II.17

MODELLING COMPUTER SYSTEMS EVOLUTIONS: NON-STATIONARY PROCESSES AND STOCHASTIC PETRI NETS - APPLICATION TO DEPENDABILITY GROWTH

J-C Laprie

Rapporteur: C Sala-Oliveras

Modeling Computer Systems Evolutions: Non-Stationary Processes and Stochastic Petri Nets — Application to Dependability Growth —

Jean-Claude Laprie, Mohamed Kaâniche, Karama Kanoun

International Seminar — University of Newcastle upon Tyne
September 3-7, 2001

---

modeling and evaluation of performance, or dependability wrt physical faults:

system structure + { workload; fault occurrence and manifestation, maintenance } → performance, performability, dependability

stochastic Petri nets (SPNs, GSPNs, ESPNs, DSPNs, SANs, etc.)

↑

attention focused on system *logic*

- underlying stochastic processes: homogeneous Markov, semi-Markov, Markov regenerative, device of stages (phase-type expansion)
- software packages: ESPN, GreatSPN, SPNP, SURF-2, TOMSPIN, UltraSAN, etc.

☞ strong implicit assumption: system does not evolve over time

↓

stationary processes

odeling and evaluation of dependability wrt design faults:

↓

non-stationary processes ⟶ software reliability

- large number of reliability growth models, mainly aimed at "black box" systems, thus ignoring system structure
- software packages: SMERFS, SRMP, SoRel, SRE Toolkit, M-élopée, etc.

