Dependability Analysis Using SAM

Tianjun Shi, Xudong He
School of Computer Science
Florida International University
{tshi01, hex}@cs.fiu.edu

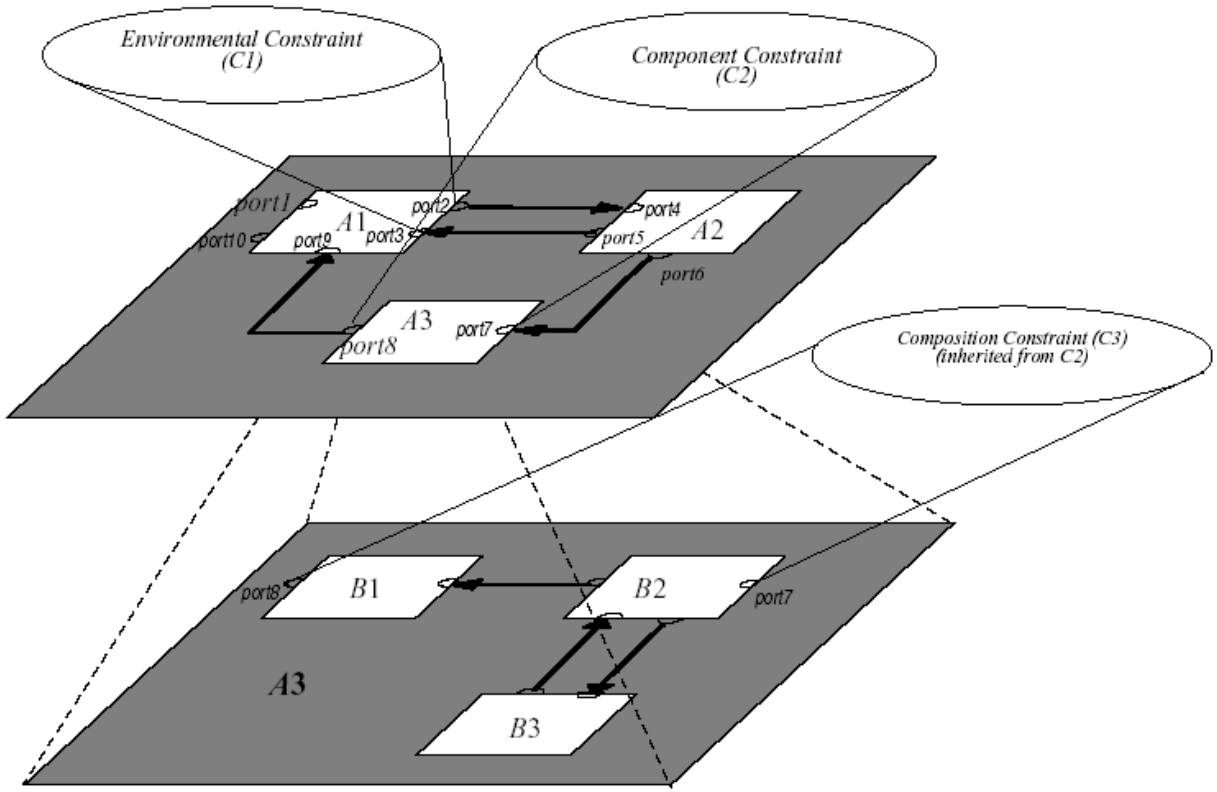
Goal & Method

- Goal
  - Enable the Software Architecture Model (SAM) to model and analyze both functional properties and common non-functional properties
- Method
  - Extend SAM with stochastic constructs
  - Transform SAM model to SRN model

The SAM Model

- A SAM model: $\{C, h\}$
  - A set of compositions $C = \{C_1, C_2, \ldots, C_k\}$
  - A hierarchical mapping h relating compositions.
- A composition: $C_i = \{Cm_i, Cn_i, Cs_i\}$
  - Cm_i: a set of components
  - Cn_i : a set of connectors
  - Cs_i : a set of composition constraints
- Components / Connectors: $C_{i_j} = \{B_{i_j}, S_{i_j}\}$
  - Bi_j : behavior model (a Petri net)
  - Si_j : property specification (a temporal logic formula)

The SAM model (Cont'd)



A graphic view of a SAM architecture model

Predicate Transition Net (PrT net)

- A PrT net is a class of high level Petri net, and is defined as a tuple (N, $Spec$, ins), where
  - $N = (P, T, F)$, the net structure
  - $Spec = (S, OP, Eq)$, the underlying specification
  - ins = (φ, L, R, M_0), the net inscription associating a net element in N with its denotation in $Spec$.

