# The RADYBAN tool: Reliability Analysis with DYnamic BAyesian Networks

Faculty: *Andrea Bobbio, Stefania Montani, Luigi Portinale*

Post-doc: *Daniele Codetta-Raiteri*

Students: *Stefano Di Nolfo, Marco Varesio*

Dipartimento di Informatica - www.di.unipmn.it

Università del Piemonte Orientale "A. Avogadro"

# Outline

- Classes of models in the Dependability/Reliability field
- Dynamic Bayesian Networks (DBN)
  - Definition
  - Case studies
- RADYBAN tool
  - Architecture and use
  - Conversion rules from Dynamic Fault Tree (DFT) to DBN
    - Case study

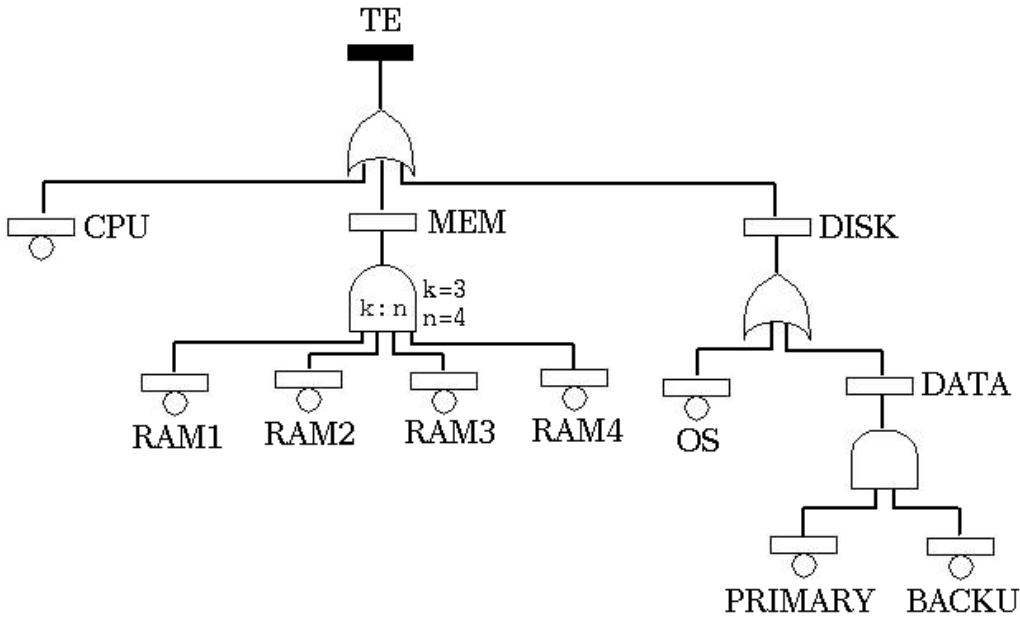
# Reliability evaluation

- **Measurement-based** evaluation
  - It requires the observation of the behaviour of the system physical components.
  - It may be expensive or unpractical.

- **Model-based** evaluation
  - A model is a convenient abstraction of the system.
  - A model has a certain degree of accuracy.
  - A model can be the object of analysis or simulation.
  - **Models classification:**
    - Combinatorial models
    - State space based models
    - Models with conditional local dependencies

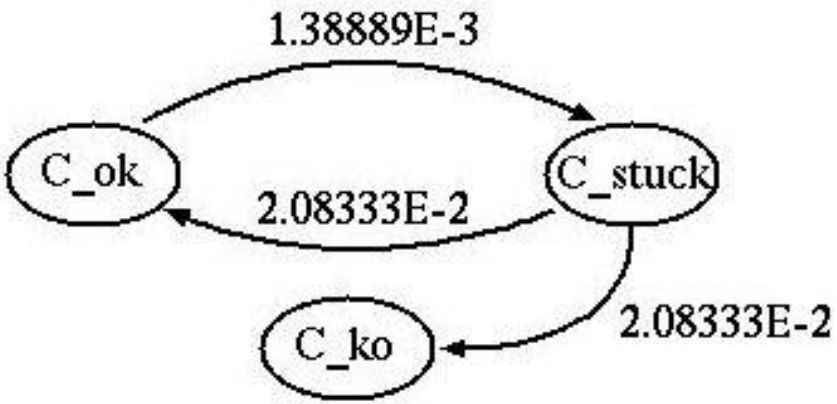
# Combinatioral models

- They represent the structure of the system in terms of logical connection of working (failed) components in order to obtain the system success (failure).
  - **Fault Trees**, Reliability Block Diagrams, Reliability Graphs
  - Easy to use, concise, analytically tractable
  - Limited modeling power



# State space based models

- They enumerate the set of meaningful states and state transitions of the system
  - **Markov Chains**, Markov Decision Processes, Petri Nets
  - State space may be over-specified with respect to the modeling needs
  - Dynamic behavior of the system may lead to the explosion of the state space size



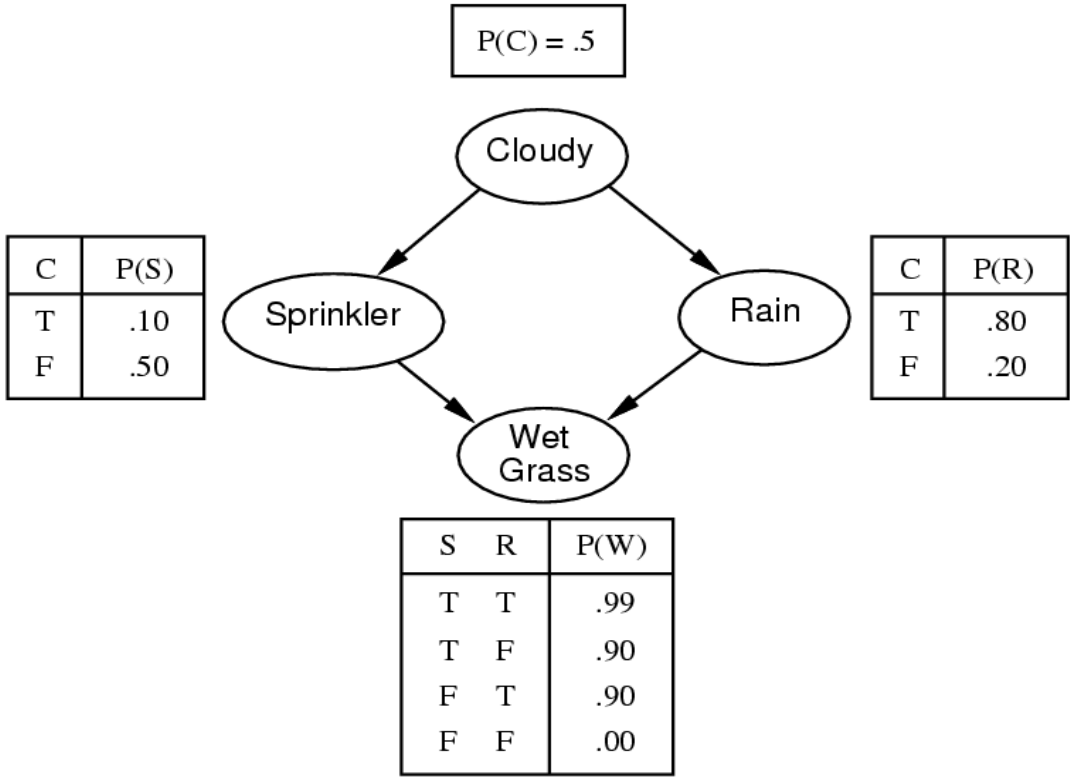
# Models with conditional local dependencies

- Bayesian (or Belief) Networks (BN) are a widely used formalism for representing uncertain knowledge in probabilistic systems, applied to a variety of real-world problems

- BN are defined by a directed acyclic graph in which discrete random variables are assigned to each node, together with the conditional dependence on the parent nodes (Conditional Probability Table (CPT))
  - Root nodes are nodes with no parents, and marginal prior probabilities are assigned to them

# Bayesian Networks

- Diagnostic inference
  - Pr(cause | effect)
    - Pr(Sprinkler | Wet grass)
    - Pr(Cloudy | Wet grass)
- Causal inference
  - Pr(effect | cause)
    - Pr(Wet grass | Cloudy)
    - Pr(Wet grass | Sprinkler)
- …




- Exact algorithms (*Clustering*, *Conditioning, Variable Elimination* (Factoring), …) or approximated algorithms (*Stochastic Simulation*) for BN inference

# Inference in BN

- Find Pr(Q=q | E=e)
  - Q is the query variable
  - E is the set of evidence variables

$$\Pr(q \mid e) = \frac{\Pr(q, e)}{P(e)}$$

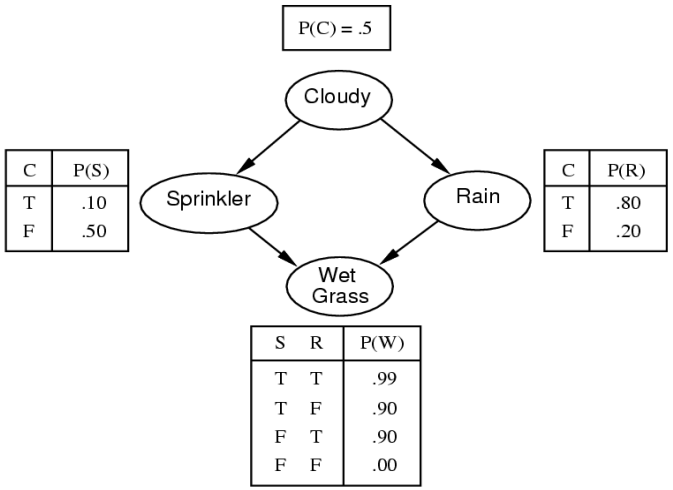
- $X_1, \ldots, X_n$ are network variables except Q, E.

