symbolic CTL model checking
SPN - symbolic CSL model checking,
XSPN - simulative PLTLc model checking
CASE STUDIES

ACADEMIC:

- botanical garden
- low-level mutex algorithm
- Dijkstra’s philosophers
- Milner’s scheduler
- solitaire
- . . .

MORE REALISTIC

- production cell
- concurrent pushers
- cruise control
- . . .

REFERENCES

Snoopy

G Czichy (1993)  
Design and Implementation of a graphical editor for hierarchical Petri net models (in German);  

R TieDEMANN (1997)  
PED - Hierarchical Petri Net Editor, Manual (in German);  

T MENZEL (1996)  
Design and Implementation of a Petri Net Tool Kit Framework Integrating Animation and Simulation (in German);  
BTU Cottbus, Dep. of CS, Major Individual Project, 1996.

M Fieber (2004)  
Design and Implementation of a Generic and Adaptive Graph Tool (in German),  

M Heiner, R Richter, M Schwarick (2008)  
Snoopy - A Tool to Design and Animate/Simulate Graph-Based Formalisms;  

C Rohr, W Marwan, M Heiner (2010)  
Snoopy - a unifying Petri net framework to investigate biomolecular networks;  

M Heiner, M Herajy, F Liu, C Rohr, M Schwarick (2012)  
Snoopy – a unifying Petri net tool;  

Fei Liu (2012)  
Colored Petri Nets for Systems Biology;  

M Herajy (2013)  
Computational Steering of Multi-Scale Biochemical Networks;  

S Laarz (2013)  
Scalable Petri nets in Snoopy (in German);  
REFERENCES II

Charlie

PH Starke, S Roch (1997)
INA - Integrated Net Analyser version 1.7:

M Schwarick (2006):
A Tool to analyse Petri net models (in German);

A Fischer (2009)
Reachability graph analysis of time-dependent Petri nets (in German);

A Franzke (2009)
Charlie 2.0 - a multi-threaded Petri net analyzer,

J Wegener, M Schwarick, M Heiner (2011)
A Plugin System for Charlie;
Proc. CSP 2011, Białystok University of Technology, 531-554, September 2011.

Marcie

A Noack (1999)
A ZBBD Package for Efficient Model Checking of Petri Nets (in German);

A Tovchigrechko (2008)
Efficient symbolic analysis of bounded Petri nets using Interval Decision Diagrams;

M Heiner, M Schwarick, A Tovchigrechko (2009)
DSSZ-MC – A Tool for Symbolic Analysis of Extended Petri Nets;

M Schwarick, A Tovchigrechko (2010)
IDD-based model validation of biochemical networks;
Theoretical Computer Science, July 2010.

M Schwarick, C Rohr, M Heiner (2011)
MARCIE - Model checking And Reachability analysis done efficienTly;

M Heiner, C Rohr, M Schwarick (2013)
MARCIE - Model checking And Reachability analysis done efficienTly;

M Schwarick (2014)
Symbolic on-the-fly analysis of stochastic Petri nets;