DEPENDABILITY ENGINEERING WITH TIME-DEPENDENT PETRI NETS

("THE PROBLEM IS CHOICE")

CONTENTS

- motivation
- time-dependent Petri nets
  - overview
  - influence of time on qualitative properties
  - zero test
- worst-case evaluation with duration interval nets
  - counter example
  - structural compression of well-formed net parts
  - non-well-formed, but 1-bounded, acyclic, ...
  - general procedure
- safety analysis with interval nets
  - unreachability of explicit error states
  - example - concurrent pushers
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

WHICH KIND OF TIME MODEL? (1)

- atomic sequential program parts -> transitions
  -> time assigned to transitions

- as simple as possible
  -> timed nets, [Ramachandani 74]
  -> duration nets (D nets, DPN)

- duration nets
  -> constant times assigned to transitions
  -> token reservation
  -> firing consumes time
IMMEDIATE TRANSITIONS

- zero (insignificant) time consumption
- time deadlocks
  - time deadlock = state from which
    - no transient state is reachable
    - or: no state is reachable
      where the system clock is able to advance
  - infinitely many firings in zero times
  - inconsistent time constraints!

- How to avoid time deadlocks?
  - invariants?

HOW TO ANALYSE DURATION NETS?

- time is running
  - change of the fire rule
    - $pn$ $t$ may fire $\rightarrow$ $tpn$ $t$ must fire
    - single step $\rightarrow$ maximal step

- special case:
  - duration of all transitions = 1 time unit
  - reachability graph construction under the maximal step firing rule

- else: transformation into special case

\[ \text{duration of all transitions} = 1 \text{ time unit} \]
\[ \text{reachability graph construction under the maximal step firing rule} \]

\[ d > 2 \]
\[ \text{lock} \]
\[ \text{d-2} \]
THE INFLUENCE OF TIME
EXAMPLE 1 (SYSTEM DEADLOCK),
PETRI NET

different initial marking!

EXAMPLE 1
SYSTEM DEADLOCK,
MAX STEP RG = RG(DPN)

DSt (pn) -> not DSt (tpn)
EXAMPLE 1
SYSTEM DEADLOCK, REACHABILITY GRAPH

INIT STATE

DEAD STATE

RG (pn)
19 nodes, 32 arcs

RG (tpn)
6 nodes, 6 arcs

THE INFLUENCE OF TIME, EXAMPLE 2

not BND (pn) -> BND (tpn)
not DTr (pn) -> DTr (tpn)
EXAMPLE 2, COVERABILITY GRAPH

- not BND, simultaneously unbounded in m1 and m2
- LIVE

EXAMPLE 2, MAX STEP RG = RG(TPN)

- BND,
  - $\text{cycle time}(p) = 2$
  - $\text{cycle time}(s) = 2$
  - $\text{cycle time}(c) = 1$
- not LIVE
  - $\text{TSCC does not contain S\_wait\_m2}$
  - $\text{S\_wait\_m2 is m}_0\text{-dead}$
EXAMPLES, SUMMARY

- example 1
  -> $DSt\ (pn)\ \rightarrow\ not\ DSt\ (tpn)$

- example 2
  -> $not\ BND\ (pn)\ \rightarrow\ BND\ (tpn)$
  -> $not\ DTr\ (pn)\ \rightarrow\ DTr\ (tpn)$

- generally
  
  $$
  
  \begin{array}{c}
  \text{PN} \\
  \text{prop}(pn) \\
  \text{RG}\ (pn)
  \end{array} \\
  \xrightarrow{T\rightarrow\text{TIME}} \\
  \begin{array}{c}
  \text{TPN} \\
  \text{prop}(tpn) \\
  \text{RG}\ (tpn)
  \end{array}
  $$

- BUT,
  for Petri net based system validation,
  we are only interested in the conclusions

  $$
  \begin{array}{c}
  \text{prop}(pn) \\
  \text{??}
  \end{array} \rightarrow \\
  \begin{array}{c}
  \text{prop}(tpn)
  \end{array}
  $$

THE INFLUENCE OF TIME ON QUALITATIVE PROPERTIES

TIME-INSENSITIVE RESULTS

- $BND\ (pn)\ \rightarrow\ BND\ (tpn)$
- $not\ DSt\ (pn)\ \rightarrow\ not\ DSt\ (tpn)$
- $DTr\ (pn)\ \rightarrow\ DTr\ (tpn)$

TIME-SENSITIVE RESULTS

- $not\ BND\ (pn)\ \rightarrow\ BND\ (tpn)$
- $DSt\ (pn)\ \rightarrow\ not\ DSt\ (tpn)$
- $not\ DTr\ (pn)\ \rightarrow\ DTr\ (tpn)$