computer systems *do* evolve with time

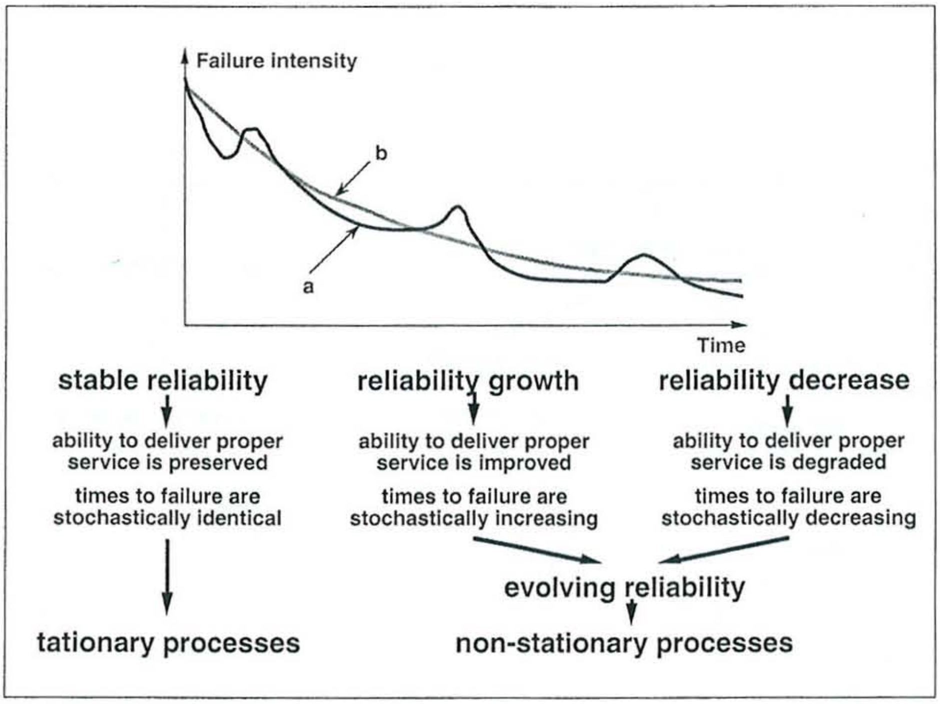
↑

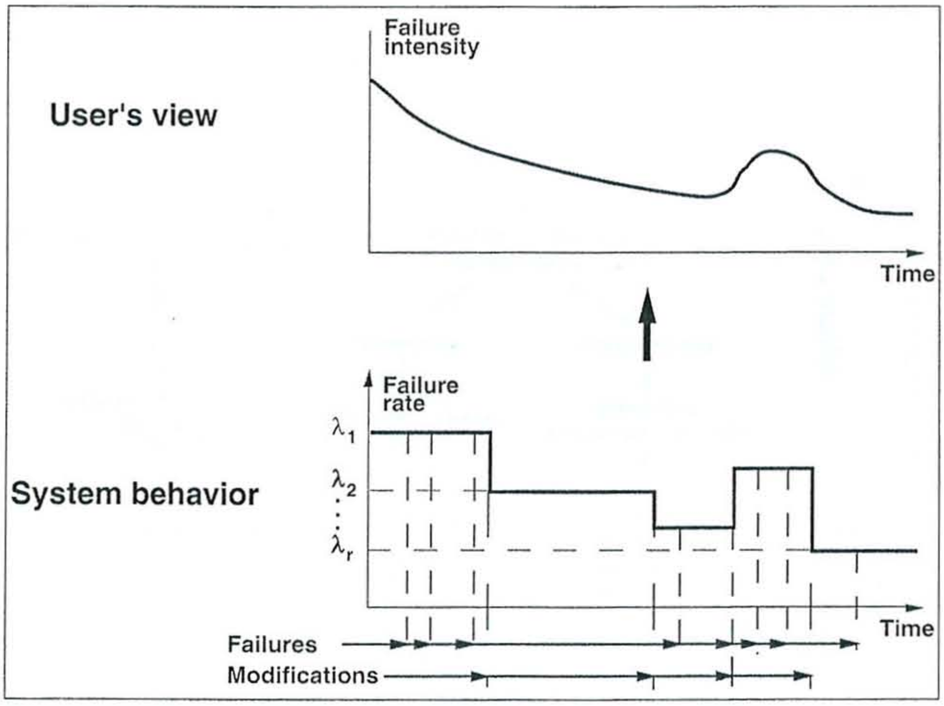
- progressive removal of (unavoidable) residual design faults
- improvement in performance of some functionalities
- addition of new functionalities

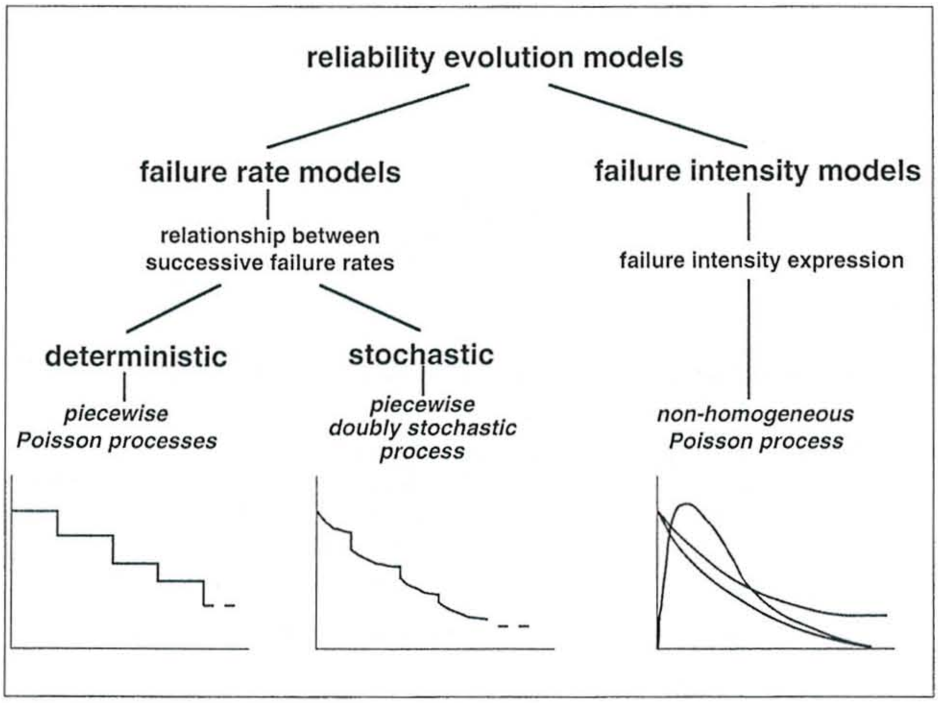
how to take advantage of the existing body of results on stochastic Petri nets to model evolution of computer systems, i.e. to model non-stationary stochastic processes ?

