

QUALITATIVE MODELLING AND ANALYSIS CONCURRENT SYSTEMS WITH PETRI NETS

Ina Koch

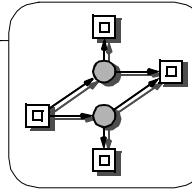
TFH BERLIN, BIOINFORMATICS

<http://www.tfh-berlin.de/bi/>

Monika Heiner

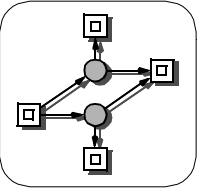
BTU COTTBUS, COMPUTER SCIENCE

<http://www-dssz.informatik.tu-cottbus.de>



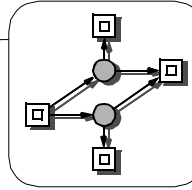
OUTLINE

1. MOTIVATION
2. INTRODUCTION INTO QUALITATIVE MODELLING
3. INTRODUCTION INTO QUALITATIVE ANALYSIS
PROPERTIES
REACHABILITY GRAPH
TRANSITION / PLACE INVARIANTS
4. SUMMARY, OUTLOOK
5. REFERENCES



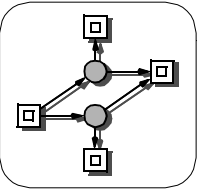
1.

MOTIVATION



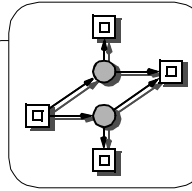
WHY PETRI NETS ?

- ❑ a suitable intermediate representation for
 - > *different languages*
 - > *different stages of certainty*
- ❑ modelling power
 - > *partial order semantics*
 - > *applicable on any abstraction level*
 - > *specification of limited resources possible*
- ❑ analysis power
 - > *various complementary analysis methods*
 - > *reliable tools*
- ❑ integration of qualitative and quantitative analyses
- ❑ **BUT:**
modelling power <-> analysis power

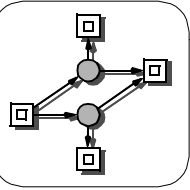


PETRI NETS, A BIT OF HISTORY

- ❑ Carl Adam Petri, 1962,
PhD University of Technology Darmstadt
-> *basic ideas introduced*
- ❑ early 1970's
-> *first papers contributing to Petri net theory*
- ❑ Petri, 1976
-> *application to chemical networks mentioned*
- ❑ early 1980's
-> *first monographs appear*
- ❑ Reddy, 1993
-> *first paper on bio application*
- ❑ late 1990's
-> *increasing interest in applying Petri nets for
modelling and analysis of bio networks*



2. INTRODUCTION INTO QUALITATIVE MODELLING



PETRI NETS, STRUCTURE

two types of nodes

-> **places**

"passive elements", conditions,
local states, chemical compounds

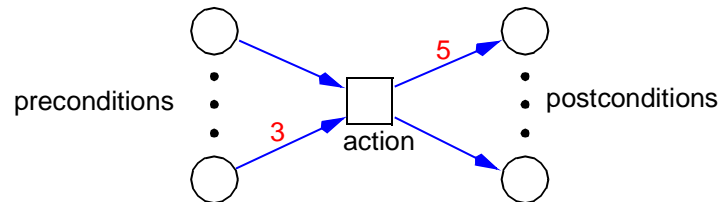


-> **transitions**

"active elements", events, actions,
chemical reactions



arcs



-> *directed*

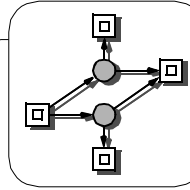
-> *never arcs between nodes of the same type*

