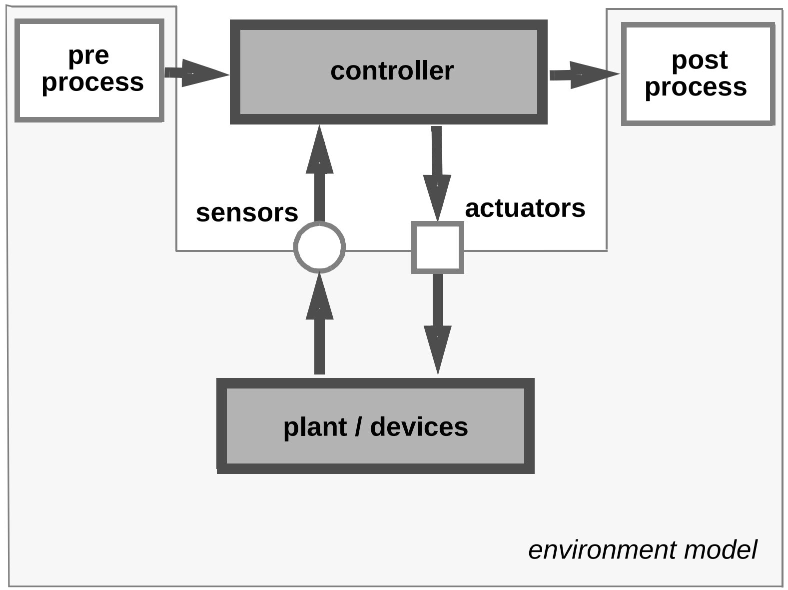
Brandenburg University
of Technology,
Computer Science Institute