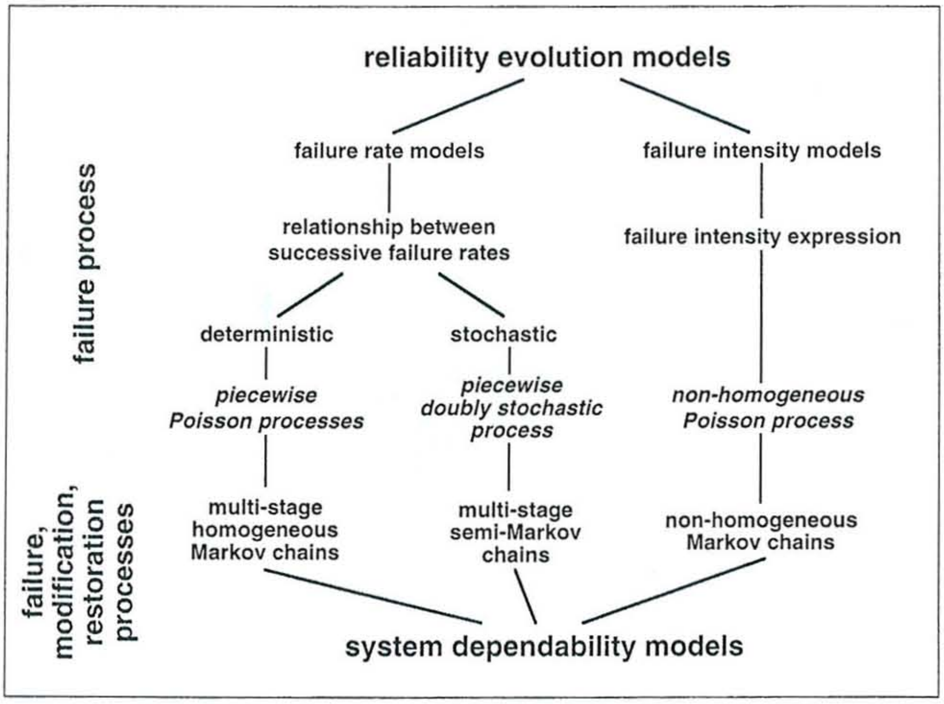
☞ preliminary exploration

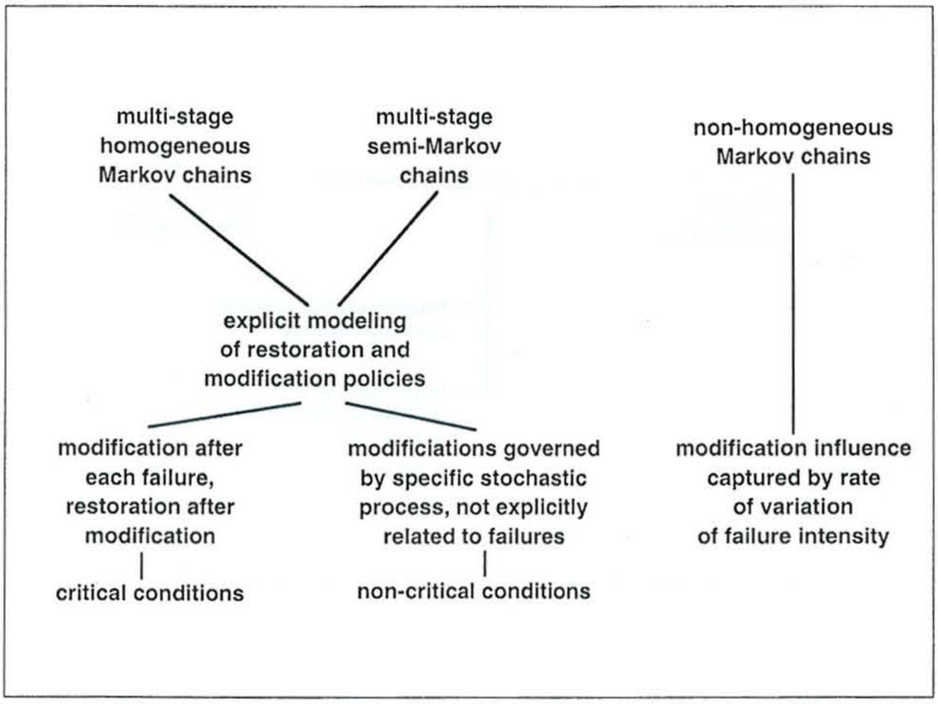
☞ focus on dependability evolutions as resulting from successive modifications