GENERALLY

- $\exists$ -properties: $\overline{\text{prop}}\ (pn)\ \rightarrow\ \overline{\text{prop}}\ (tpn)$
- $\forall$ -properties: $\text{prop}\ (pn)\ \leftarrow\ \text{prop}\ (tpn)$
**PROBE EFFECT**

- **observation** -
  the system exhibits in test mode other (less) behavior than in standard operation mode

- **cause** -
  sw test means (debugger) affect the timing behavior

- **result** -
  masking of certain types of system behavior / bugs
  
  - $D\text{St} (pn)$ --- $not \ D\text{St} (tpn)$
  
  - $live(pn)$ --- $not \ live (tpn)$
  
  - $not \ B\text{ND} (pn)$ --- $B\text{ND} (tpn)$

- **consequence** -
  systematic & exhaustive testing of concurrent systems is generally impossible

- **wayout** -
  qualitative models considering any timing behavior

---

**TIME-INARIANT NET STRUCTURES**

- **time-invariant == time independently live**

- **D nets** [Starke 90]
  
  $\rightarrow$ homogenous ES nets

  ![Diagram](image1)

- **generalization ?**
  
  $\rightarrow$ behavioral ES nets ?

- **troubblemaker** - confusing combination of channel and control flow conflicts

  ![Diagram](image2)

  $\rightarrow$ “The problem is choice!”
CONFUSION

- concurrency and conflict overlap
  - $t_1 \# t_2$ and $t_2 \# t_3$, but $t_1$ concurrent to $t_3$

- **case 1**: $t_1 < t_3$
  - $t_2 \# t_3$ disappears, firing of $t_3$ does not involve a conflict decision

- **case 2**: $t_3 < t_1$
  - $t_2 \# t_3$ exists, firing of $t_3$ involves a conflict decision

- the interleaving sequences of concurrency may encounter different amount of decisions

- an observer outside of the system does not know whether a decision took place or not

ARE THERE TIME-INvariant SOFTWARE STRUCTURES?
INFLUENCE OF COMMUNICATION PATTERNS ON NET STRUCTURE CLASSES

<table>
<thead>
<tr>
<th>addressing waiting\</th>
<th>direct / semi-direct-by-sender</th>
<th>indirect / semi-direct-by-receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>deterministic</td>
<td>EFC</td>
<td>ES</td>
</tr>
<tr>
<td>non-deterministic</td>
<td>ES</td>
<td>ICP</td>
</tr>
</tbody>
</table>

- simplified view
  -> provided, pre- and postprocesses do not access the same communication object from different control points

- known to be time-independently live [Starke 90]
  *i.e. a live net remains live under any constant delay timing.*

INFLUENCE OF COMMUNICATION PATTERNS ON CONFLICT STRUCTURES

<table>
<thead>
<tr>
<th>\addressing waiting</th>
<th>direct / semi-direct-by-sender</th>
<th>indirect / semi-direct-by-receive</th>
</tr>
</thead>
<tbody>
<tr>
<td>deterministic</td>
<td>no dynamic</td>
<td>channel &amp; control flow conflicts appear only separately</td>
</tr>
<tr>
<td>non-deterministic</td>
<td>channel conflicts</td>
<td>confusing combination of channel &amp; control flow conflicts possible</td>
</tr>
</tbody>
</table>
WHICH KIND OF TIME MODEL? (2)

- adequate characterization of time consumption
  - alternatives, iterations
  - time nets, [Merlin 74]
  - interval nets, I nets

- structural simplicity, e.g. alternative as
  - duration net (with token reservation)
    - (constant times)
    - (firing consumes time)
  - interval net (no token reservation)
    - (interval times)
    - (firing itself timeless)

- duration interval net, DI net
  - interval times
  - with token reservation
  - firing consumes time

<table>
<thead>
<tr>
<th>Non-stochastic T-Time-Dependent Petri Nets, Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>firing principle</strong></td>
</tr>
<tr>
<td><strong>times</strong></td>
</tr>
<tr>
<td>constant</td>
</tr>
<tr>
<td>timed nets [Ramchandani 74]</td>
</tr>
<tr>
<td>interval</td>
</tr>
<tr>
<td>time nets [Merlin 74]</td>
</tr>
</tbody>
</table>

D NETS

DI NETS

I NETS
RELATION OF TIME-DEPENDENT PETRI NETS (TRANSITION \rightarrow \text{TIME})