PETRI NET BASED DEPENDABILITY ENGINEERING OF REACTIVE SYSTEMS

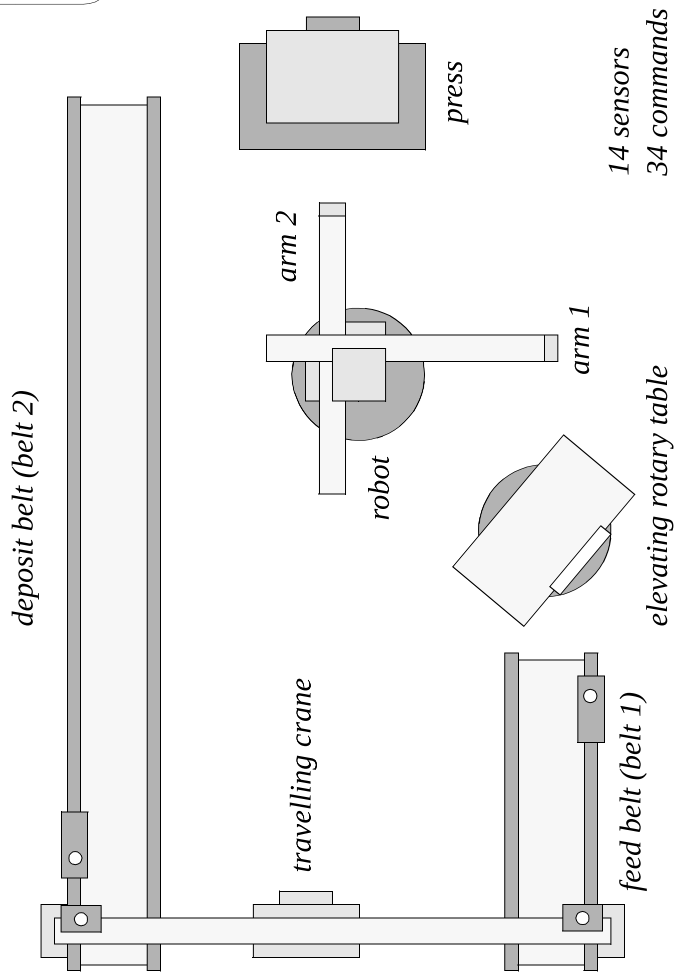
MONIKA HEINER

monika.heiner@b-tu.de
http://www.informatik.tu-cottbus.de

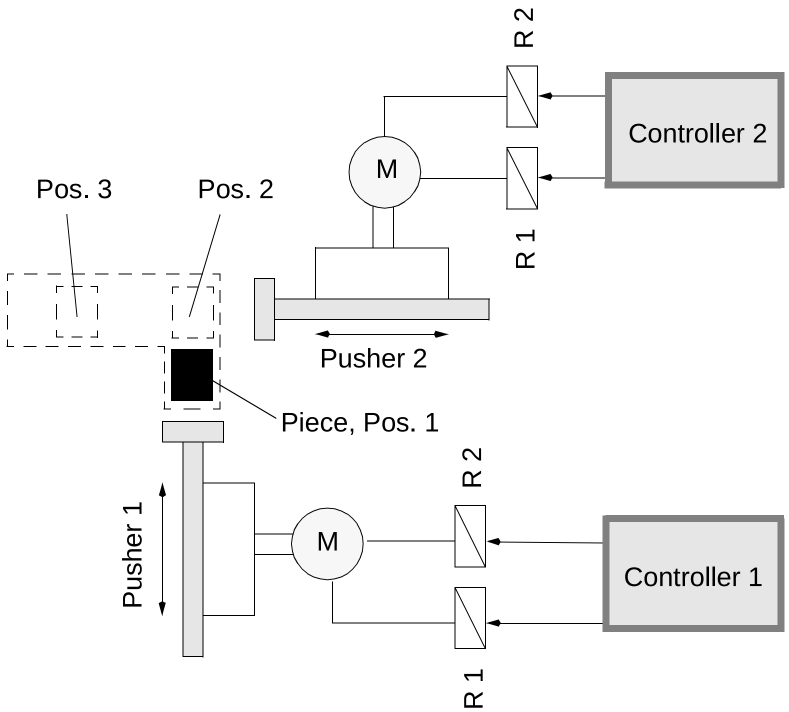
BASIC STRUCTURE OF REACTIVE SYSTEMS



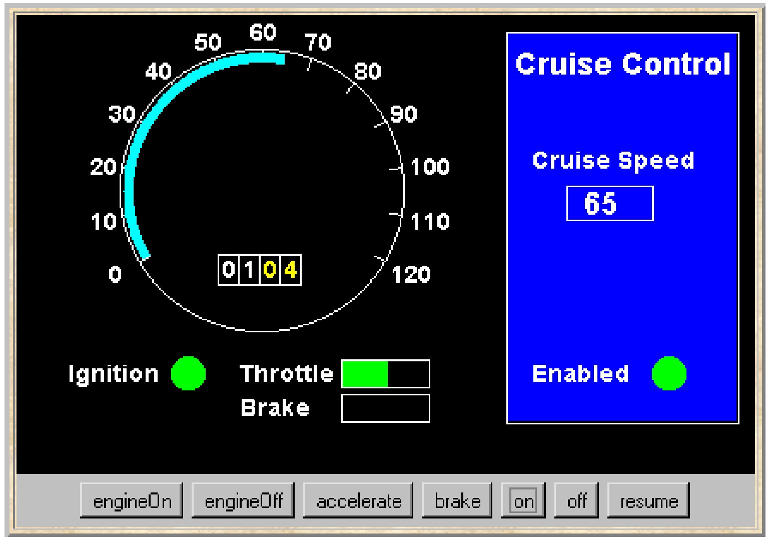
EXAMPLE, PRODUCTION CELL:



EXAMPLE, CONCURRENT PUSHERS



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

CONCURRENCY 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!

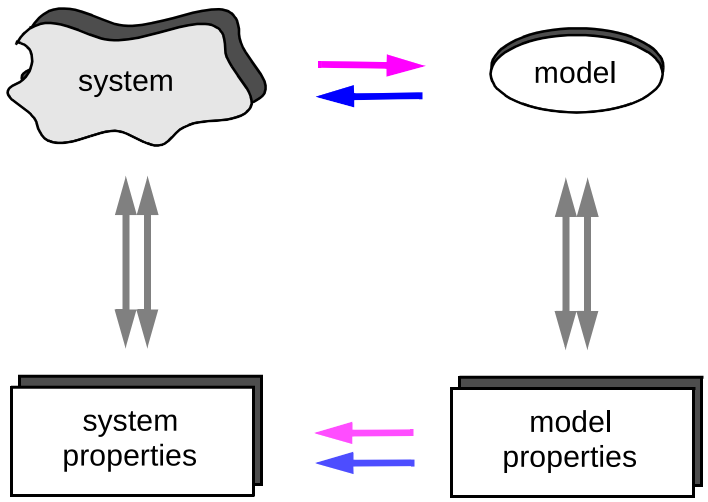
  ➡ dedicated models
  for different kinds of properties;