-> *for any node,
arbitrary number of pre-nodes and post-nodes*

arc inscriptions

-> *arc weight / multiplicity*

-> *amount of units of the substances
involved in the basic (re-) action*



PETRI NETS, SYSTEM STATE

tokens

-> *moving objects,
e. g. units of substances (e. g. Mole), ...*

○ condition is not fulfilled

● condition is (one times) fulfilled

○_n condition is n times fulfilled

-> *token amount -
amount of available units of a given compound*

marking

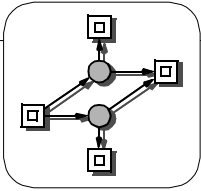
-> *How many tokens are on each place?*

-> *system state*

-> *substance distribution*

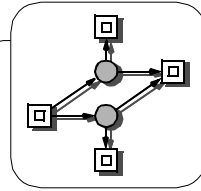
-> *initial marking*

-> *initial substance distribution*

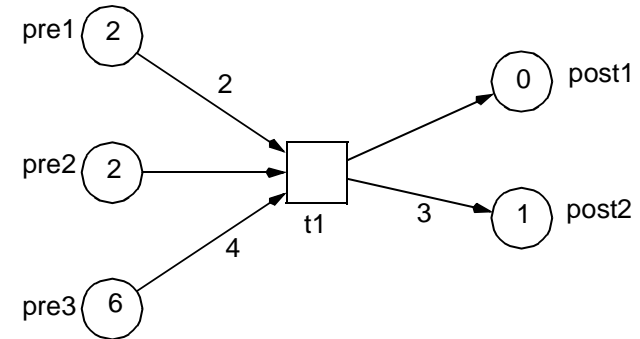


PETRI NETS, BEHAVIOUR

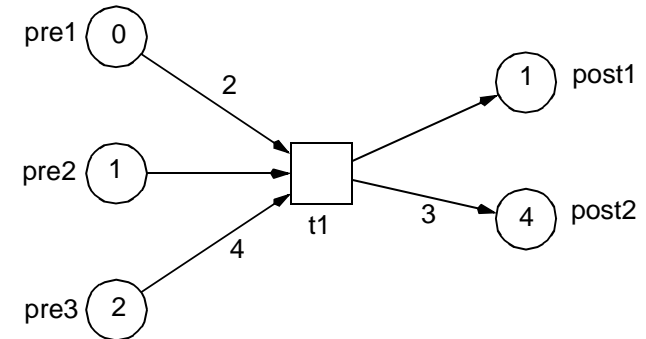
- flow of tokens
-> defined by firing rule
- an action **may** happen (fire), if
-> all preconditions are fulfilled (corresponding to the arc weights);
- **if** an action happens (fires), **then**
-> tokens are removed from all preconditions (corresponding to the arc weights), **and**
-> tokens are added to all postconditions (corresponding to the arc weights);
- an action happens (firing of a transition)
-> **atomic**
-> **time-less**