I NETS

D NETS \leftrightarrow DI NETS

DI NET -> I / D NET

\[
\begin{align*}
&\text{DI NET} \\
p_1 &\xrightarrow{t_1} [1,2] p_2 \\
&\text{I NET} \\
p_1 &\xrightarrow{t_1_{\text{free}}} [0,0] t_1_{\text{run}} \xrightarrow{t_1} [1,2] p_2 \\
&\text{D NET} \\
p_1 &\xrightarrow{t_1_{\text{free}}} [0,0] t_1_1, t_1_2 \xrightarrow{t_1_{\text{run}}} [1,2] t_1_1, t_1_2, t_2_{\text{free}} \xrightarrow{t_2} [3,5] p_2
\end{align*}
\]
ZERO TEST FOR TURING POWER

PETRI NET?

\[
\begin{align*}
\text{test} & \rightarrow p \rightarrow \text{tFalse} \\
\text{tTrue} & \rightarrow p\text{True} \\
\text{tFalse} & \rightarrow p\text{False}
\end{align*}
\]

I NET

\[
\begin{align*}
\text{test} & \rightarrow p \rightarrow \text{tFalse} \\
\text{tTrue} & \rightarrow p\text{True} \\
\text{tFalse} & \rightarrow p\text{False}
\end{align*}
\]

\[
\begin{align*}
\text{tTrue} & \rightarrow [1,1] \\
\text{tFalse} & \rightarrow [2,2]
\end{align*}
\]

D NET

\[
\begin{align*}
\text{p} & \rightarrow \text{t1} <1> \\
\text{t2} & \rightarrow <1> \\
\text{t3} & \rightarrow <1> \\
\text{t4} & \rightarrow <1>
\end{align*}
\]

\[
\begin{align*}
\text{pFalse} & \rightarrow \text{t1} <1> \\
\text{t2} & \rightarrow <1> \\
\text{t3} & \rightarrow <1> \\
\text{t4} & \rightarrow <1>
\end{align*}
\]

False: t1, t2, t4
True: t1, t2+t3, t5

TIME-IN Variant
NET STRUCTURES (I NETS)

A live Petri net remains live under any timing, [Popova 95]

- if it is persistent,
- if the earliest firing time of all transitions is zero,
- if the latest firing time of all transitions is infinite,
- if it is an homogeneous & timely homogeneous EFC without purely immediate transitions

not homogeneous not timely homogeneous

\[
\begin{align*}
\text{t1} & \rightarrow 1,3 \\
\text{t2} & \rightarrow 2,4
\end{align*}
\]

not homogeneous not timely homogeneous

\[
\begin{align*}
\text{t1} & \rightarrow 1,2 \\
\text{t2} & \rightarrow 2,4
\end{align*}
\]

- if it is an homogeneous & timely homogeneous behaviourally free choice net without purely immediate transitions.

\[
\begin{align*}
\text{AG} (\text{enabled}(t1) \Leftrightarrow \text{enabled}(t2))
\end{align*}
\]
WORST-CASE EVALUATION
WITH DI NETS,
INPUT PARAMETERS

- time consumption of sequential program parts
  - at least $l$ time units
    (lower bound of duration time, $low(t_{ij}) = l$)
  - at most $m$ time units
    (upper bound of duration time, $upp(t_{ij}) = m$)
  - or any (continuous) time in between
    measured by monitoring OR calculated from computer instructions

- no explicit branching probabilities

COMMUNICATION
TIME MODEL

- time to write into channel
- receiving
- time to read from channel
- transmission time
  (of communication medium)
WORST-CASE EVALUATION WITH DI NETS
OUTPUT PARAMETERS

- \( t_{\text{begin, end}} = [\text{min}, \text{max}] \)

- min execution duration (shortest path), max execution duration (largest path)

- esp. valuable for systems which require predictable timing behaviour (to meet given deadlines)

- calculations can be based on discrete reachability graph (only integer states)
  - \( \text{INA} \)

COMPUTATION OF MINIMAL PATH BY LOWER BOUNDS ONLY, COUNTER EXAMPLE

min_duration(dresden, weimar):
- D net with lower bounds only: 14
- DI net with lower and upper bounds: 7

-> maximal path by upper bounds only? (!)
COUNTER EXAMPLE AS D NET

dresden

berlin

dresden2leipzig

berlin2leipzig

take_fast_train

leipzig1

leipzig2

take_slow_train

in_slow_train

in_fast_train

slow_train

fast_train

weimar

troubemaker

overlapping time windows of
dresden2leipzig & berlin2leipzig

STRUCTURAL COMPRESSION OF WELL-FORMED NET STRUCTURES, EXAMPLE

init

par [2,3]

if

a1 [2,3]

a3 [1,5]

a4 [3,8]

endif

a2 [3,5]

a5 [1,1]

a6 [3,4]

loop

(0,5)

endloop

[0,0]

number of iterations
STRUCTURAL COMPRESSION OF WELL-FORMED NET STRUCTURES, EXAMPLE (CONT.)