METHODS

SOFTWARE DEPENDABILITY

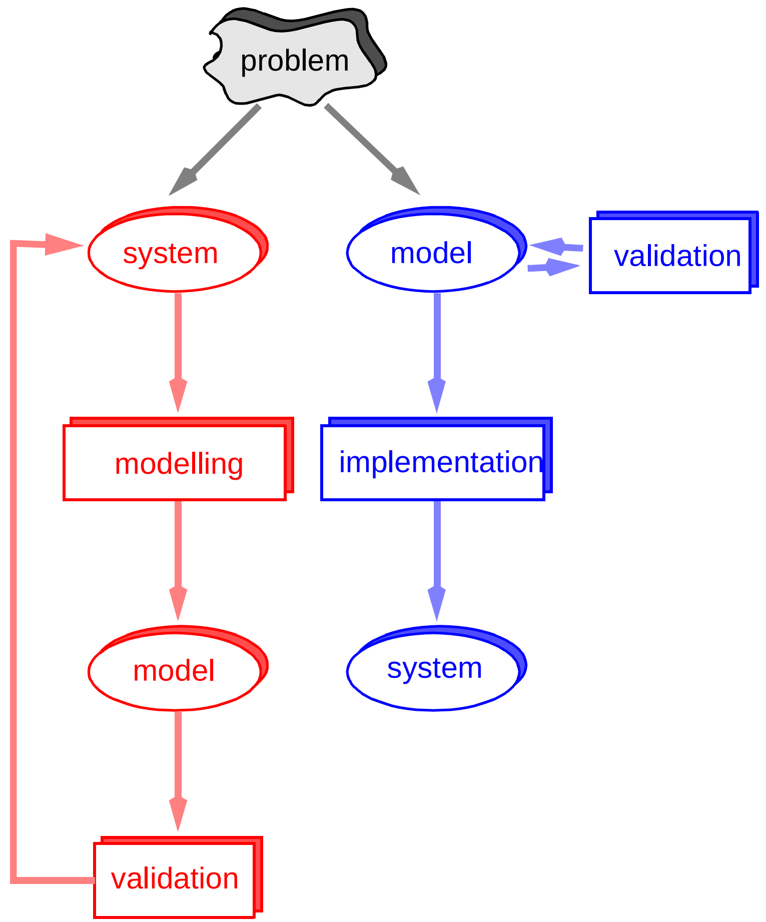
- FAULT AVOIDANCE - - -➤ *development phase*
  - FAULT PREVENTION
  - FAULT REMOVAL
    - MANUAL
    - COMPUTER-AIDED ➜ **VALIDATION**
      - animation / simulation / testing
      - context checking (static analysis)
      - consistency checking (verification)
- FAULT TOLERANCE - - -➤ *operation phase*
  - FAULT MASKING
    - DEFENSIVE
    - DIVERSITY
  - FAULT RECOVERY

MODEL BASED SYSTEM VALIDATION, GENERAL PRINCIPLE

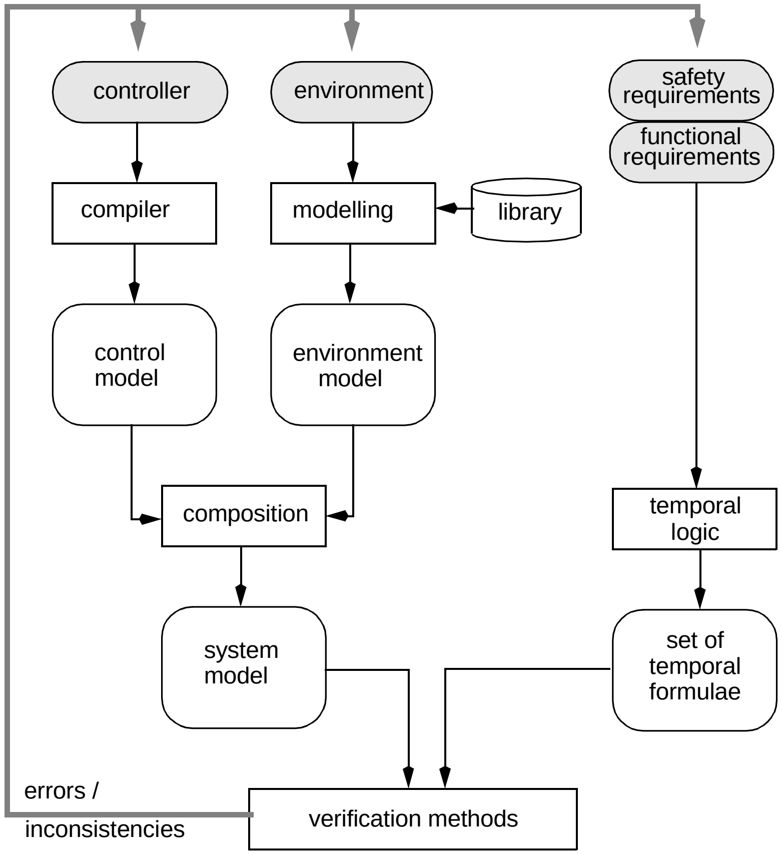


What was in the beginning ?