$$\Pr(q, e) = \sum_{x_1, \ldots x_n} \Pr(q, e, x_1, \ldots, x_n)$$

- $Y_1, \ldots, Y_n$ are network variables except E.

$$\Pr(e) = \sum_{y_1, \ldots y_n} \Pr(e, x_1, \ldots, x_n)$$

# Inference example




## Joint probability distribution

$$P(\bar{c},\bar{s},\bar{r},\bar{w}) = P(\bar{c})P(\bar{s}\,|\,\bar{c})P(\bar{r}\,|\,\bar{c})P(\bar{w}\,|\,\bar{s},\bar{c}) = p1 \quad (0.5\cdot 0.5\cdot 0.8\cdot 1 = 0.2)$$

$$P(\bar{c},\bar{s},\bar{r},w) = P(\bar{c})P(\bar{s}\,|\,\bar{c})P(\bar{r}\,|\,\bar{c})P(w\,|\,\bar{s},\bar{c}) = p2$$

$$P(\bar{c},\bar{s},r,\bar{w}) = P(\bar{c})P(\bar{s}\,|\,\bar{c})P(r\,|\,\bar{c})P(\bar{w}\,|\,\bar{s},\bar{c}) = p3$$

$$\vdots$$

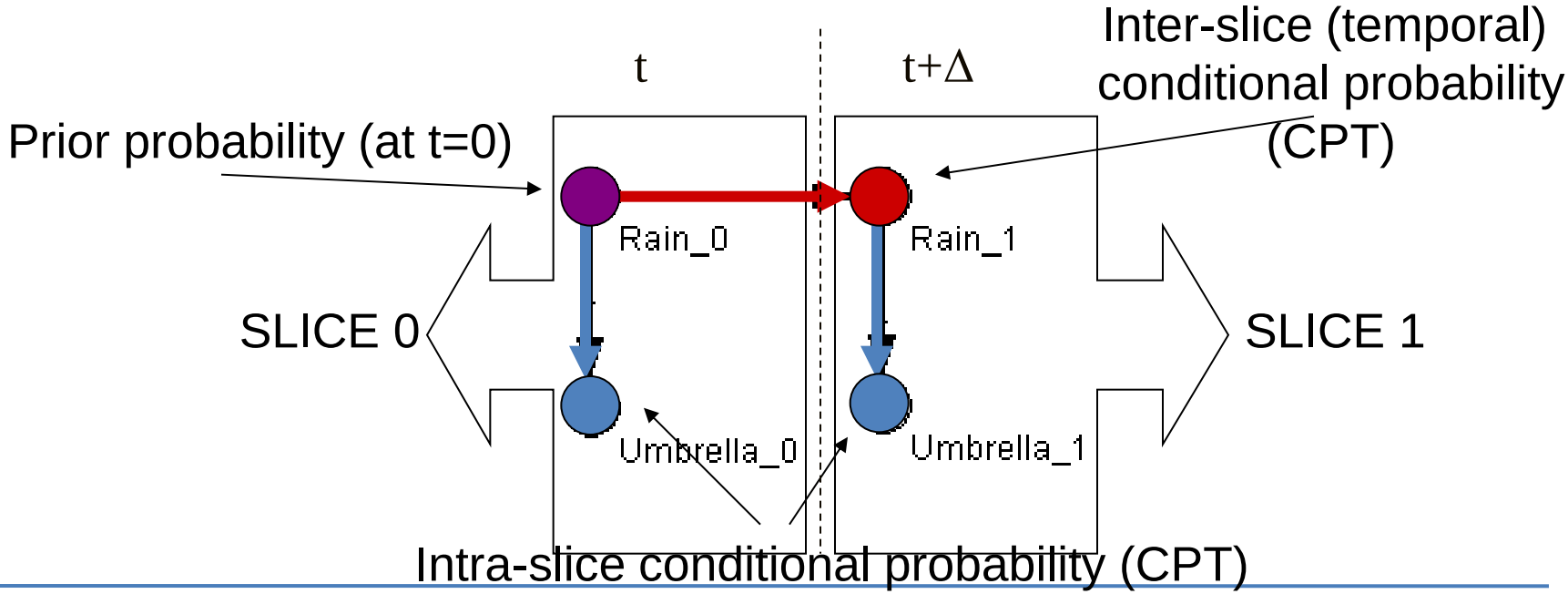
$$P(c,s,r,w) = P(c)P(s\,|\,c)P(r\,|\,c)P(w\,|\,s,c) = p16$$

**Query**:

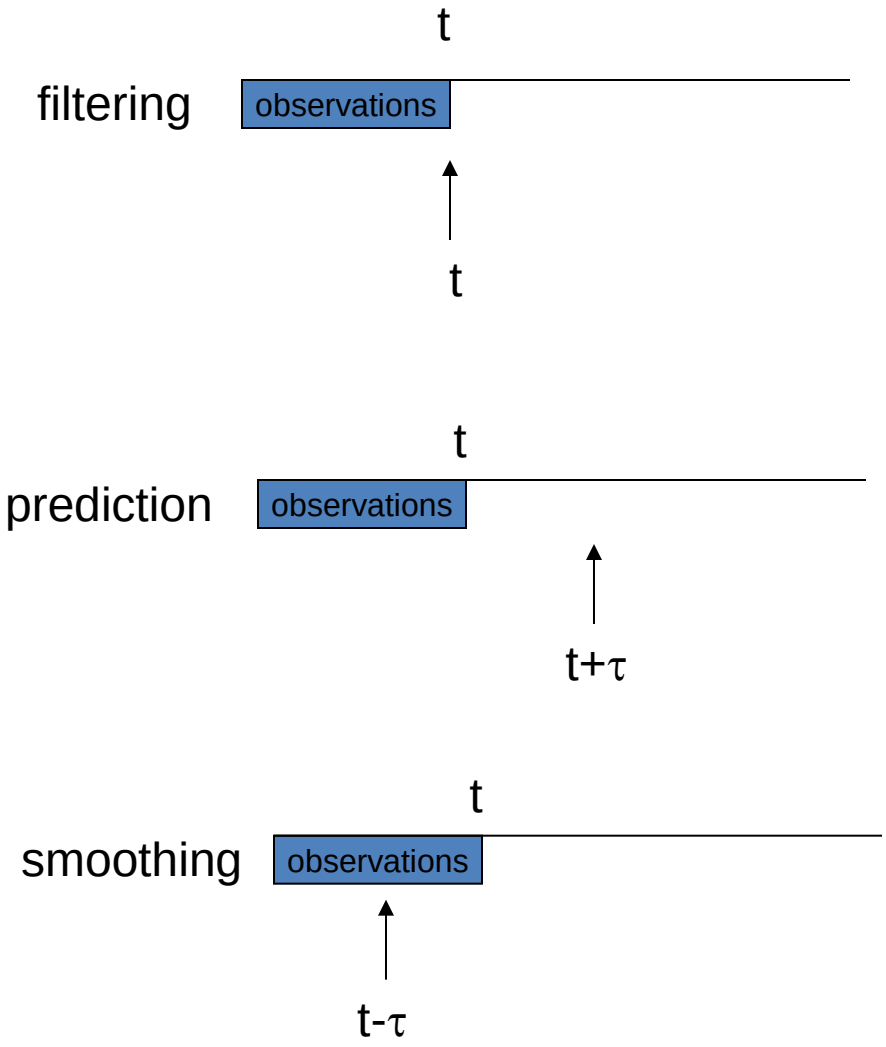
$$P(\bar{c}\,|\,\bar{r},w) = \frac{P(\bar{c},\bar{r},w)}{P(\bar{r},w)} = \frac{P(\bar{c},\bar{s},\bar{r},w)+P(\bar{c},s,\bar{r},w)}{P(\bar{c},\bar{s},\bar{r},w)+P(\bar{c},s,\bar{r},w)+P(c,\bar{s},\bar{r},w)+P(c,s,\bar{r},w)}$$

# Dynamic Bayesian Network

- DBN introduce a **discrete** temporal dimension:
  - The system is represented at several time slices
  - Conditional dependencies among variables at different slices, are introduced to capture the temporal evolution.
  - Time invariance is assumed: typically 2 time slices (t, t+Δ) are assumed in DBN: Markovian assumption