Stochastic Reward Net (SRN)

- SRN is an extension to Stochastic Petri Net
  - A firing rate for each transition, which could be marking dependent
  - Enabling Function for each transition
  - Priority for each transition
- Tools for SRNs
  - SPNP
  - SMART

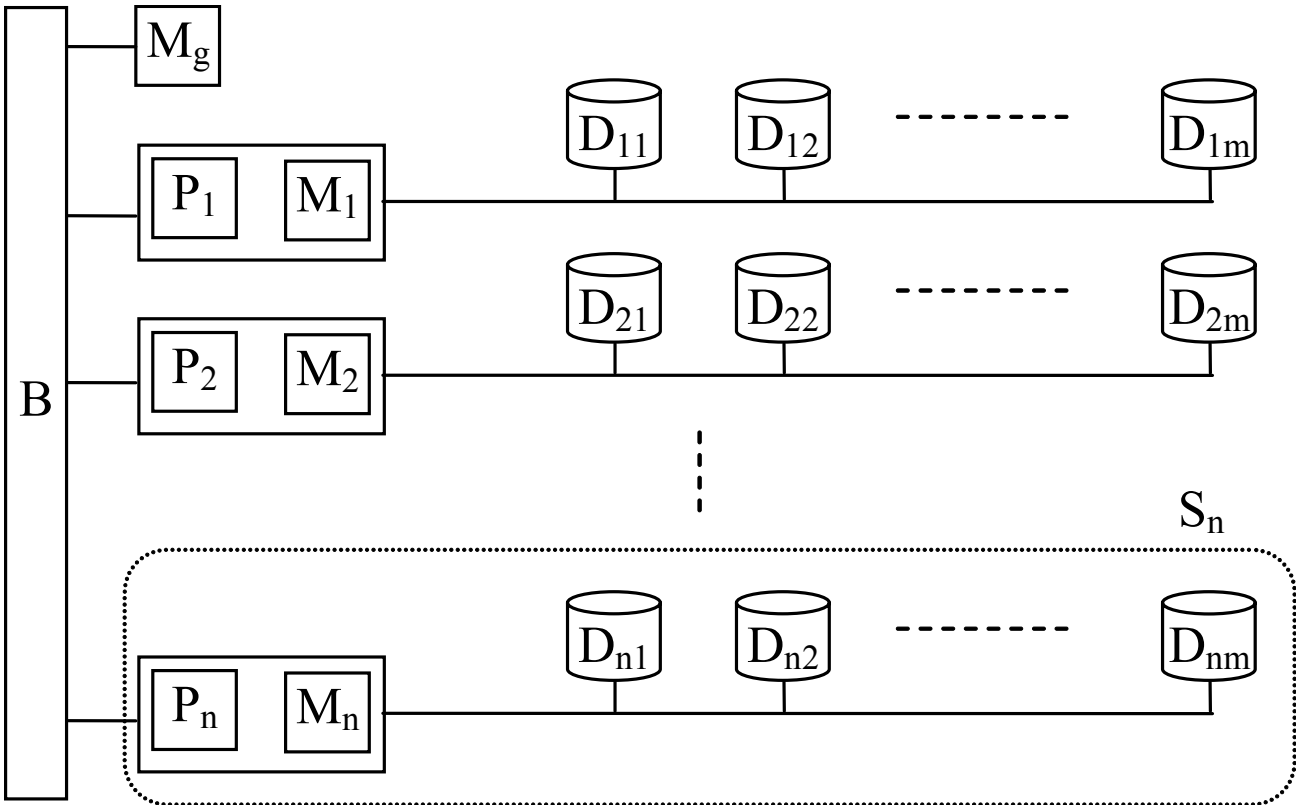
Extension on SAM

- Add a stochastic construct into the behavior model expressed in a PrT net
  - A special variable *RATE* is used in the constraint of a transition to specify the firing rate.
  - Firing rate is not necessarily constant
- Formally specify non-functional property requirements using *Probabilistic real time Computation Tree Logic* (*PCTL*)