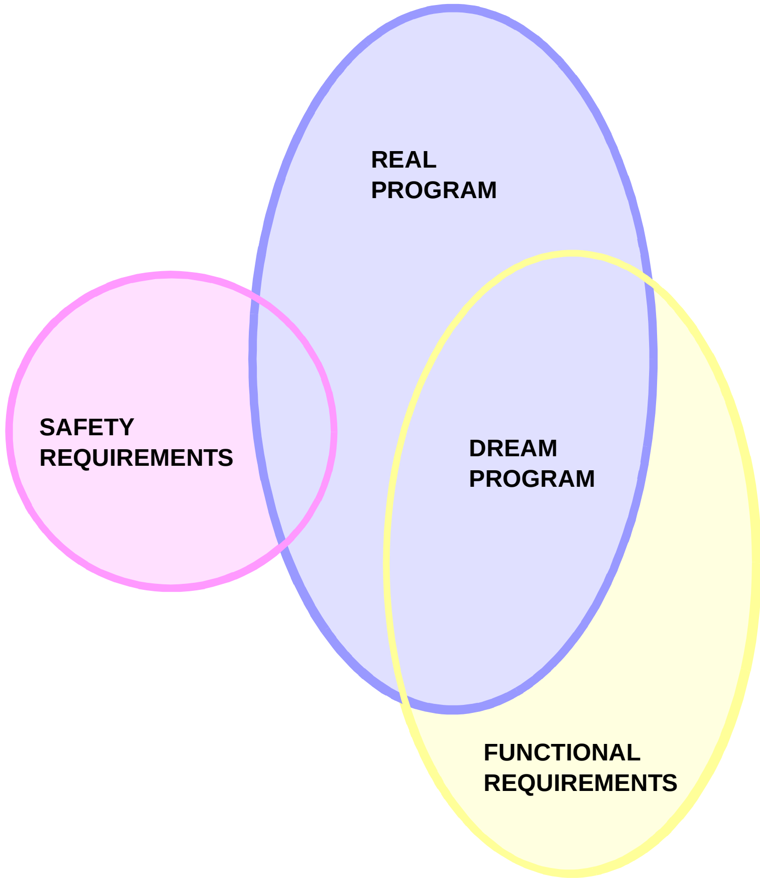
MODEL-BASED SYSTEM VALIDATION, TWO APPROACHES



A POSTERIORI MODEL BASED SYSTEM VALIDATION, PROCESS AND TOOLS



OBJECTTIVE - REUSE OF CERTIFIED COMPONENTS

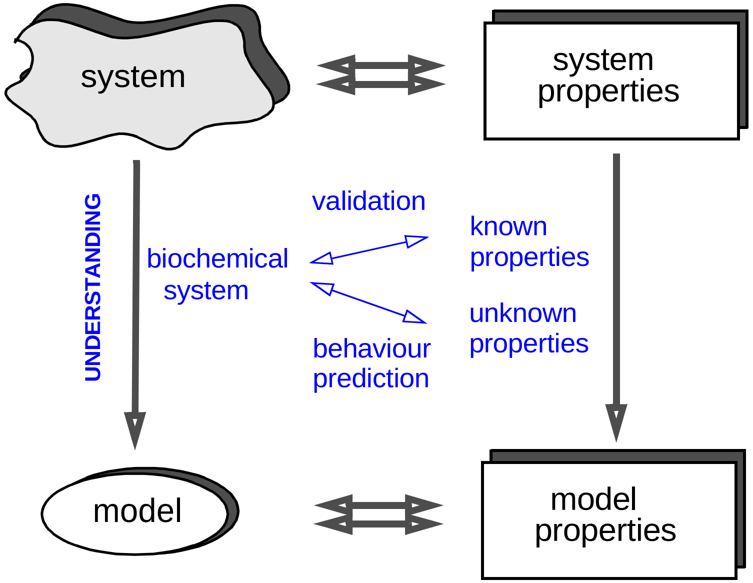


ANOTHER APPLICATION: BIOCHEMICAL SYSTEMS, EXAMPLES

- 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