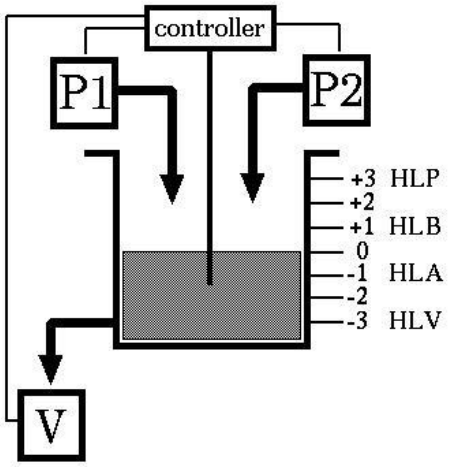
# Inference on DBN

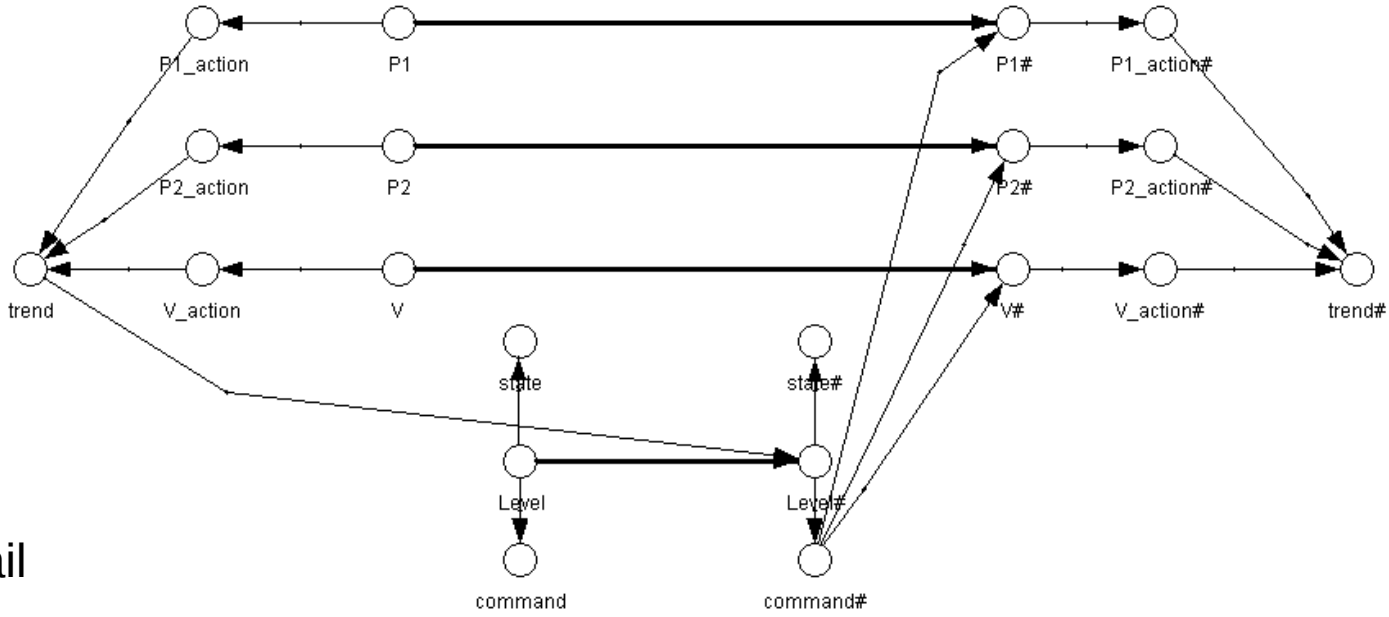



Inference algorithms:

- Slice Junction Tree (Exact)
- Boyen-Koller (Approximated)
- …

# DBN model and analysis of a benchmark



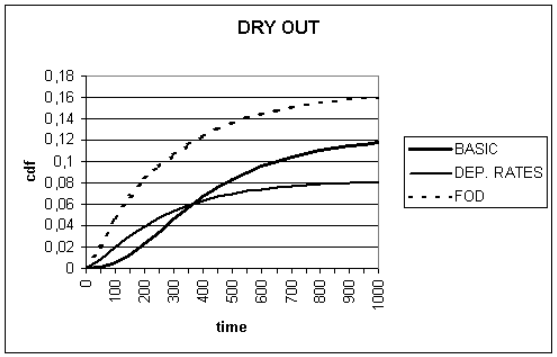

Each component may fail

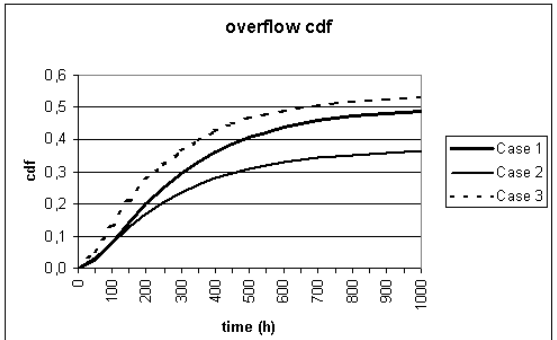
**Control laws:**

- $H \leq HLA \Rightarrow$ P1:ON, P2:ON, V:OFF.
- $H \geq HLB \Rightarrow$ P1:OFF, P2:OFF, V:ON.

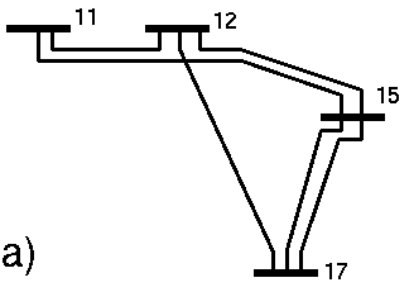
**Failure conditions:**

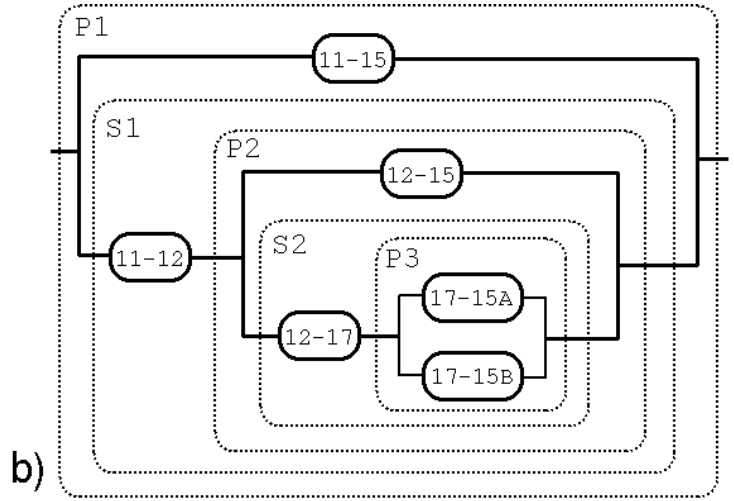
- Dry out (H<HLV)
- Overflow (H>HLP)

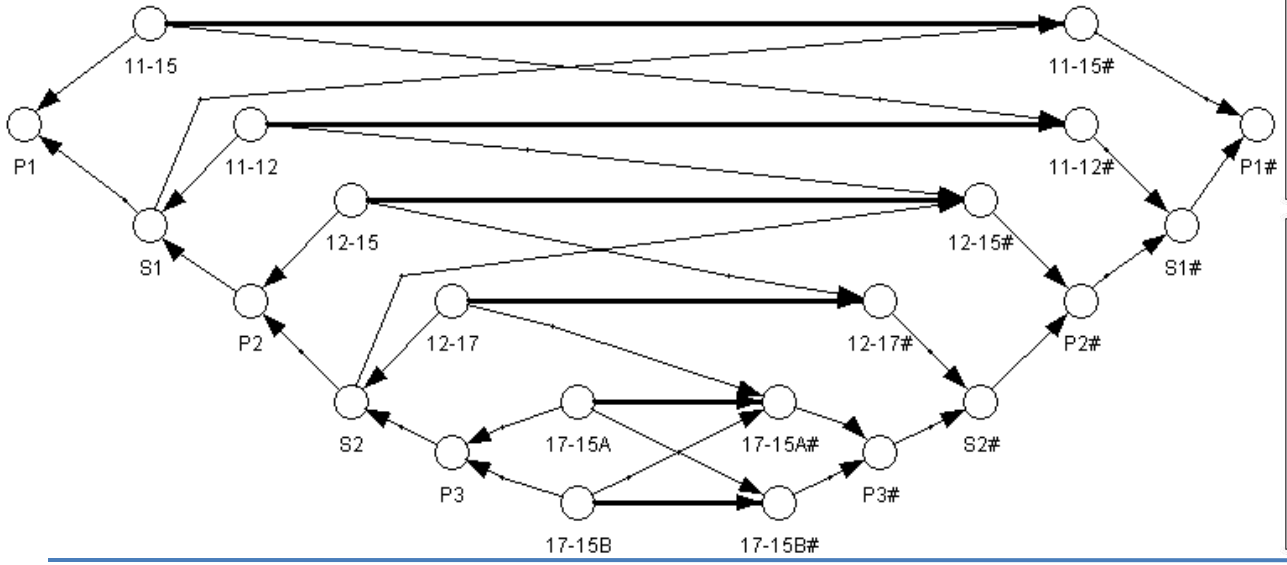


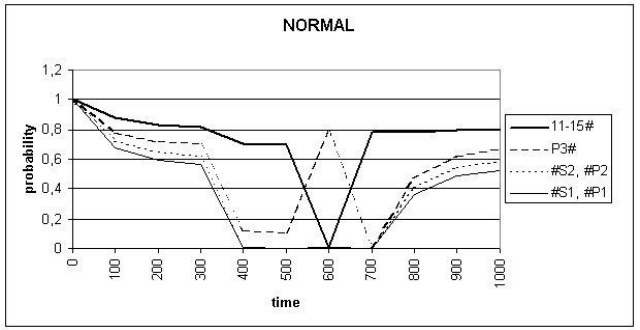



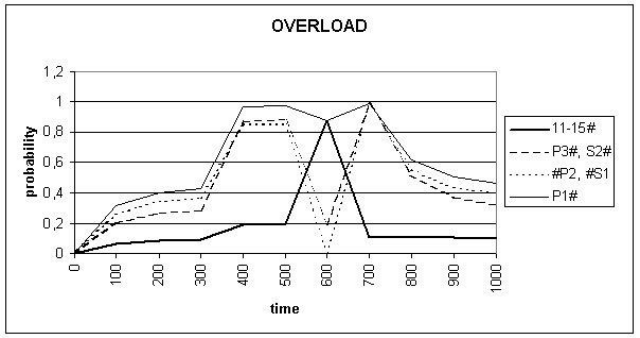
# DBN model of cascading effects in a power grid