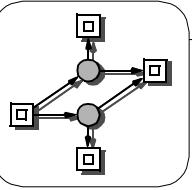
FIRING RULE, EXAMPLE 2



t1 fires



fire2.spped

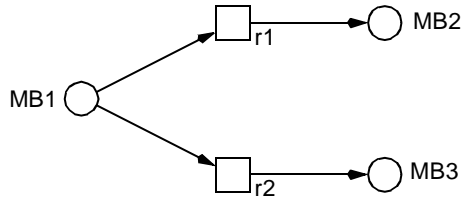


TYPICAL BASIC STRUCTURES

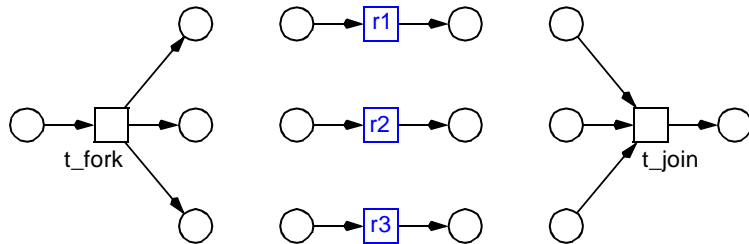
CHAIN OF REACTIONS



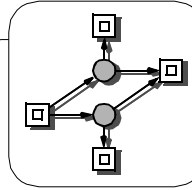
(FREE-CHOICE) BRANCHING / CONFLICT



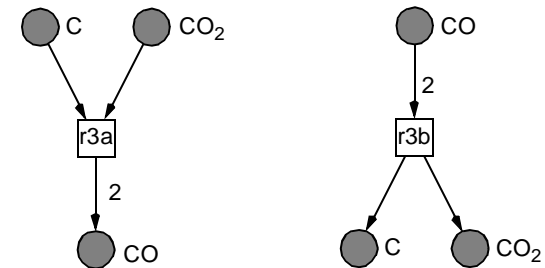
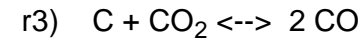
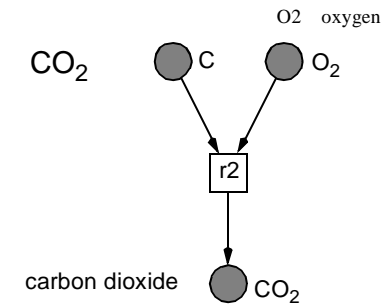
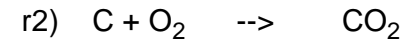
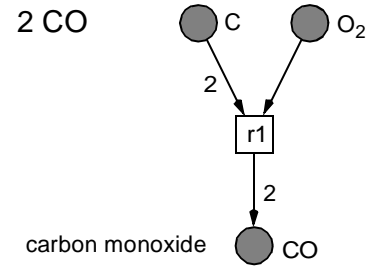
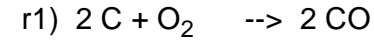
CONCURRENCY