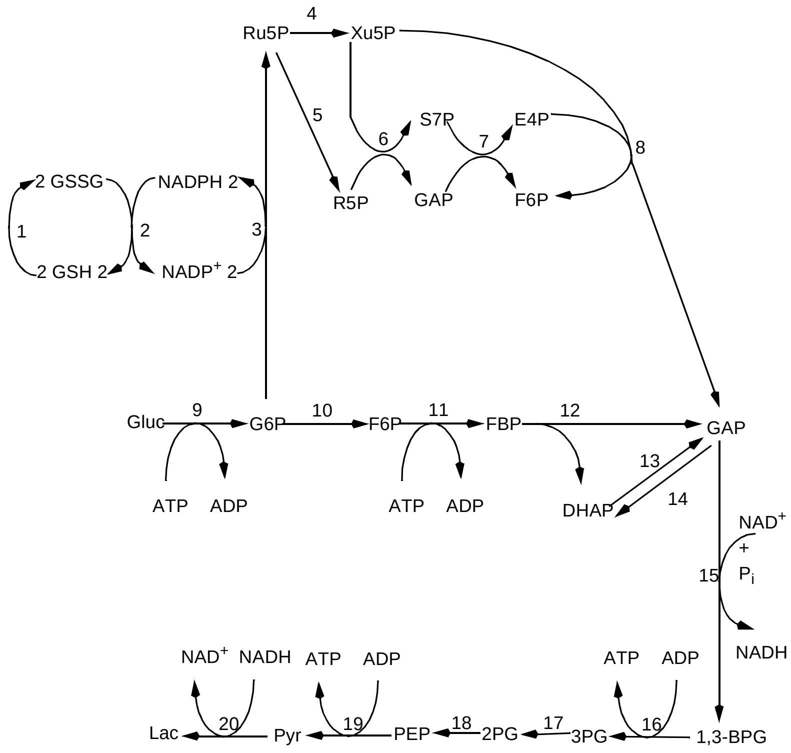
=>> networks of elementary actions

MODEL- BASED SYSTEM ENGINEERING



GENERALIZATION TO BIOCHEMICAL SYSTEMS

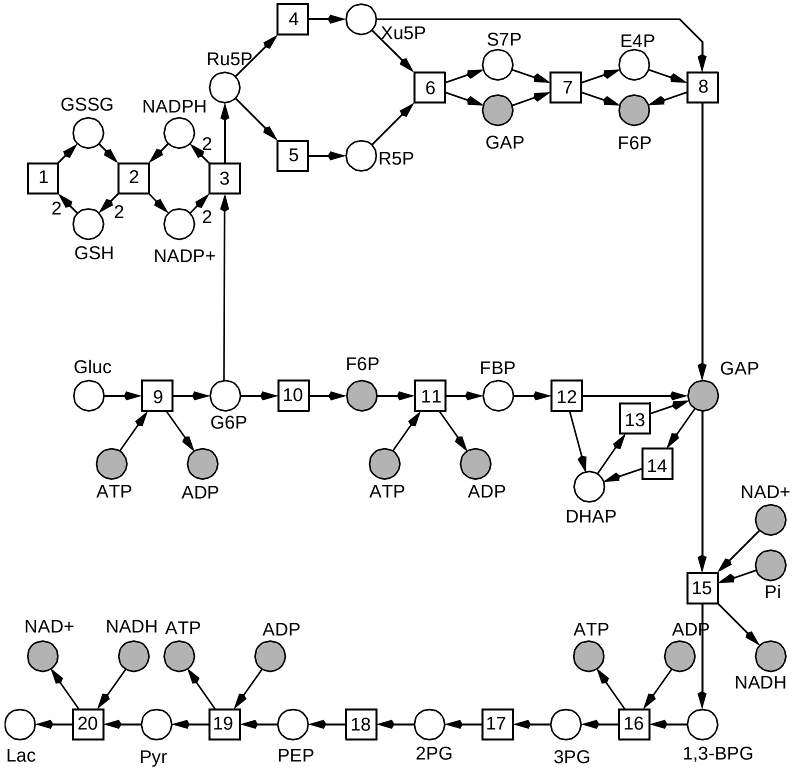
BIONETWORK, EX 1
G-PP PATHWAYS



GLYCOLYSIS / PENTOSE PHOSPHATE PATHWAYS IN ERYTHROCYTES

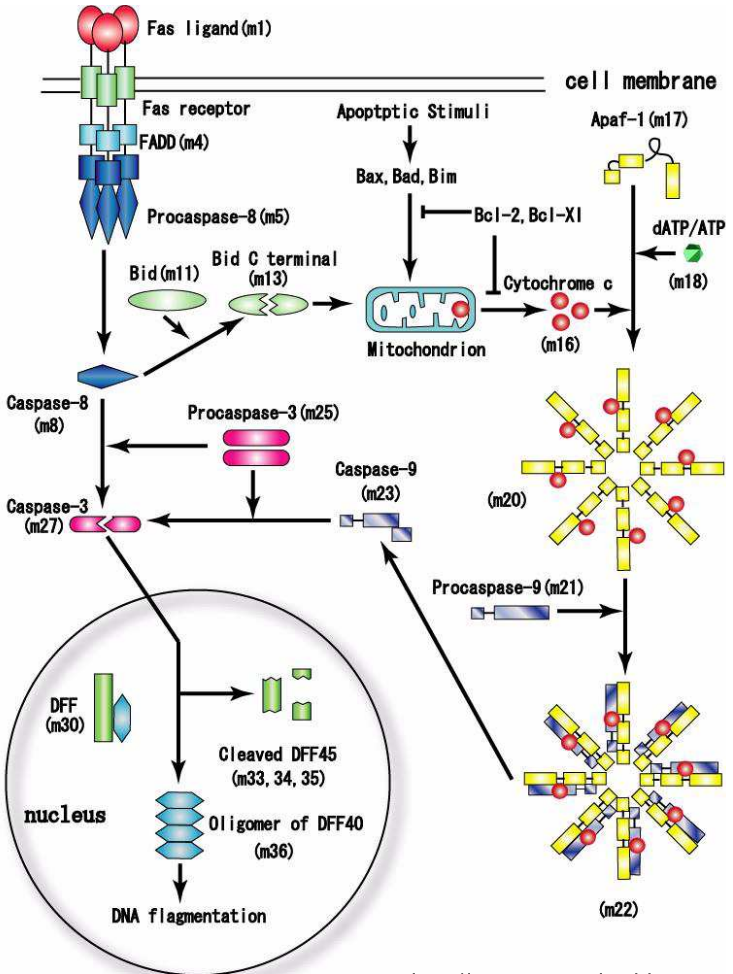
[Reddy 1996]