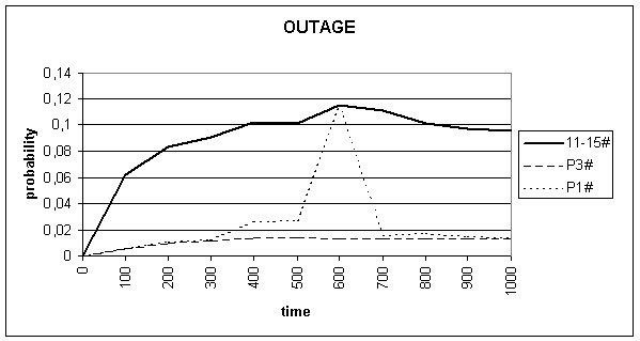


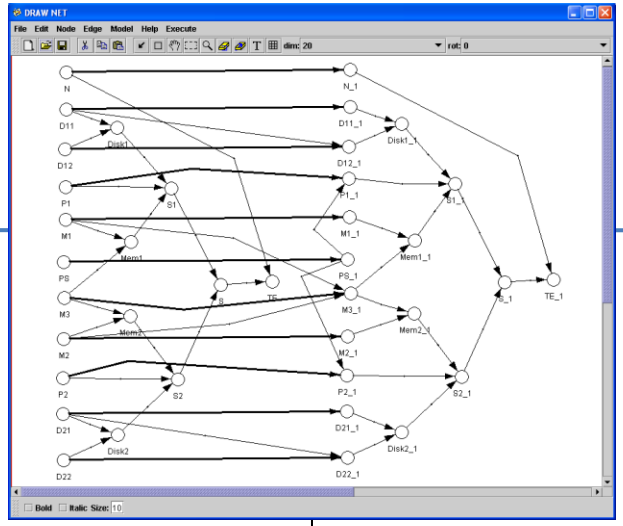





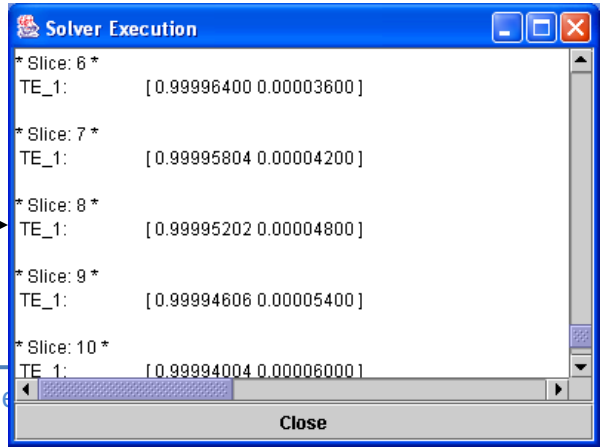

# RADYBAN architecture and use (1)

**DFT2DBN**

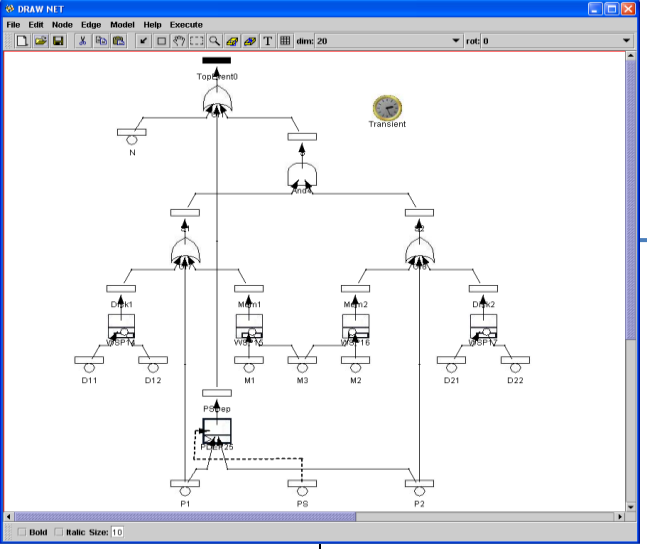
**DBN analyzer**

DBN.xml

Results



# RADYBAN architecture and use (2)




DFT.xml

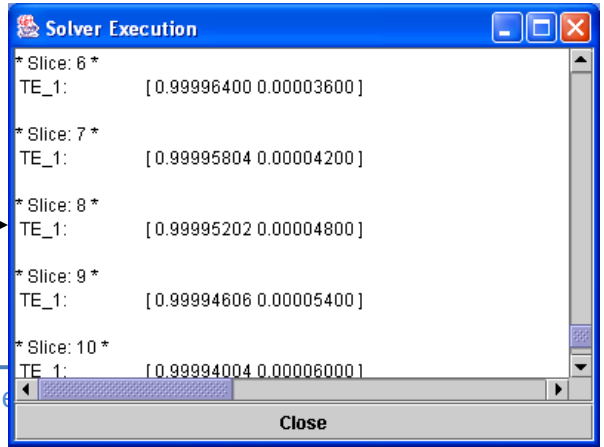
**DFT2DBN**

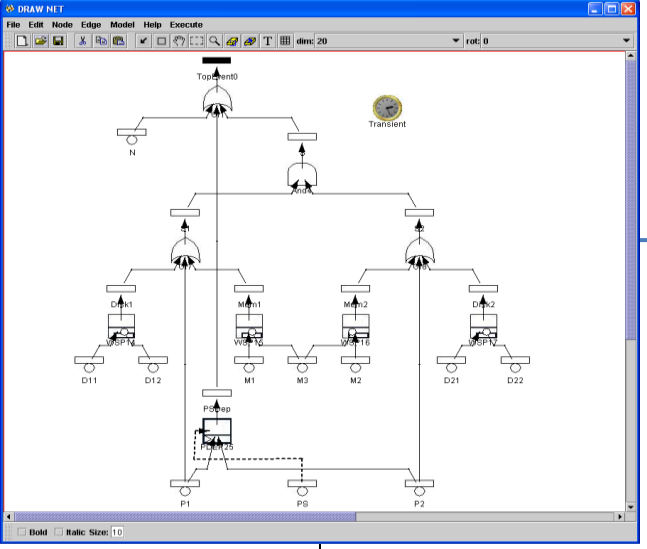
DBN.xml

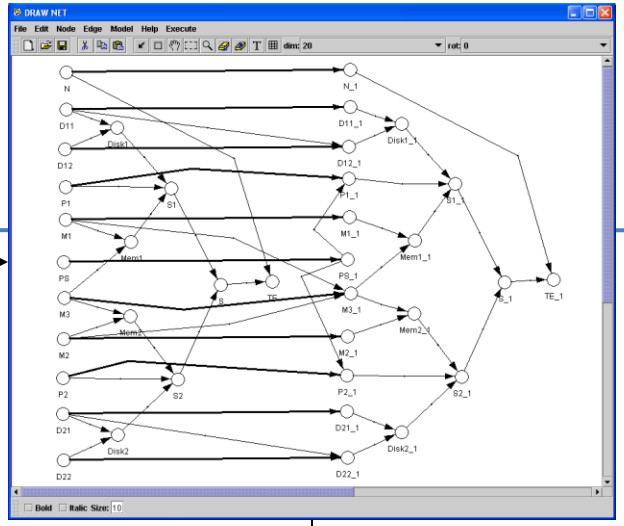
DBN.xml

**DBN analyzer**

Results






DFT.xml

**DFT2DBN**

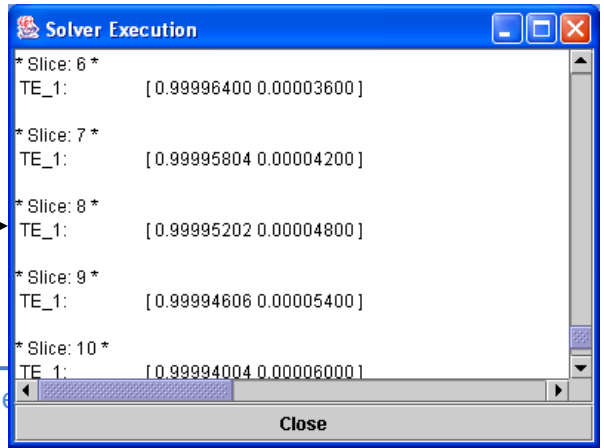
DBN.xml

# RADYBAN architecture and use (3)

**DBN analyzer**

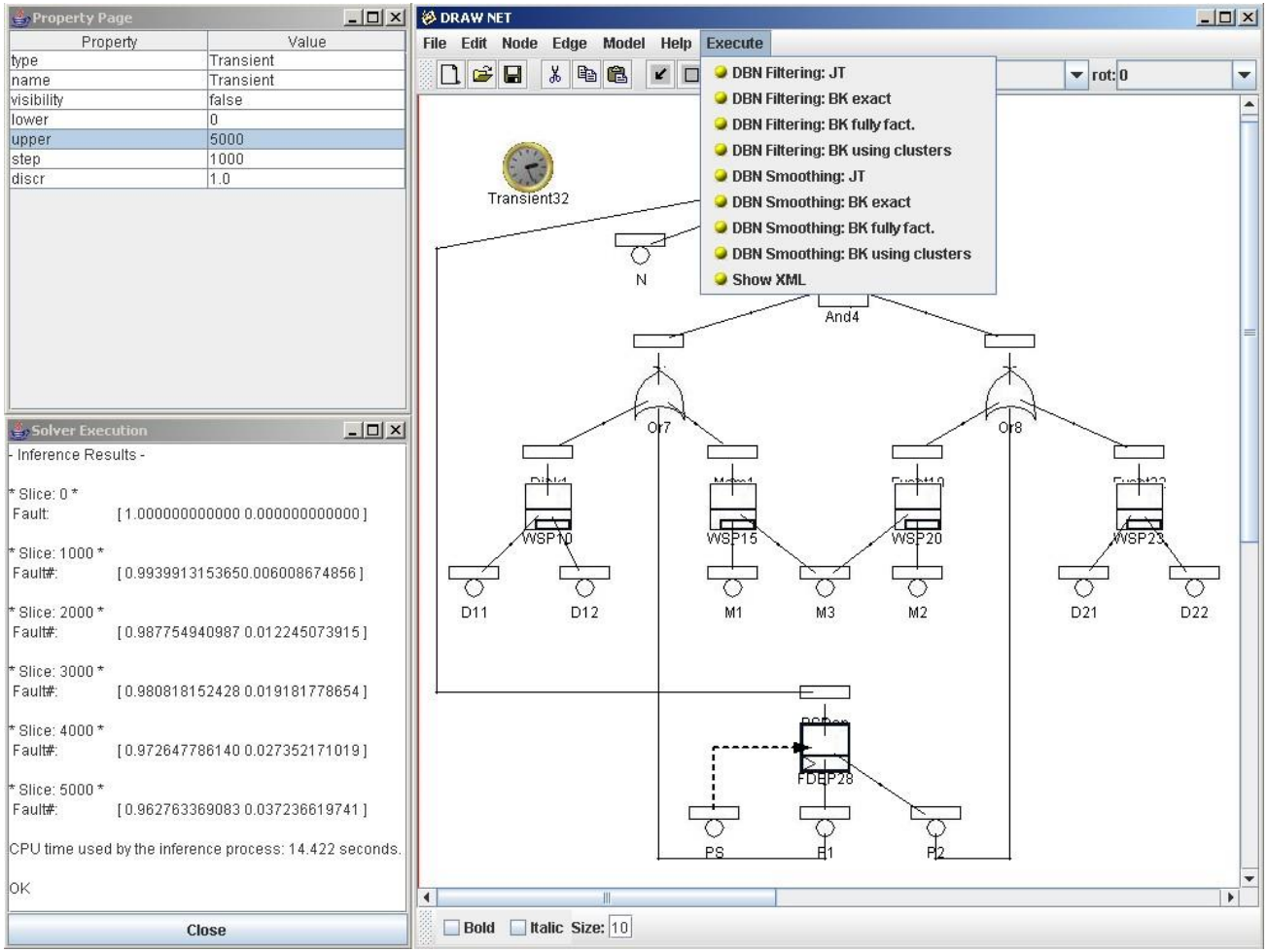
DBN.xml

Results



# RADYBAN

Reliability Analysis with DYnamic BAyesian Networks

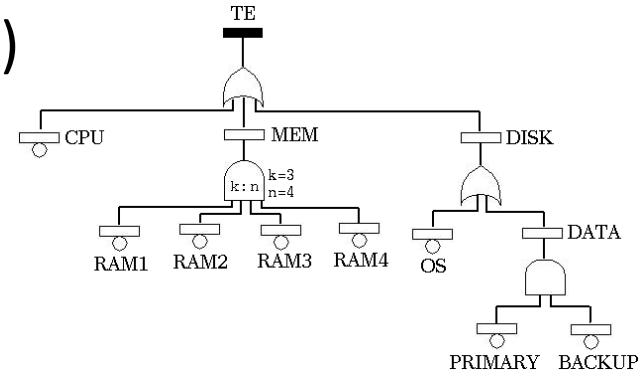



Draw-Net GUI
http://www.draw-net.com

INTEL PNL C++ libraries for DBN inference
http://sourceforge.net/projects/openpnl/

# Fault Tree extensions

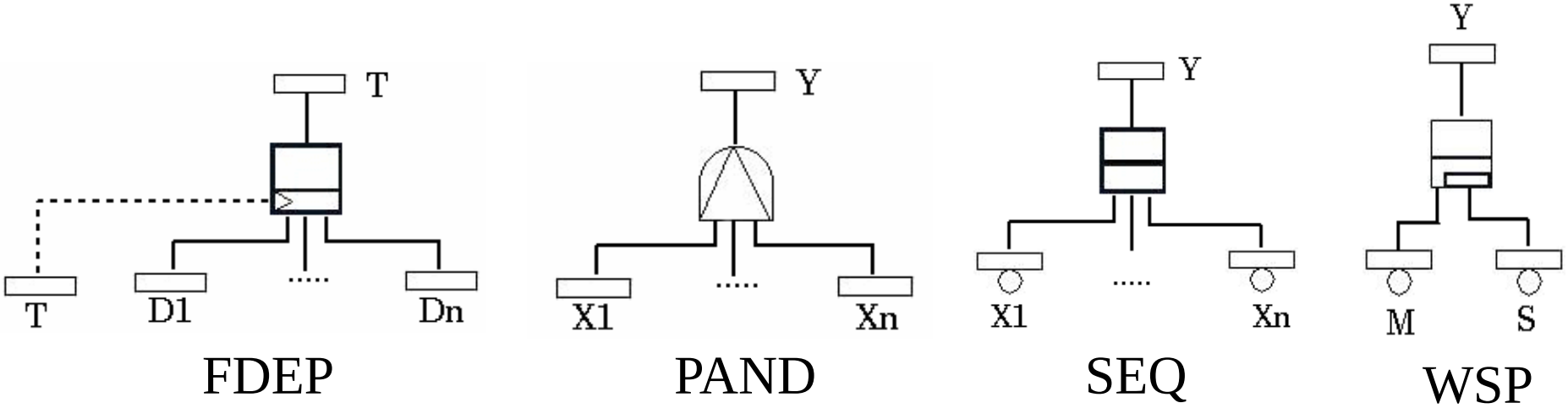