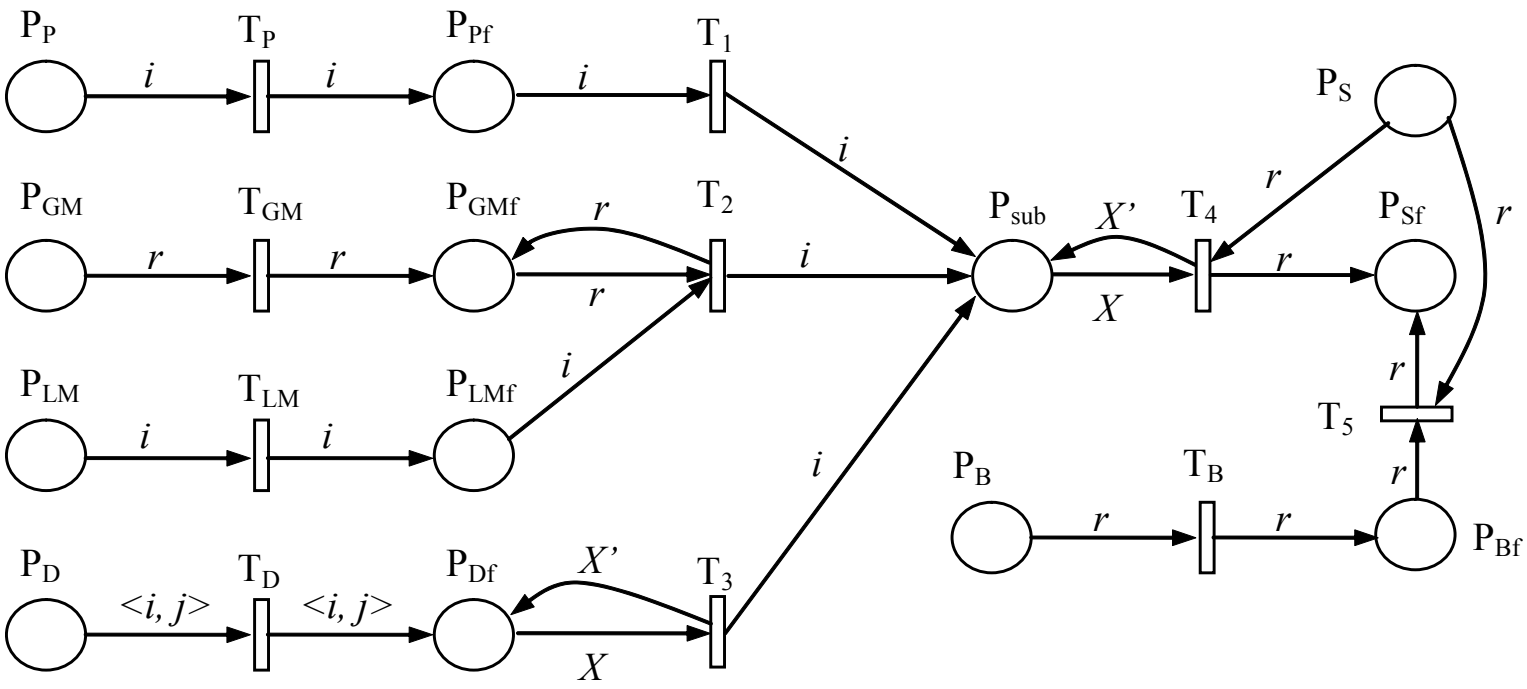
Transformation from SAM to SRN

- Unfold the behavior model to a low level Petri net.
  - Unfold each transition *T* into a set of transitions based on the set of constant substitution that satisfy the constraint of *T*.
  - Places are connected to the unfolded transitions according to the substitution.
  - Remove the dead transitions and combine equivalent elements if any.
- Assign the firing rate to each transition based on the stochastic construct.
- Solve the transformed SRN to evaluate dependability.