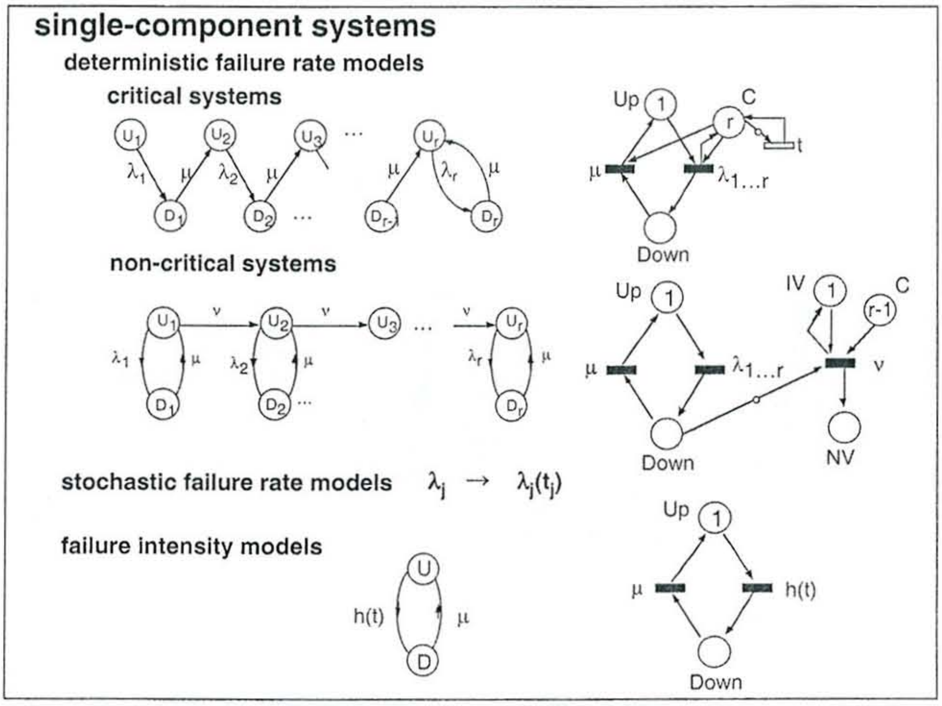

Failure intensity
b
a
Time
stable reliability
ability to deliver proper service is preserved
times to failure are stochastically identical
tationary processes
reliability growth
ability to deliver proper service is improved
times to failure are stochastically increasing
reliability decrease
ability to deliver proper service is degraded
times to failure are stochastically decreasing
evolving reliability
non-stationary processes


User's view
Failure intensity
Time
System behavior
Failure rate
λ1
λ2
λr
Time
Failures
Modifications


reliability evolution models
failure rate models
failure intensity models
relationship between
successive failure rates
failure intensity expression
deterministic
stochastic
piecewise
Poisson processes
piecewise
doubly stochastic
process
non-homogeneous
Poisson process


reliability evolution models
failure process
failure rate models
failure intensity models
relationship between
successive failure rates
failure intensity expression
deterministic
stochastic
piecewise
Poisson processes
piecewise
doubly stochastic
process
non-homogeneous
Poisson process
failure,
modification,
restoration
processes
multi-stage
homogeneous
Markov chains
multi-stage
semi-Markov
chains
non-homogeneous
Markov chains
system dependability models


multi-stage homogeneous Markov chains
multi-stage semi-Markov chains
non-homogeneous Markov chains
explicit modeling of restoration and modification policies
modification after each failure, restoration after modification
critical conditions
modificiations governed by specific stochastic process, not explicitly related to failures
non-critical conditions
modification influence captured by rate of variation of failure intensity

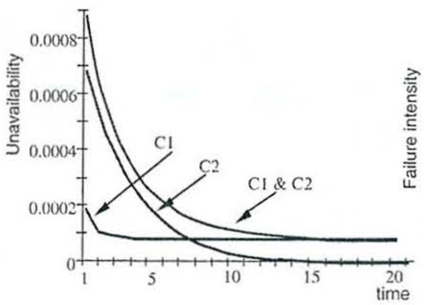

single-component systems
deterministic failure rate models
critical systems
non-critical systems
stochastic failure rate models $\lambda_j \rightarrow \lambda_j(t_j)$
failure intensity models
Up
Down
C
IV
NV
h(t)

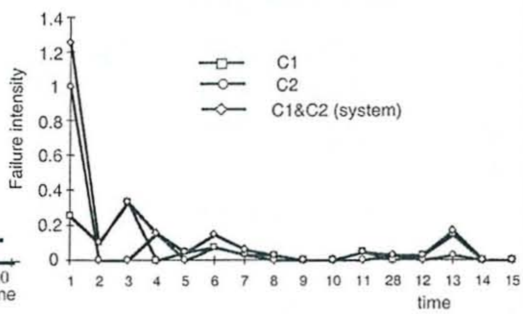
stable reliability stochastic Petri net — **decoration** → **evolving reliability stochastic Petri net**

decoration:
- retains compactness and modularity
- depends upon reliability evolution model

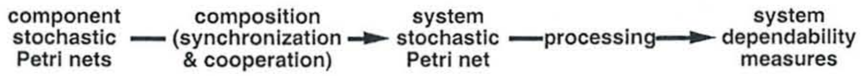
❊ **modification and restoration processes: stationary processes**

multi-component systems: **impact of various component reliability on system dependability**