- FT modeling power is limited by several assumptions:
  - Events as binary variables (working / failed)
  - basic events independency
  - Only Boolean gates




- Extensions to the FT formalism in the literature:
  - *Parametric Fault Tree* (PFT)
  - ***Dynamic Fault Tree*** (DFT)
  - *Repairable Fault Tree* (RFT)
  - *Non deterministic Repairable Fault Tree* (NdRFT)
  - …

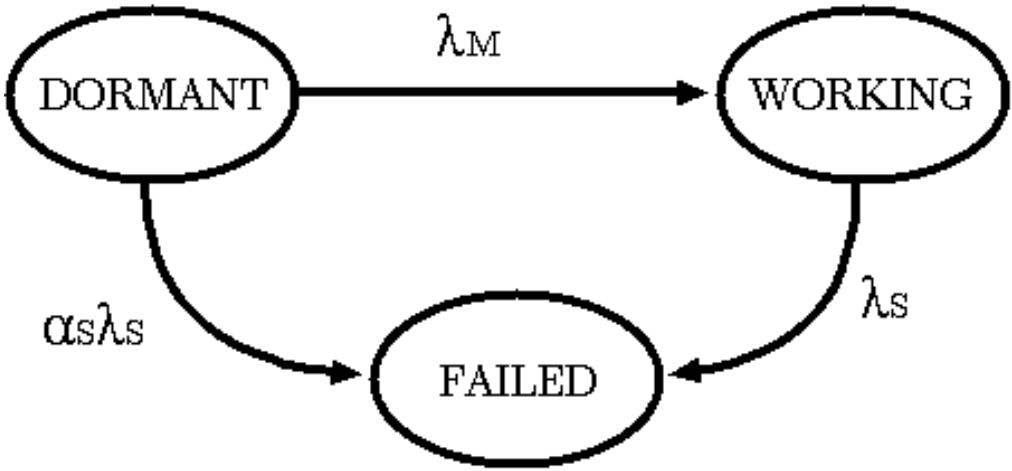
# Dynamic Fault Trees

- A dependency arises when the failure behaviour of a component depends on the state of another item (component, subsystem, system).
- DFTs are characterized by the dynamic gates
  - Functional dependencies (FDEP gate)
  - Temporal dependencies (PAND gate, SEQ gate)
  - (Warm) spare components (WSP gate): multi-state components



# Spare components

- M is the main component; S is its spare component.
- States of S:
  - Stand-by (dormant): $\alpha_s\lambda_s$
  - Working: $\lambda_s$
  - Failed
- $\lambda_s$ is the failure rate
- $\alpha_s$ is the dormancy factor