r1, r2, r3 are concurrent = independent

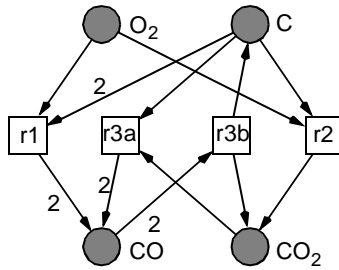


EXAMPLE, CARBON OXIDATION, BASIC REACTIONS

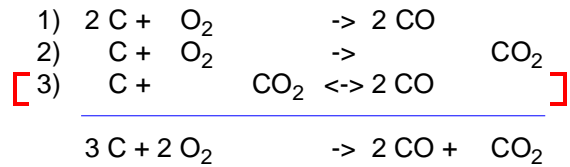


EXAMPLE, COMPOSITION

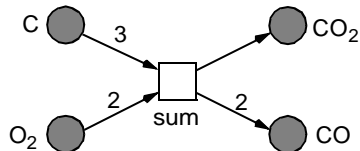
BASIC MODEL



SYSTEM'S TOTAL EQUATION



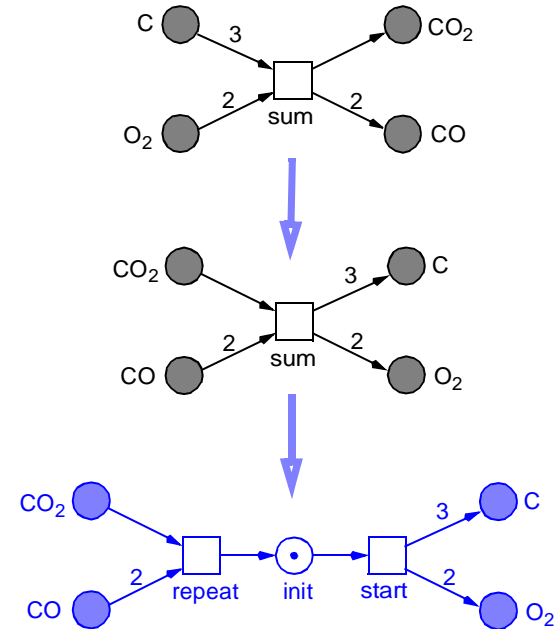
MODEL OF THE SYSTEM'S TOTAL EQUATION



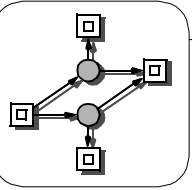
ENVIRONMENT BEHAVIOUR

- strong assumptions about quantitative relations of input / output compounds

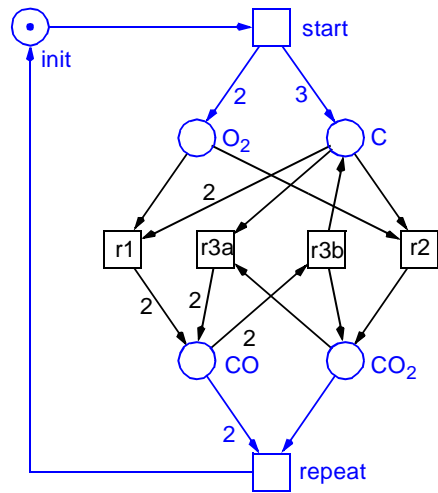
- 'inverse' total equation



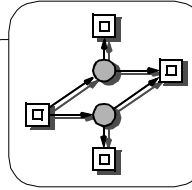
- there are no boundary nodes



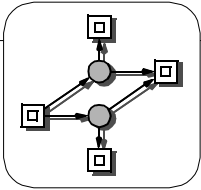
CARBON OXIDATION, SYSTEM MODEL,



carbon2.spped



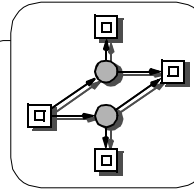
3. INTRODUCTION INTO QUALITATIVE ANALYSIS



PETRI NET PROPERTIES, OVERVIEW / INA

1. SIMPLE STRUCTURAL PROPERTIES

- ORD** ordinary (*1-multiplicity of all arcs*)
- HOM** homogeneous (*all output arcs of a given place have the same multiplicity*)
- NBM** non-blocking multiplicity (*for each place applies: MIN multiplicity of input arcs \geq MAX multiplicity of output arcs*)
- PUR** pure (*no side conditions*)
- CSV** conservative (*any firing preserves token amount*)
- SCF** static conflict free
- CON** connected
- SC** strongly connected
- Ft0** there is a transition without pre-place
- tF0** there is a transition without post-place
- Fp0** there is a place without pre-transition
- pF0** there is a place without post-transition
- MG** marked graph (*synchronization graph*)
- SM** state machine
- FC** free choice net
- EFC** extended free choice net
- ES** extended simple net



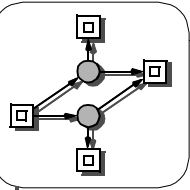
PETRI NET PROPERTIES, OVERVIEW / INA

2. MORE EXPENSIVE STRUCTURAL PROPERTIES

- DTP** deadlock trap property
- SMC** state machine coverable (*covered with SM components*)
- SMD** state machine decomposable (*covered with SCSM components*)
- SMA** state machine allocatable
- CPI** covered with place invariants
- CTI** covered with transition invariants
- SB** structurally bounded

3. BEHAVIOURAL PROPERTIES

- B** bounded
- REV** reversible (*the initial state m_0 can be reached again from all reachable states: home state*)
- DSt** dead states (*a state where no transition is enabled*)
- BSt** bad states (*a state where a fact is enabled*)
- DTr** dead transitions (*at the initial state*)
- DCF** dynamically conflict free
- L** live
- LV** live, excepted transitions dead at the initial marking
- L&S** live & safe (*1-bounded*)



BEHAVIOURAL NET PROPERTIES, OVERVIEW

MARKABILITY of places

- markable (*place liveness*)
- **k-bounded** (*1-bounded / safe*)
- unbounded

LIVENESS of transitions

- zero times firing (*m_0 -dead*)
- finite times firing (*dead, non-live*)
- **infinite times firing, probably** (*live*)
- infinite times firing, definitely (*livelock free*)

general semantic properties

REACHABILITY of states

- dead states
- reproducibility
- **reversibility** (*m_0 - home state*)
- bad states (*facts*)
- user-specified states