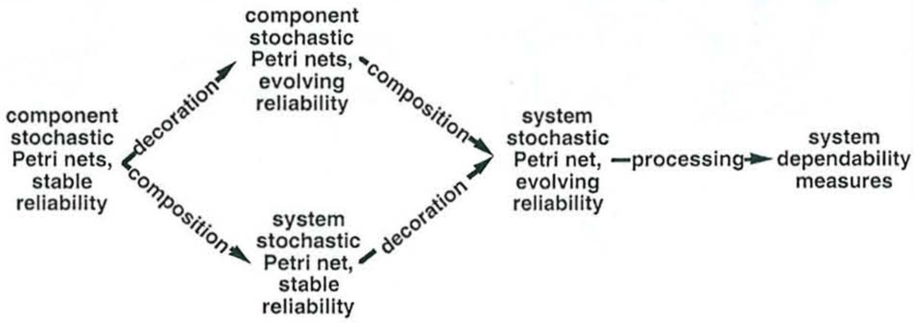




Stable reliability

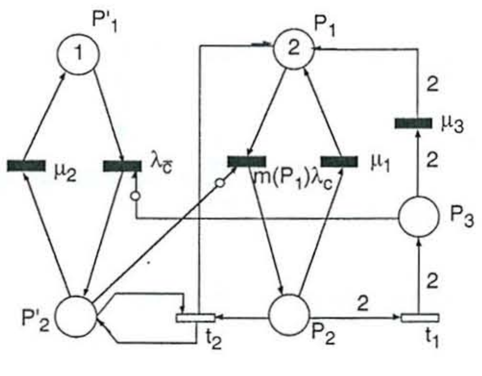


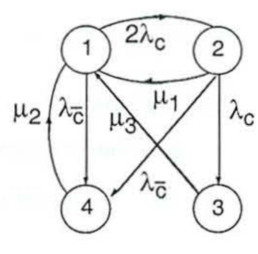
Evolving reliability



stable reliability

2-component fault-tolerant system with imperfect coverage and single repair facility

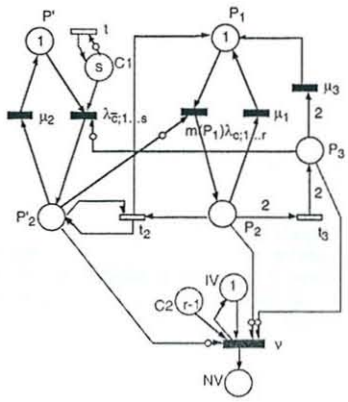


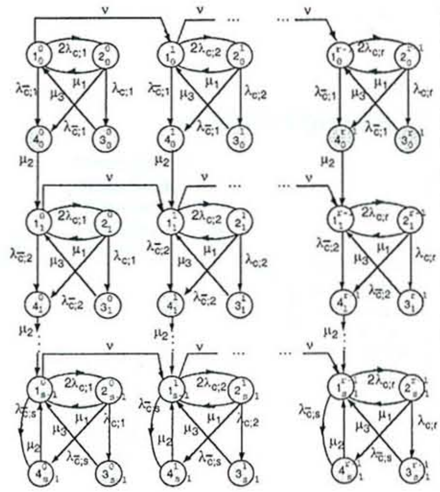


State	Marking
1	{2P1,P'1}
2	{P1,P2,P'1}
3	{2P3,P'1}
4	{2P1,P'2}

eterministic failure rate models

2-component fault-tolerant system





$\lambda_{c.1...s} = \lambda_{c;j}$ with j = f(m(C1)) = m(C1) + 1 - s

$\lambda_{\bar{c}.1...r} = \lambda_{\bar{c};j}$ with j = f(m(NV)) = m(NV) + 1

---

general case

stable reliability ⟶ n-state Markov chain

evolving reliability

p evolutive processes

m_i, i=1,...,p evolution stages for process i ⟶ $n \prod_{i=1}^{p} m_i$ state Markov chain

❊ significant increase in state space cardinality

❊ randomization techniques, possibly exploiting sparse transition matrices

stochastic failure rate models —| practical severe limitations of semi-Markov chains