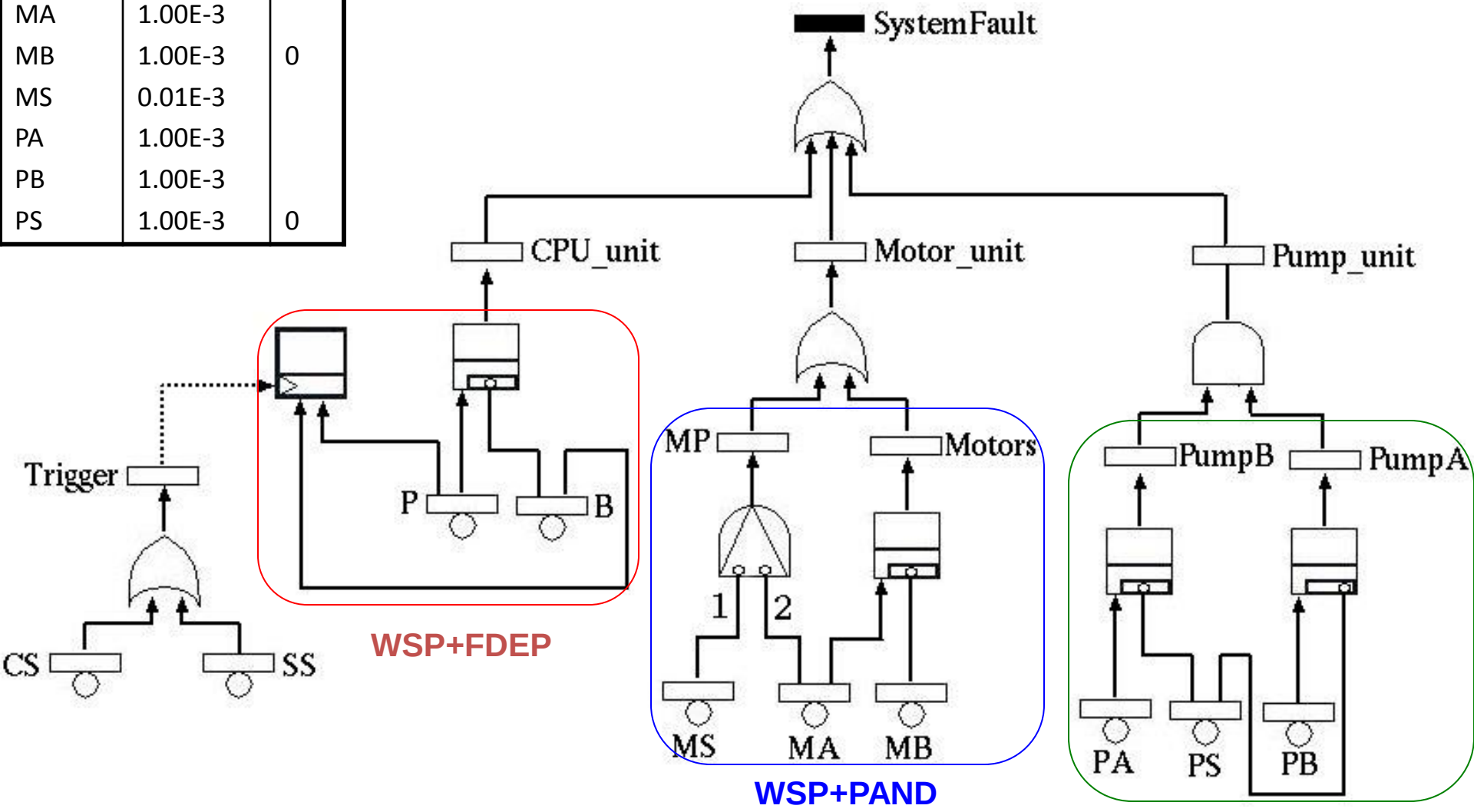

- Warm spare: $0 < \alpha < 1$
- Cold spare: $\alpha = 0$
- Hot spare: $\alpha = 1$

# Cardiac Assist System

- The failure of either one of the modules causes the whole system failure:
  - The CPU module consists of the primary cpu P and a warm spare B:
    - Both P and B are functionally dependent on a cross switch CS and a system supervision SS
    - Both P and B are considered as repairable
  - The Motor module consists of the primary motor MA and a cold spare MB:
    - MB turns into operation when the MA fails, because of a motor switching component MS
      - if MS fails before MA, then the spare cannot become operational
  - The Pump module is composed by two primary pumps PA and PB running in parallel and a cold spare PS

| Comp. | λ (1/h) | α |
|---|---|---|
| P | 0.50E-3 | |
| B | 0.50E-3 | 0.5 |
| CS | 0.20E-3 | |
| SS | 0.20E-3 | |
| MA | 1.00E-3 | |
| MB | 1.00E-3 | 0 |
| MS | 0.01E-3 | |
| PA | 1.00E-3 | |
| PB | 1.00E-3 | |
| PS | 1.00E-3 | 0 |

# DFT of the case study



# DFT analysis

- DFT relaxes assumptions holding for FT
- DFT analysis must capture the system evolution during the time. Solutions:
  - DFT → BDD + CTMC (modular approach)
    - Dynamic module → Continuous Time Markov Chains (CTMC)
      - Univ. of Virginia
    - Dynamic module → (Colored) Stochastic Petri Nets → CTMC
      - Univ. del Piemonte Orientale
  - DFT → algebraic formula including ◁ operator
      - ENS Cachan
  - DFT → I/O Interactive Markov Chains
      - Univ. of Twente
  - **DFT → Dynamic Bayesian Networks (DBN)**

# DBN for DFT analysis

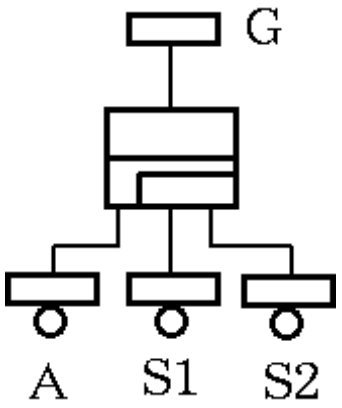
- DBN remove the assumption on binary events
  - Multistate components
- DBN remove the assumption on statistical independence
  - Event dependency
- DBN remove the assumption on Boolean gates (AND, OR)
  - Noisy OR, noisy AND
  - Dynamic gates
- DBN provide a more flexible forward and backward analysis, possibly based on observations
  - Forward (predictive) analysis: Pr(TE), Pr(Sub), Pr(TE|A), Pr(Sub|A)
  - Backward (diagnostic) analysis: Pr(A|TE), Pr(Sub|TE), …
- DBN avoid the state space generation
  - The model does not enumerate all the system states and transitions

# DFT conversion into DBN

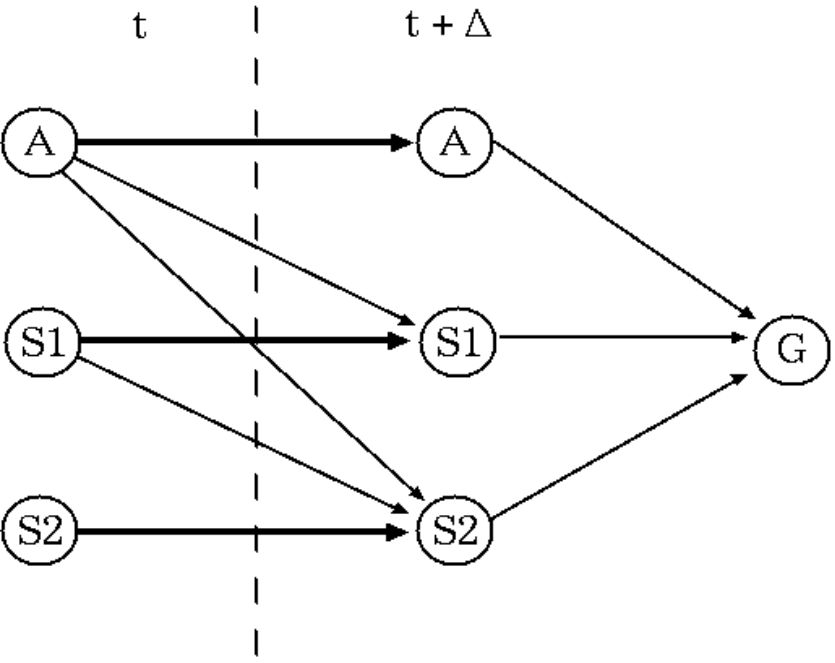
- Modular approach:
  - First, every single gate is converted into DBN
  - Then, the resulting DBNs are connected together in correspondance to the nodes they share.
    - Connection of DBN1 with DBN2
    - An adjustment to the CPT of a node is required when new arcs enter the node:
      - add all the parents derived from DBN1 and DBN2 as columns in the new CPT;
      - in every entry of the table, set the probability of failure of the node to the maximum between the corresponding entries of the CPT in DBN1 and in DBN2

- The connection of all the DBNs corresponding to the single gates, provides the DBN expressing the DFT model.

# Warm Spare gate




- A is the main component
  - failure rate: $\lambda$
- S1, S2 are the warm spare components
  - stand by → $\alpha\lambda$  $\alpha$ is the dormancy factor ($0<\alpha<1$)
  - working → $\lambda$

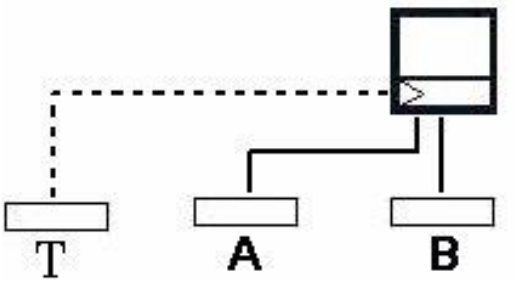



$$\Pr\{A(t+\Delta)=1|A(t)=1\}=1$$
$$\Pr\{A(t+\Delta)=1|A(t)=0\}=1-e^{-\lambda_A \Delta}$$