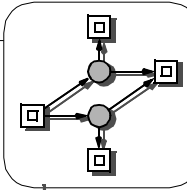
NET INVARIANTS

- **transition invariants**
- **place invariants**

special semantic properties

temporal relationship of logic formulae

- safety properties
- progress properties



QUALITATIVE ANALYSIS METHODS, OVERVIEW

NET REDUCTION

STRUCTURAL PROPERTIES

LINEAR PROGRAMMING

- place / transition invariants
- state equation
- trap equation

static analysis

STATE SPACE ANALYSIS

- (complete) reachability graph

compressed state spaces

- BDDs, NDDs, ..., xDDs
- Kronecker products

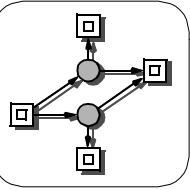
dynamic analysis

reduced state spaces

- coverability graph
- symmetry
- stubborn sets

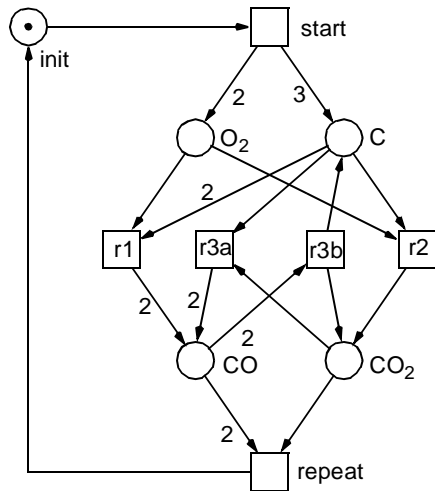
(model checking)

branching process

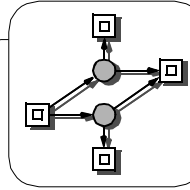


REACHABILITY GRAPH (RG)

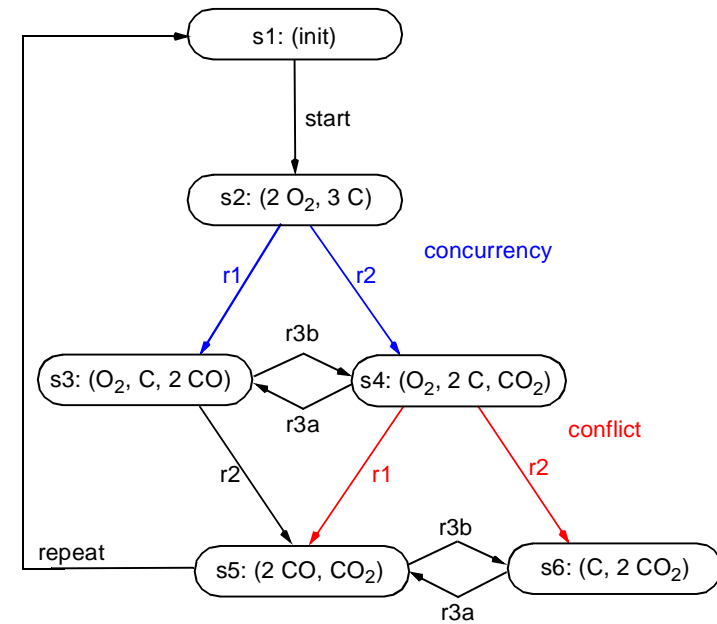
- nodes - system states
- arcs - the (single) firing transition
- example - carbon oxidation, environment style 3



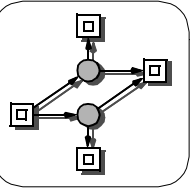
carbon2.ssped



RG (CARBON OXIDATION)