2-component fault-tolerant system

not a semi-Markov chain

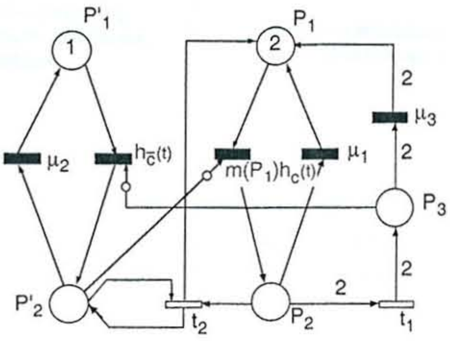
some specific cases

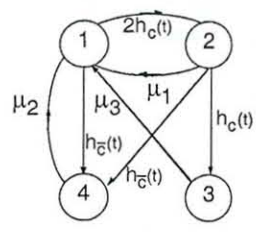
Markov regenerative chains?

device of stages?

ailure intensity models

2-component fault-tolerant system





general case

- ❋ state space cardinality unchanged
- ❋ processing non-homogeneous Markov chains

↓

solution of Kolmogorov forward differential equations

- explicit solutions cumbersome, if at all possible
- alternatives:
  - numerical integration
  - approximation of continuous-time processes with discrete-time equivalents

	multi-stage homogeneous Markov chains	non-homogeneous Markov chains
	processing can be performed via classical techniques	compact model
	dramatic increase in state space cardinality	processing necessitates complex, sometimes adhoc, techniques

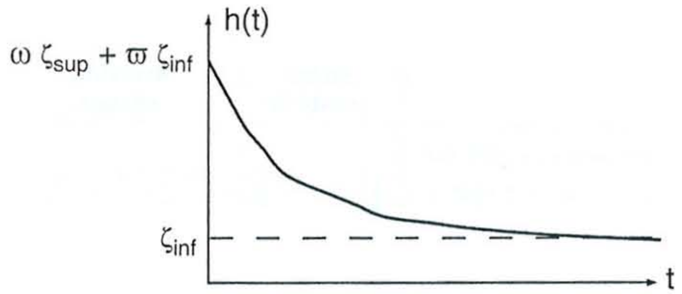
proposed compromise to overcome
the above difficulties:

transformation approach,

based on the Markov interpretation
of the hyperexponential model for evolving reliability

hyperexponential model

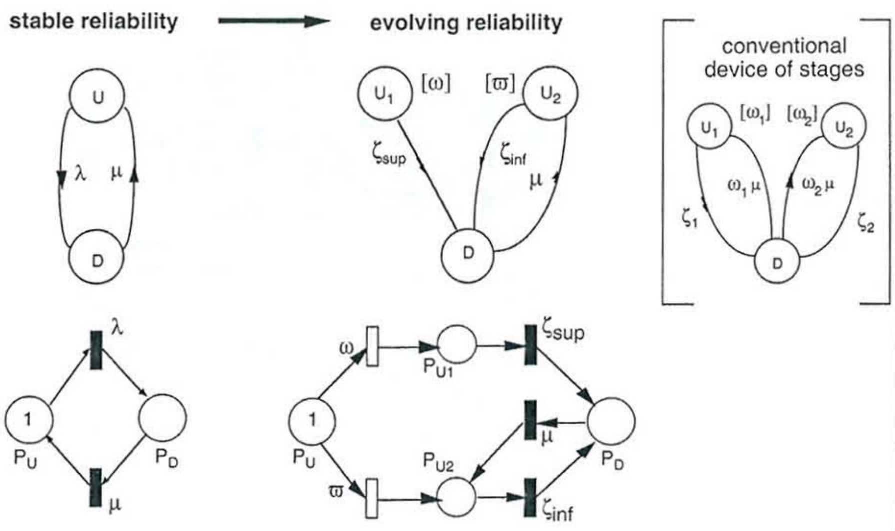
$$h(t) = \frac{\omega\ \zeta_{sup}\, e^{-\zeta_{sup} t} + \varpi\ \zeta_{inf}\, e^{-\zeta_{inf} t}}{\omega\, e^{-\zeta_{sup} t} + \varpi\, e^{-\zeta_{inf} t}} \qquad 0 \le \omega \le 1 \qquad \omega + \varpi = 1$$



arkov interpretation of hyperexponential model

↓

transformation approach



multi-component systems

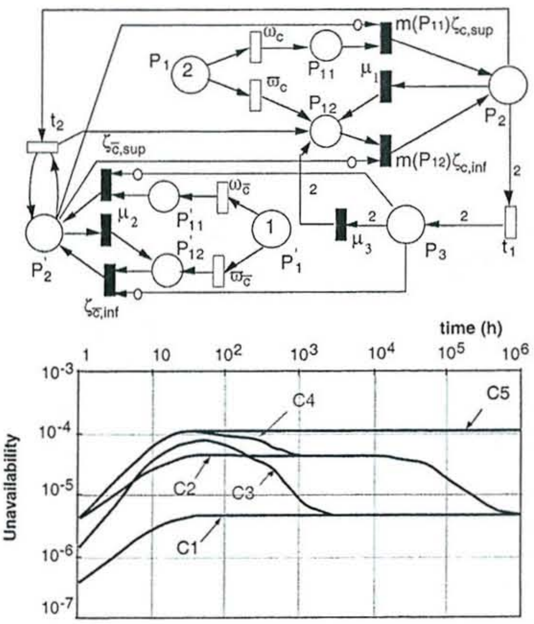
1) construction of the stochastic Petri net assuming stable reliability