BIONETWORK, EX 1
AS PETRI NET, VERSION 1



glucose1.spped

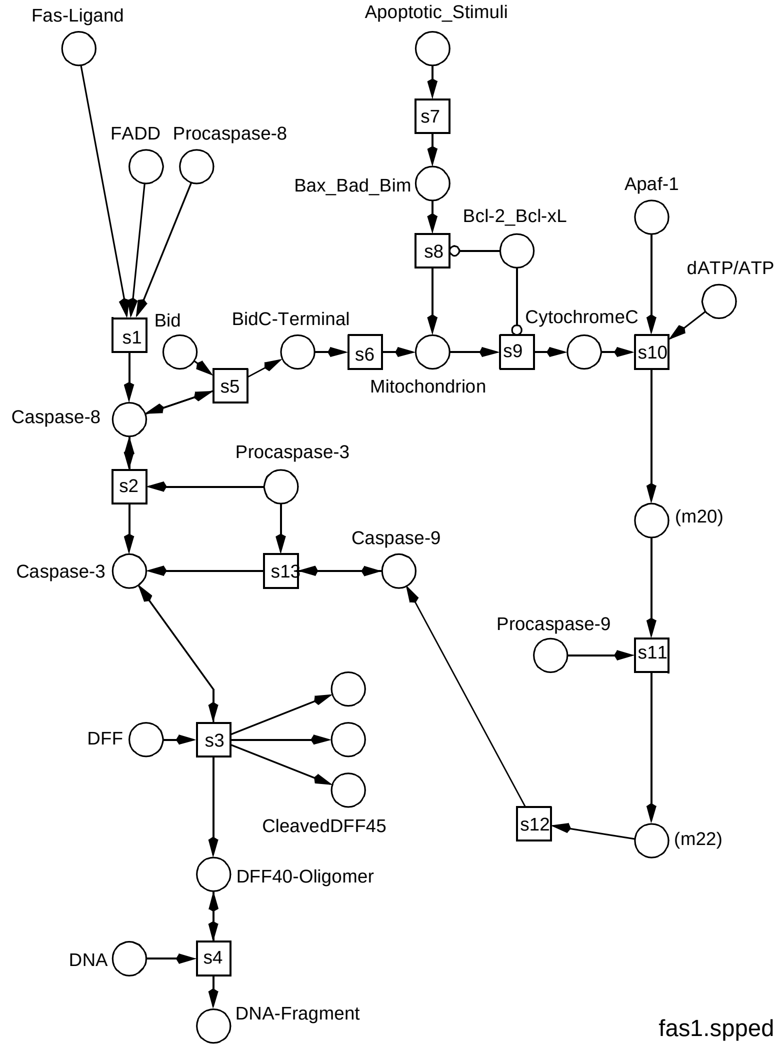
BIONETWORK, EX2, APOPTOSIS



http://www.genomicObject.net

APOPTOSIS IN MAMMALIAN CELLS

BIONETWORK, EX2, AS PETRI NET, VERSION 1

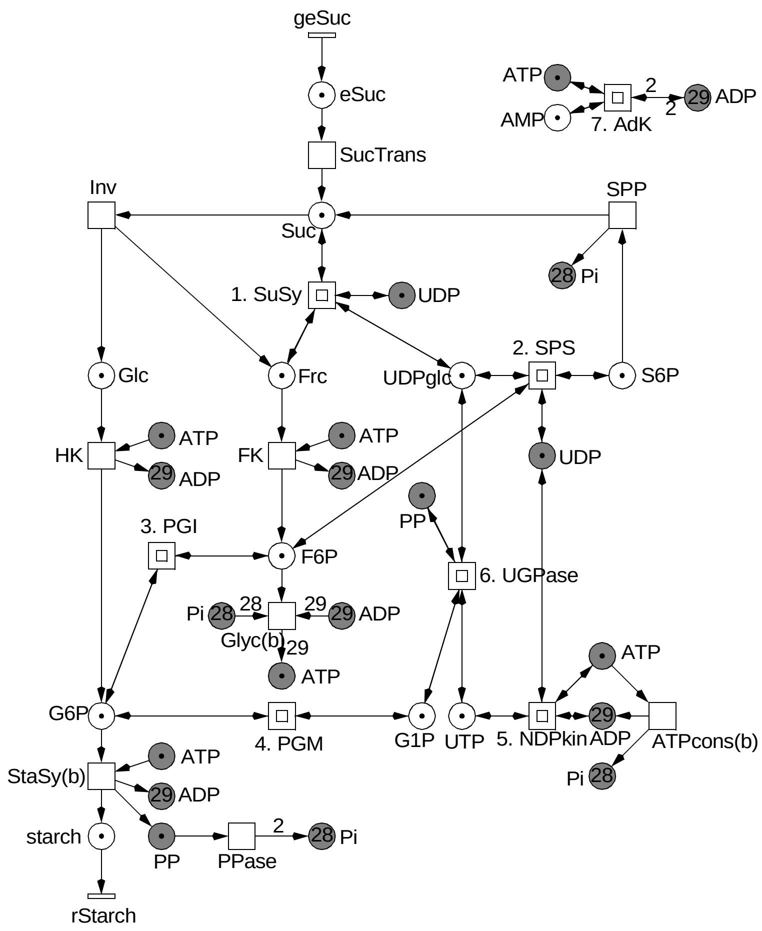


fas1.spped

BIONETWORK, EX3, POTATO TUBER

R1. SuSy:	*sucrose synthase*
	Suc + UDP <--> UDPglc + Frc
R2. UGPase:	*UDPglucose pyrophosphorylase*
	UDPglc + PP <--> G1P + UTP
R3. PGM:	*phosphoglucomutase*
	G6P <--> G1P
R4. FK:	*fructokinase*
	Frc + ATP --> F6P + ADP
R5. PGI:	*phosphoglucose isomerase*
	G6P <--> F6P
R6. HK:	*hexokinase*
	Glc + ATP --> G6P +ADP
R7. Inv:	*invertase*
	Suc --> Glc + Frc
R8. Glyc(b):	*glycolysis*
	F6P + 29 ADP + 28 P_i --> 29 ATP
R9. SPS:	*sucrose phospahate synthase*
	F6P + UDPglc <--> S6P + UDP
R10. SPP:	*sucrose phosphate phosphatase*
	S6P --> Suc + P_i
R11. NDPkin:	*NDP kinase*
	UDP + ATP <--> UTP +ADP
R12. SucTrans:	*sucrose transporter*
	eSuc --> Suc
R13. ATPcons(b):	*ATP consumption*
	ATP --> ADP + P_i
R14. StaSy(b):	*starch synthesis*
	G6P + ATP --> starch + ADP + PP
R15. AdK:	*adenylate kinase*
	ATP + AMP <--> 2 ADP
R16: PPase:	*pyrophosphatase*
	PP --> 2 P_i