$$t_{ik} = [a, b] \quad t_{kj} = [c, d]$$

$$t_{ij} = [a + c, b + d]$$

assumption:
$$\{\text{lower bound} \} \quad \text{bound} \quad \{\text{upper bound} \}$$

of iterations given
STRUCTURAL COMPRESSION
OF WELL-FORMED NET STRUCTURES,
GENERAL (CONT.)

\[ t_{ij} = \begin{bmatrix} \text{low}(t_{ij}), \text{low}(t_{i'j'}) + \text{low}(t_{jj}) + \text{low}(t_{ij}) \\ \text{upp}(t_{ij}), \text{upp}(t_{i'j'}) + \text{upp}(t_{jj}) + \text{upp}(t_{ij}) \end{bmatrix} \]

EXAMPLE - INTERVAL EVALUATION
OF NON-WELL-FORMED STRUCTURES,
BUT 1-BOUNDED, ACYCLIC, ...

\[ t_{ij'} = \begin{bmatrix} \text{max}(\text{low}(t_{i1j1}), \text{low}(t_{i2j2})), \\ \text{max}(\text{upp}(t_{i1j1}), \text{upp}(t_{i2j2})) \end{bmatrix} \]

[Reske 95, p. 92]

[18,46] cycle time
INTERVAL EVALUATION,
GENERAL PROCEDURE

- net structure transformation
  - [first state -> init state]
  - [last state -> dead state]
  - resolution of (unlimited) cycles, if any

- net type transformation
  - DI net -> I net or DI net -> D net;

- determine (set of) state numbers of
  - first state
  - last state
  - of the path to be measured;

- evaluation of
  - reachability graph OR ?
  - other descriptions of all possible behaviours
    - prefix of branching processes
    - concurrent automaton

MIN (pathes(init, any dead state));
MAX (pathes(init, any dead state));
ENVIRONMENT MODEL, WITH EXPLICIT ERROR STATES

CONCURRENT PUSHERS, (PART OF THE) REACHABILITY GRAPH

Remark:
Only the interesting parts of the markings are shown.

\[
\text{init, } R_1\text{ off, basic} \\
\vdots \\
\text{tr1; } R_1\text{ set on;}
\]

\[
\text{step1, } R_1\text{ on, ext} \\
\text{step2, } R_1\text{ on, ext} \\
\text{R1 set off; tr3}
\]

\[
\text{step3, } R_1\text{ off, ext}
\]

\[
\text{bad state!}
\]

\[
\text{bad state!}
\]

\[
\rightarrow \text{(preemptive) interval nets}
\]

unreachability of bad states, 
\( m_{\text{dead}}(\text{ext2far}) \) if:

\[
lft(tr2) < eft(\text{ext2far}) \land \\
lft(R1\text{ set off}) < eft(\text{ext2far})
\]
REFERENCES

[Bause 96]  
Bause, F.; Kritzinger, P. S.:  
Stochastic Petri Nets, An Introduction to the Theory;  
Vieweg 1996.

[Heiner 97a]  
Heiner, M.; Popova-Zeugmann, L.:  
Worst-case Analysis of Concurrent Systems with Duration Interval Petri Nets;  

[Heiner 97b]  
Heiner, M.; Popova-Zeugmann, L.:  
On Integration of Qualitative and Quantitative Analysis of Manufacturing Systems Using Petri Nets;  

[Merlin 74]  
Merlin, P.:  
A Study of the Recoverability of Communication Protocols;  

[Popova 91]  
Popova, L.:  
On Time Petri Nets;  
J. Information Processing and Cybernetics EIK 27(91)4, 227-244.

[Popova 94]  
Popova-Zeugmann, L.:  
On Time Invariance in Time Petri Nets;  

[Popova 95]  
Popova, L.:  
On Liveness and Boundedness in Time Petri Nets;  

[Ramchandani 74]  
Ramchandani, C.:  
Analysis of Asynchronous Concurrent Systems Using Petri Nets;  

[Starke 90]  
Starke, P. H.:  
Analysis of Petri Net Models (in German);  
B.G. Teubner 1990.

[Starke 95]  
Starke, P.:  
A Memo On Time Constraints in Petri Nets;  
Humboldt-University zu Berlin, Informatik-Bericht Nr. 46, August 1995.