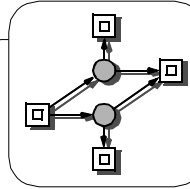
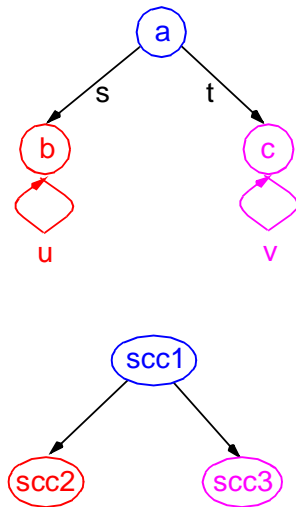
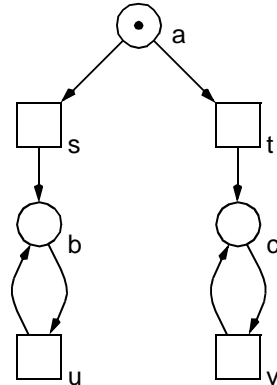


-> interleaving description
of the whole system behaviour



EXAMPLE: RG AND THREE BASIC PN PROPERTIES

- no concurrency
-> $rg(pn) == pn$
- rg - finite
-> bounded pn
- rg - not sc
-> pn not reversible
- no dead states, but liveness?
- condensed rg
node - sc component (scc)
scc:
maximal set of sc nodes;
a terminal scc
-> possible terminal system behaviour
-> must contain all transitions in a live pn
- not all terminal scc contain all transitions
-> the pn is not live



BASIC PROPERTIES & RG, SUMMARY

- How many tokens may reside at most in a given place . . .
-> $(0, 1, k, \infty)$?

-> boundedness

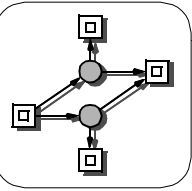
-> rg is finite
- How often may a transition fire . . .
-> $(0\text{-times}, n\text{-times}, \infty\text{-times})$?

-> liveness

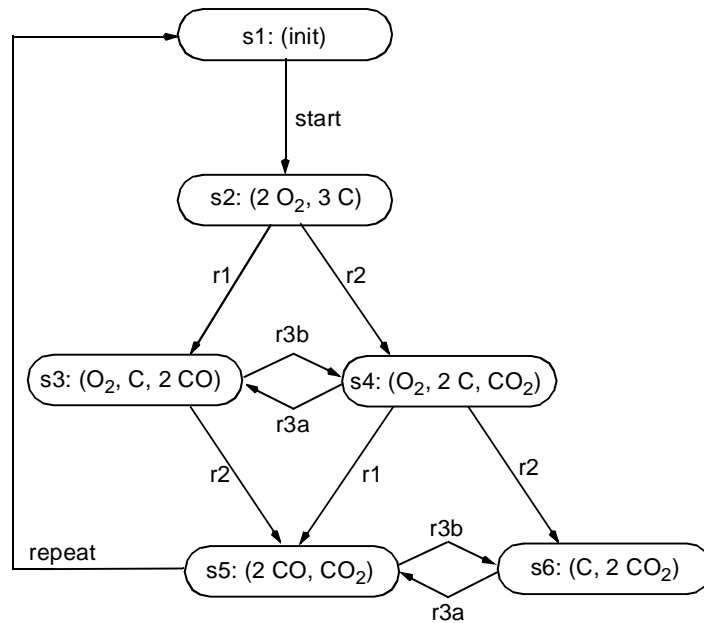
-> every terminal scc contains all transitions
- Is the initial system state . . .
-> always reachable again ?