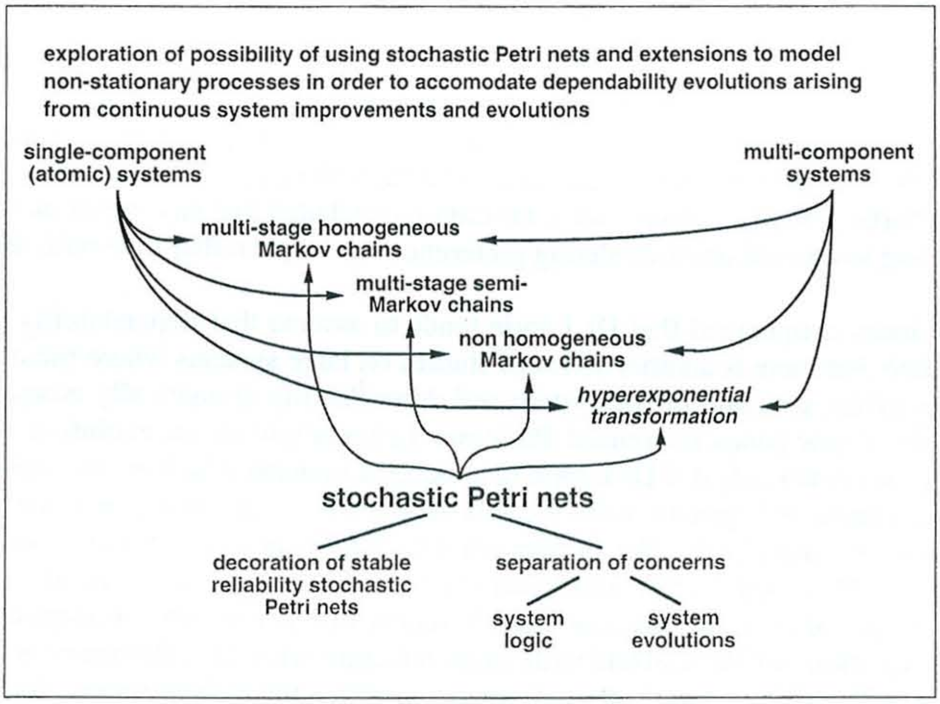
2) transformation of the stochastic Petri net

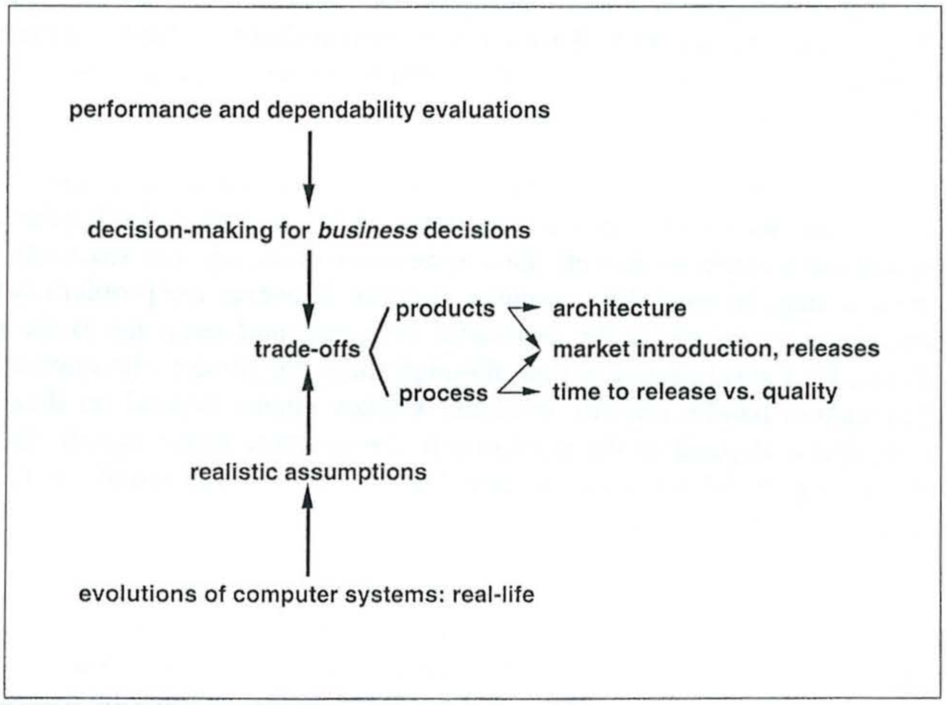
3) processing of the transformed stochastic Petri net