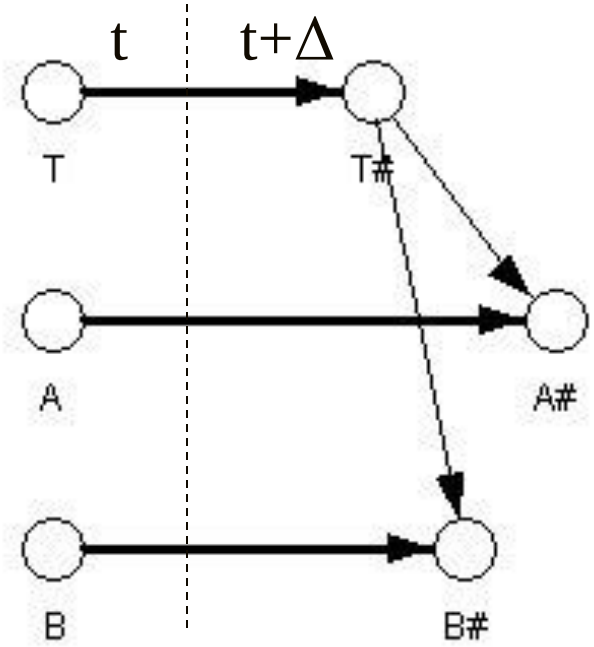
$$\Pr\{S1(t+\Delta)=1|S1(t)=1\}=1$$
$$\Pr\{S1(t+\Delta)=1|A(t)=0,S1(t)=0\}=1-e^{-\alpha\lambda_{S1} \Delta}$$
$$\Pr\{S1(t+\Delta)=1|A(t)=1,S1(t)=0\}=1-e^{-\lambda_{S1} \Delta}$$

$$\Pr\{S2(t+\Delta)=1|S2(t)=1\}=1$$
$$\Pr\{S2(t+\Delta)=1|A(t)=0,S1(t)=0,S2(t)=0\}=1-e^{-\alpha\lambda_{S2} \Delta}$$
$$\Pr\{S2(t+\Delta)=1|A(t)=0,S1(t)=1,S2(t)=0\}=1-e^{-\alpha\lambda_{S2} \Delta}$$
$$\Pr\{S2(t+\Delta)=1|A(t)=1,S1(t)=0,S2(t)=0\}=1-e^{-\alpha\lambda_{S2} \Delta}$$
$$\Pr\{S2(t+\Delta)=1|A(t)=1,S1(t)=1,S2(t)=0\}=1-e^{-\lambda_{S2} \Delta}$$

# Functional Dependency gate

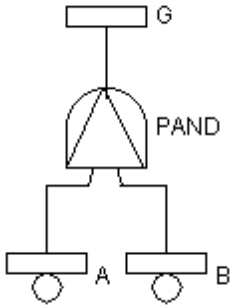


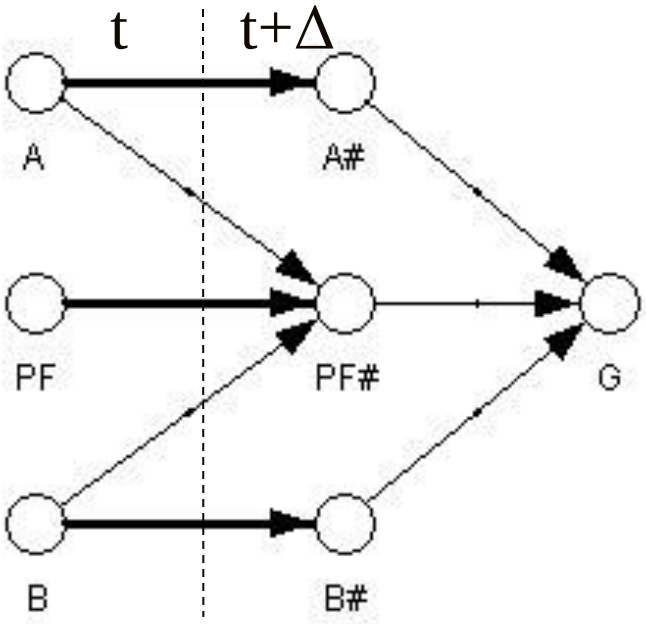

- T is the trigger event
  - failure rate: $\lambda_T$
- A, B are the dependent events
  - failure rate: $\lambda_A$, $\lambda_B$




$$\Pr\{T(t+\Delta)=1|T(t)=1\}=1$$
$$\Pr\{T(t+\Delta)=1|T(t)=0\}=1-e^{-\lambda_T \Delta t}$$
$$\Pr\{A(t+\Delta)=1|A(t)=1\}=1$$
$$\Pr\{A(t+\Delta)=1|A(t)=0,T(t+1)=0\}=1-e^{-\lambda_A \Delta t}$$
$$\Pr\{A(t+\Delta)=1|A(t)=0,T(t+1)=1\}=p_{dep}(=1)$$

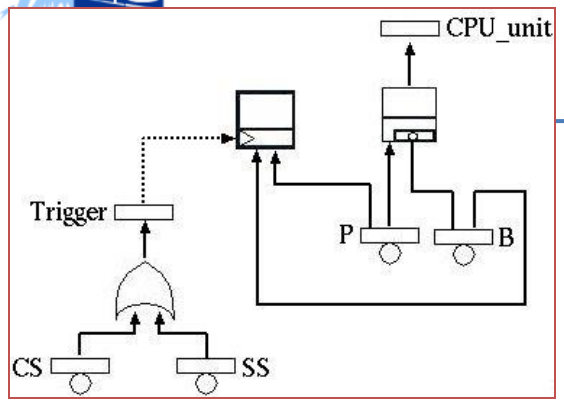
# Priority AND gate

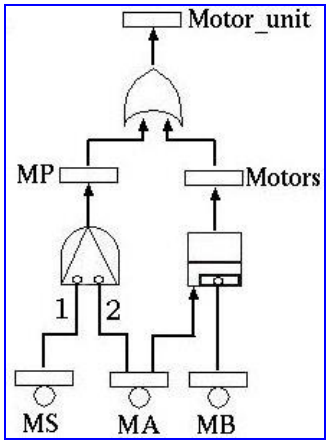


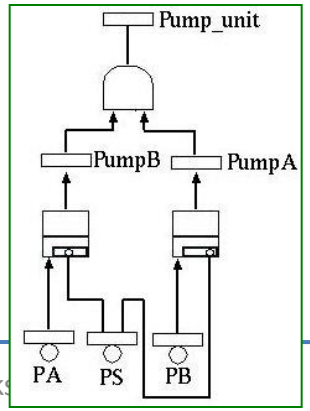



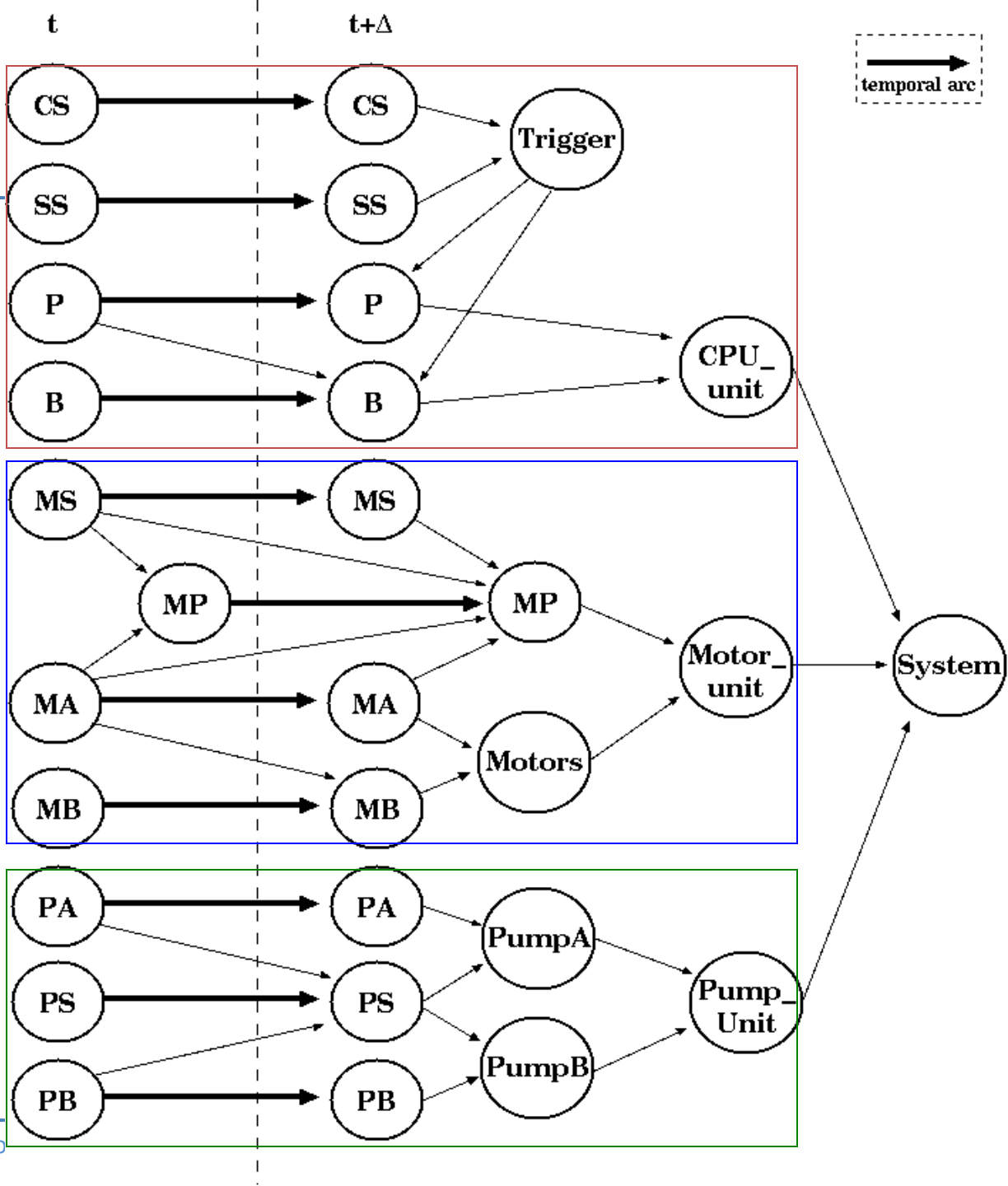

$$\Pr\{A(t+1)=1|A(t)=1\}=1$$
$$\Pr\{A(t+1)=1|A(t)=0\}=1-e^{-\lambda_A \Delta t}$$
$$\Pr\{B(t+1)=1|B(t)=1\}=1$$
$$\Pr\{B(t+1)=1|B(t)=0\}=1-e^{-\lambda_B \Delta t}$$
$$\Pr\{PF(t+1)=1|*,PF(t)=1\}=0$$
$$\Pr\{PF(t+1)=1|\ A(t)=0,\ B(t)=0,PF(t)=0\}=0$$
$$\Pr\{PF(t+1)=1|\ A(t)=1,\ B(t)=0,PF(t)=0\}=1$$
$$\Pr\{PF(t+1)=1|\ A(t)=0,\ B(t)=1,PF(t)=0\}=0$$
$$\Pr\{PF(t+1)=1|\ A(t)=1,\ B(t)=1,PF(t)=0\}=1$$