-> reversibility

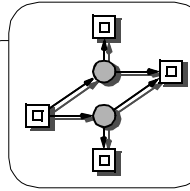
-> rg is sc (consists of one scc)



RG(CARBON OXIDATION), EVALUATION



- ❑ RG is finite
-> *BND*
- ❑ 1 Strongly Connected Component (SCC)
-> *REV*
- ❑ the only SCC contains all transitions
-> *LIVE*



REACHABILITY GRAPH, CONSTRUCTION ALGORITHM

PROCEDURE rg (IN Net pn , IN Marking m_0 ,
OUT MSet $nodes$, OUT ArcSet $arcs$);

MSet $U = \{m_0\}$, // unprocessed markings
 $N = \emptyset$; // rg nodes
 ArcSet $E = \emptyset$; // rg arcs (pre, post, t)
 Marking m' ; // successor marking
 Transition t ;

WHILE $U \neq \emptyset$ **DO**

choose one $m \in U$;
 $U = U - \{m\}$; $N = N \cup \{m\}$;

FOR ALL t enabled at m **DO**

$m' = m + \Delta t$;
IF $m' \notin N \cup U$ // new marking

THEN $U = U \cup \{m'\}$

ENDIF;

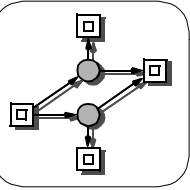
$E = E \cup \{(m, m', t)\}$

ENDFOR

ENDWHILE;

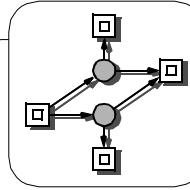
$nodes = N$; $arcs = E$;

ENDPROC rg.

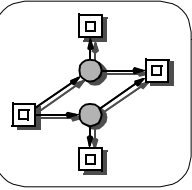


REACHABILITY GRAPH, OBSERVATIONS

- ❑ **unbounded** Petri net
 - > *the RG is **infinite***
- bounded** Petri net
 - > *the RG is **finite***
- ❑ simple construction algorithm
 - > *single step firing rule*
- ❑ concurrency
 - > *enumeration of all interleaving sequences*
- ❑ branching arcs in the RG
 - > *conflict **OR***
 - > *concurrency*
- ❑ RG tend to be very large
 - > *automatic evaluation necessary*
- ❑ **worst case: over-exponential growth**
 - > *alternative analyses techniques ?*



5. SUMMARY AND OUTLOOK



MODEL CLASSES

PETRI NETS

PLACE/TRANSITION
PETRI NET
(COLOURED PN)

validation by
Petri net theory

validation/prediction
by model checking

TIME-DEPENDENT PN

DISCRETE^{*)}
PETRI NET

worst-case
evaluation

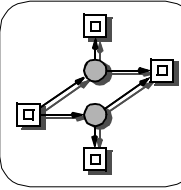
CONTINUOUS
PETRI NET

behaviour
prediction

STOCHASTIC
PETRI NET

reliability
prediction

*) DISCRETELY TREATABLE



REFERENCES

PETRI NETS, THEORY

Baumgarten, B.:

Petri Nets, Principles and Applications (in German);
Wissenschaftsverlag 1990.

Lautenbach, K.:

Exact Liveness Conditions of a Petri Net Class (in German);
Berichte der GMD 82, Bonn 1973.

Murata, T.:

Petri Nets: Properties, Analysis and Applications;
Proc. of the IEEE 77(1989)4, pages 541-580.

Petri, C. A.:

Communication with Automata (in German);
PhD Thesis, Schriften des Rheinisch-Westfälischen Instituts für Instrumentelle
Mathematik an der Univ. Bonn, No. 2, 1962.

Petri, C. A.:

Interpretations of Net Theory;
GMD, Interner Bericht 75-07, 2nd improved edition December 1976.

Reisig, W.:

Petri Nets - An Introduction;
Springer 1982.

Starke, P. H.:

Analysis of Petri Net Models (in German);
B. G. Teubner 1990.

for more related books and papers see

<http://www.daimi.au.dk/PetriNets/> -> Petri Net Bibliography

PETRI NETS, TOOLS

Fieber, M.:

Design and Implementation of a Generic, Adaptive Tool to Handle Graphs (in
German);
Master Thesis, BTU Cottbus, CS Dep., Juli 2004.

Starke, P. H.:

INA - Integrated Net Analyzer, Manual;
<http://www.informatik.hu-berlin.de/lehrestuehle/automaten/ina/#manual>
Humboldt University Berlin, CS Dep., 1998.