	stable reliability	evolving reliability
number of states for n component system	$\left[n+1\,,\,2^{n}\right]$	$\left[\frac{(n+1)(n+2)}{2}\,,\,3^{n}\right]$

2-component fault-tolerant system




exploration of possibility of using stochastic Petri nets and extensions to model non-stationary processes in order to accomodate dependability evolutions arising from continuous system improvements and evolutions
single-component (atomic) systems
multi-component systems
multi-stage homogeneous Markov chains
multi-stage semi-Markov chains
non homogeneous Markov chains
hyperexponential transformation
stochastic Petri nets
decoration of stable reliability stochastic Petri nets
separation of concerns
system logic
system evolutions


performance and dependability evaluations
decision-making for business decisions
trade-offs
products
process
architecture
market introduction, releases
time to release vs. quality
realistic assumptions
evolutions of computer systems: real-life

DISCUSSION

Rapporteur: C Sala-Oliveras

Lecture Two

Dr Rushby thought that stochastic Petri nets can be fed into a tool like Ultrasound and Markov chains can be fed into a Markov chain tool, but he wondered what would cause someone to choose between one or another tool. Dr Laprie replied that one could use tools such as SPN or GSPN, which one can feed with Petri Nets and in intermediate stages one can have Markov chain representations. Dr Laprie concluded that any use of one or another tool has more to do with one's modeling preferences and style rather than with the tool.

Professor Jones commented that Dr Laprie tends to assume that dependability gets better over the time, but there is another series of studies on large systems where there appears to be a decay of the structure of the system and dependability dramatically worsens as time goes by, (Professor Jones referenced Professor Lehman's work on evolution of OS360). Then Professor Jones asked if Dr Laprie is looking at systems which are in narrow enough domains in which Dr Laprie may not see this decrease of dependability in his evolutionary model. Dr Laprie asked for a clarification on what Professor Jones meant by deterioration of structure. Professor Jones remarked that even though there is not a precise characterization of structure deterioration, it seems that it could be conceptualized by a systems' evolution, which start-off with an architecture which fits the original purpose of the system, followed by change of requirements at which the architecture no longer fits the revised purpose. One observation from that is, for a simple change, one needs to touch an increasing percentage of the modules in the system and then one could have the feeling that the structure of the system is no longer right. Dr Laprie replied that he assumes global increase of dependability over time. Dr Laprie went on saying that there is a particular case where some corrections may worsen the system; however, these cases can be also accommodated into the mathematical formula.

Professor Malek commented on this known difficulty in modelling complexity, and he asked where we are now and what size of systems can be modelled. Dr Laprie said that in simple cases, these models work well. Dr Laprie went on to say that stochastic Petri nets have made great steps in modelling complex systems. However the problem is the size of the Markov chains produced by the stochastic Petri nets and referring to the size of the Markov chains, Dr Laprie observed that although there are some techniques available, in general, they cannot handle realistic systems. Markov chains depend on their stochastic Petri nets which also depend on the regularity in the structure of the system and the states and transitions one could see in a system. Dr Laprie concluded that, at the moment, realistic systems cannot be modelled.

Professor Strigini pointed out that when you get a more complete model, an important problem to solve would be to estimate the model's parameters or validate the structure, because one cannot assume independence between behaviours of 2 components without the obligation to estimate some behavioural common failure. Professor Strigini went on saying that this parameter estimation requires a modelling effort and he asked for comments on levels of decomposition (of systems) to which Dr Laprie's models are helpful. Dr Laprie replied that Professor Stringini summarized the problem well, that there is needed a modelling effort. Dr Laprie added that there are some processes and techniques to simplify this modelling effort however Dr Laprie agreed that it is still a difficult problem.

Professor Strigini asked for an existing example with the level of granularity so that makes sense to decompose it. Dr Laprie responded that, for instance, telecommunication software is an example. Dr Laprie also pointed out that decomposition has to be consistent with the specifications (and functions of the system) regarding not to decompose the system into too many components as the model would get too complicated.