BIONETWORK, EX3, AS PETRI NET



potato.spped

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

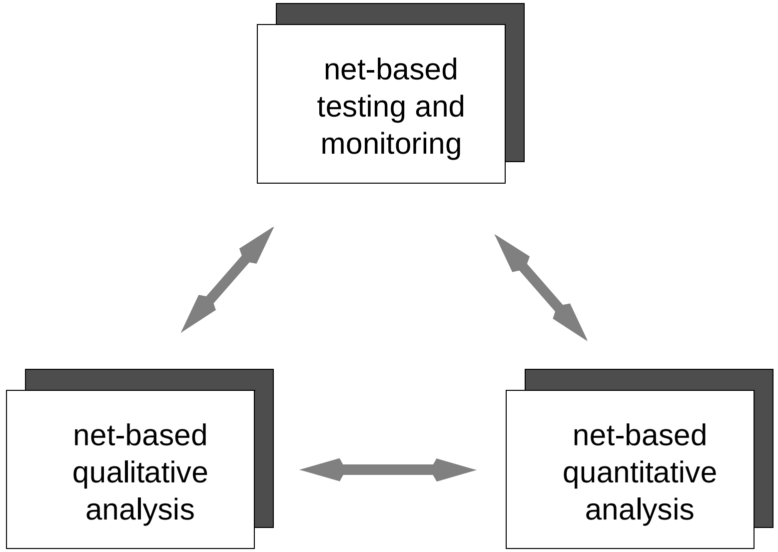
- ❑ analyzing power

  not restricted to reachability graph

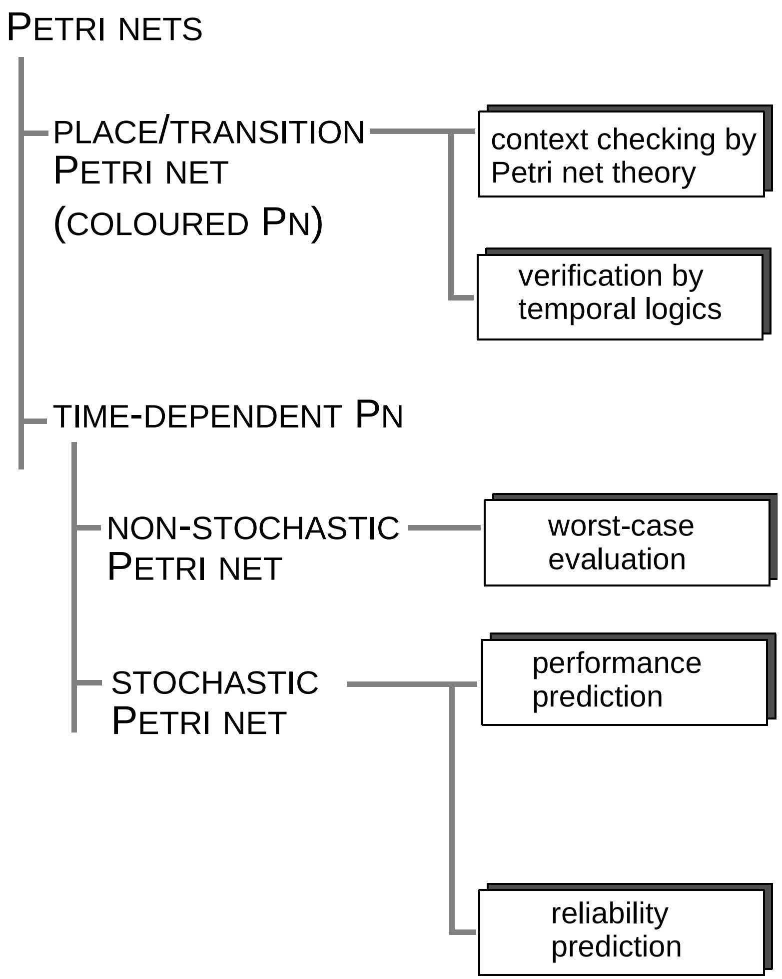
- ❑ **BUT:** modelling power <-> analyzing power

- ❑ integration of qualitative and quantitative analyses

INTEGRATION OF QUALITATIVE & QUANTITATIVE ANALYSES



MODEL CLASSES



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 I

Snoopy

G Czichy (1993)
Design and Implementation of a graphical editor for hierarchical Petri net models (in German);
TU Dresden, Dep. of CS, Master Thesis 1993.

R Tiedemann (1997)
PED - Hierarchical Petri Net Editor, Manual (in German);
BTU Cottbus, Dep. of CS, Internal Techn. Report, May 1997.

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),
BTU Cottbus, Dep. of CS, Master Thesis, July 2004

M Heiner, R Richter, M Schwarick (2008)
Snoopy - A Tool to Design and Animate/Simulate Graph-Based Formalisms;
Proc. PNTAP 2008, associated to SIMUTools 2008, ACM digital library, 2008.

C Rohr, W Marwan, M Heiner (2010)
Snoopy - a unifying Petri net framework to investigate biomolecular networks;
Bioinformatics 26(7):974-975, 2010.

M Heiner, M Herajy, F Liu, C Rohr, M Schwarick (2012)
Snoopy – a unifying Petri net tool;
Proc. PETRI NETS 2012, Hamburg, Springer, LNCS 7347, 398-407, June 2012.

Fei Liu (2012)
Colored Petri Nets for Systems Biology;
PhD thesis, BTU Cottbus, Dep. of CS, January 2012.

M Herajy (2013)
Computational Steering of Multi-Scale Biochemical Networks;
PhD thesis, BTU Cottbus, Dep. of CS, January 2013.

S Laarz (2013)