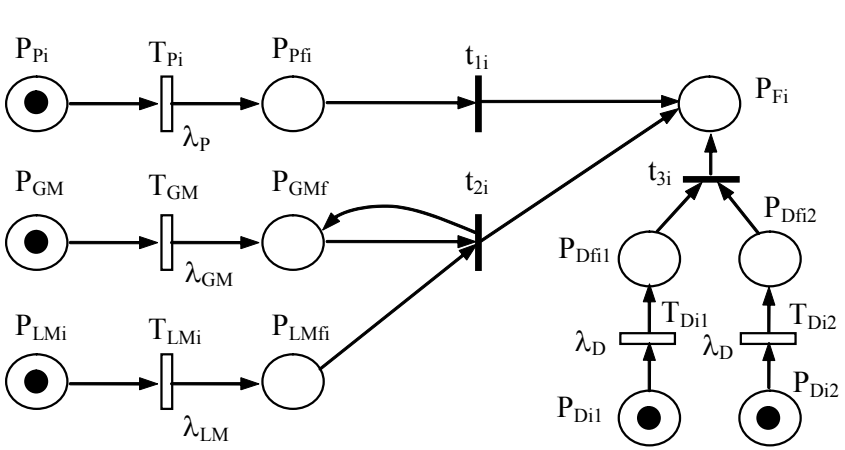
An example: the multiprocessor system



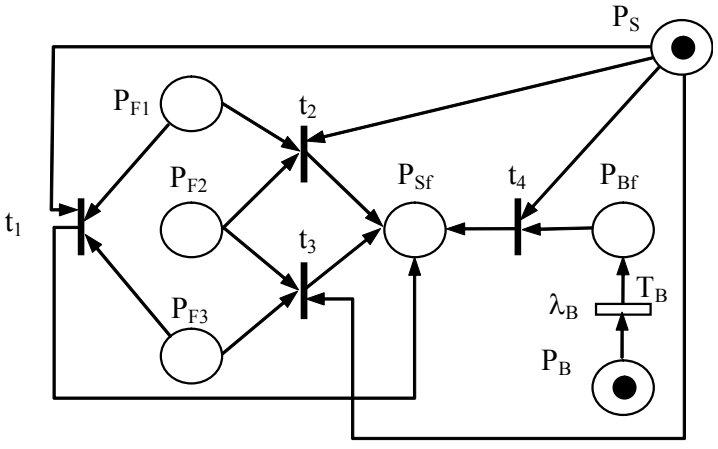
The behavior model of the example system



The SRN model of the example system

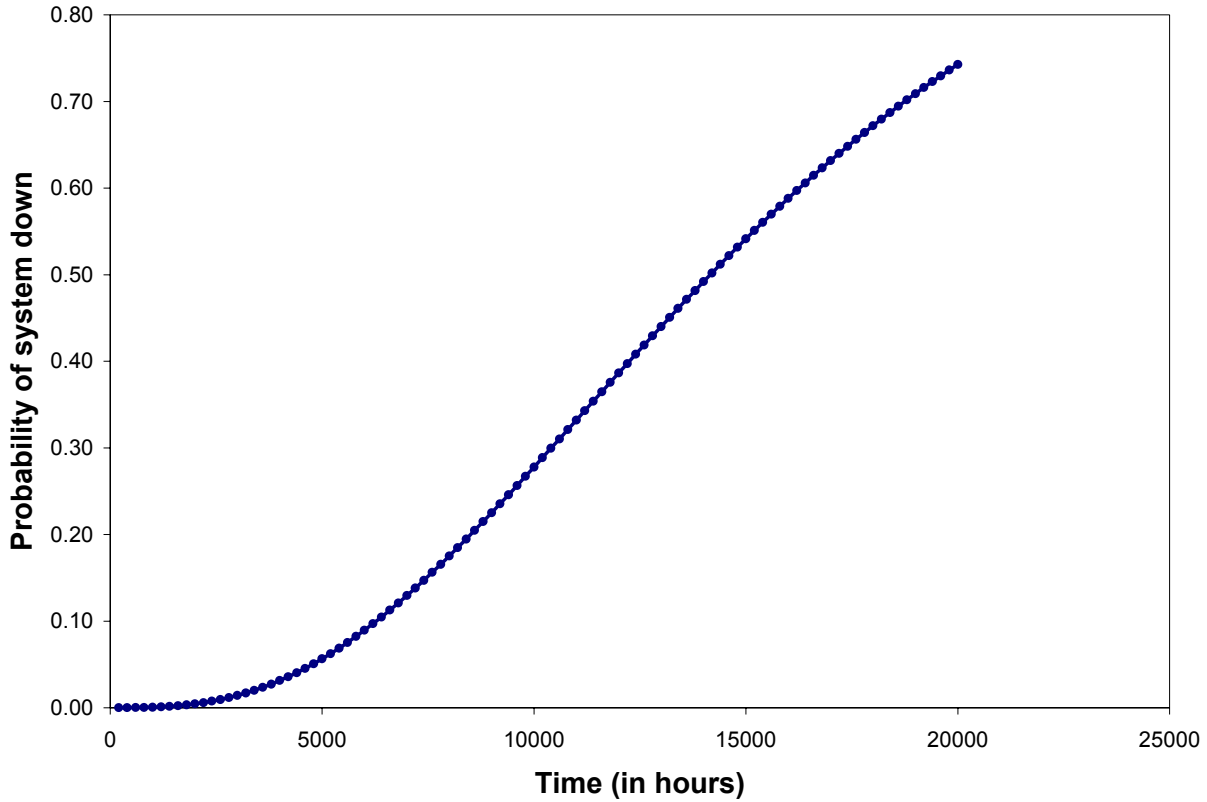


(a)



(b)

Analysis Results



Conclusion

- The analysis of dependability using SRNs is not new.
- Our contribution lies in incorporating stochastic information into the SAM so that both functional properties and common non-functional properties like dependability can be analyzed under a unified framework.
- Developing tools to automate the transformation is being considered in our future work.