Scalable Petri nets in Snoopy (in German),
BTU Cottbus, Dep. of CS, Master Thesis, Februar 2013

REFERENCES II

Charlie

PH Starke, S Roch (1997)
INA - Integrated Net Analyser version 1.7;
Technical report, Humbold-Universität zu Berlin, 1997.

M Schwarick (2006):
A Tool to analyse Petri net models (in German);
BTU Cottbus, Dep. of CS, Master Thesis, September 2006.

A Fischer (2009)
Reachability graph analysis of time-dependent Petri nets (in German);
BTU Cottbus, Dep. of CS, Master Thesis, Oktober 2009.

A Franzke (2009)
Charlie 2.0 - a multi-threaded Petri net analyzer,
BTU Cottbus, Dep. of CS, Master Thesis, December 2009

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

Marcie

A Noack (1999)
A ZBBD Package for Efficient Model Checking of Petri Nets (in German);
BTU Cottbus, Dep. of CS, Major Individual Project, 1999.

A Tovchigrechko (2008)
Efficient symbolic analysis of bounded Petri nets using Interval Decision Diagrams;
PhD thesis, BTU Cottbus, Dep. of CS, October 2008.

M Heiner, M Schwarick, A Tovchigrechko (2009)
DSSZ-MC – A Tool for Symbolic Analysis of Extended Petri Nets;
Proc. PETRI NETS 2009, Paris, Springer, LNCS, volume 5606, pages 323–332, June 2009

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;
Proc. QEST 2011, Aachen, Germany, IEEE CS Press, pages 91–100, September 2011.

M Heiner, C Rohr, M Schwarick (2013)
MARCIE - Model checking And Reachability analysis done effiCIEntly;
Proc. PETRI NETS 2013, Milano, Springer, LNCS, volume 7927, pages 389–399, June 2013.

M Schwarick (2014)
Symbolic on-the-fly analysis of stochastic Petri nets;
PhD thesis, BTU Cottbus, Dep. of CS, June 2014.