# DBN→DFT

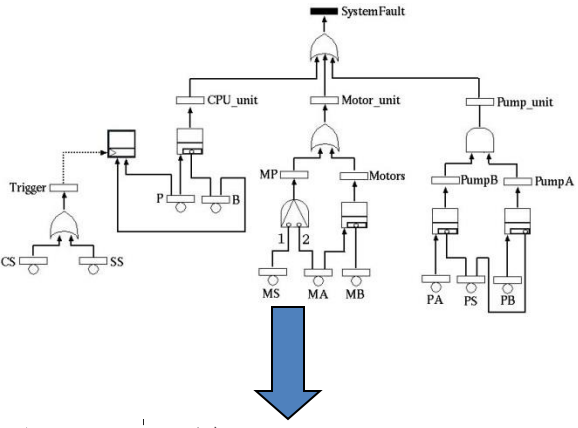

CPU_unit
Trigger
P
B
CS
SS

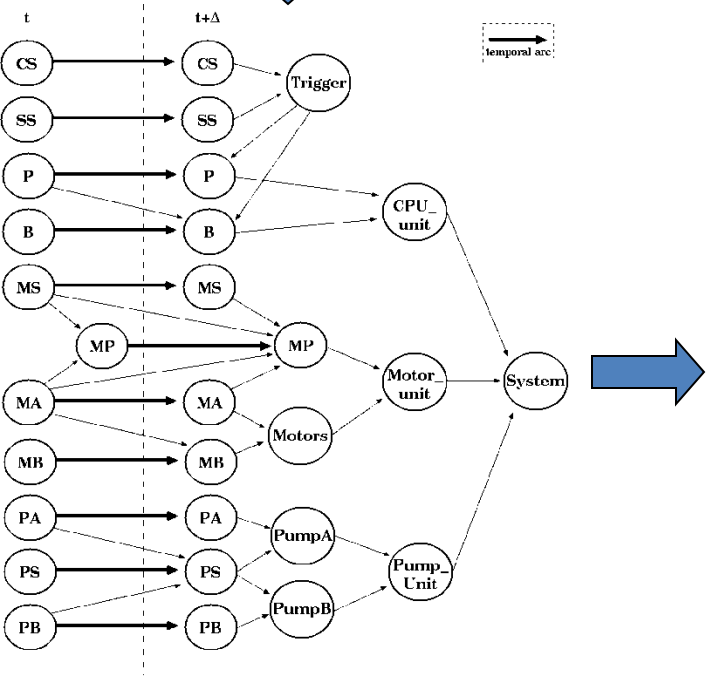

Motor_unit
MP
Motors
1
2
MS
MA
MB


Pump_unit
PumpB
PumpA
PA
PS
PB


t
t+Δ
temporal arc
CS
SS
P
B
Trigger
CPU_ unit
MS
MP
MA
MB
Motors
Motor_ unit
PA
PS
PB
PumpA
PumpB
Pump_ Unit
System

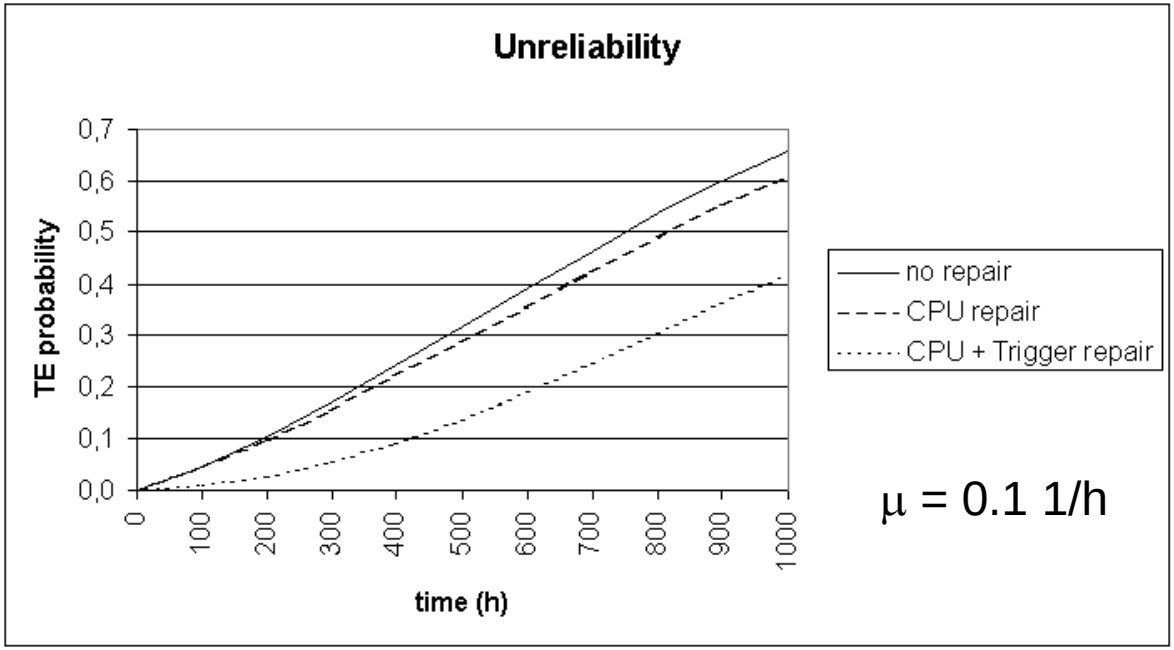
# Inference results




Case of filtering

with no observations

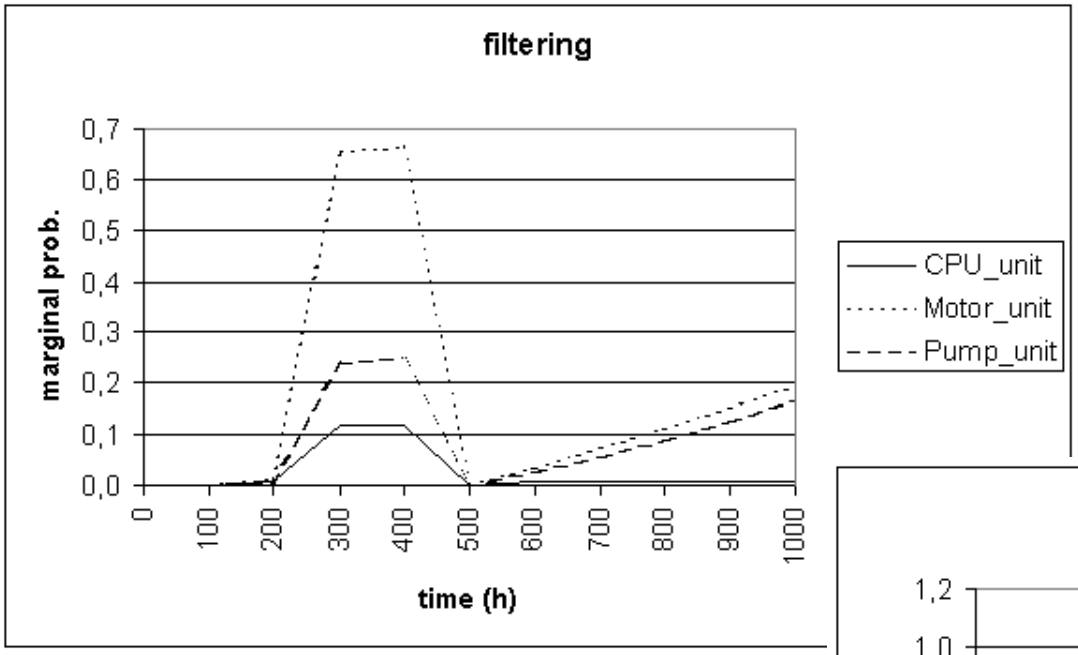


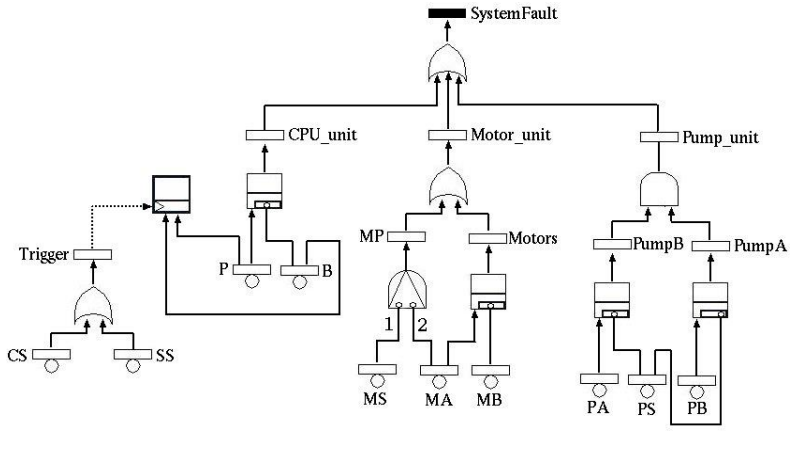
# Results comparison

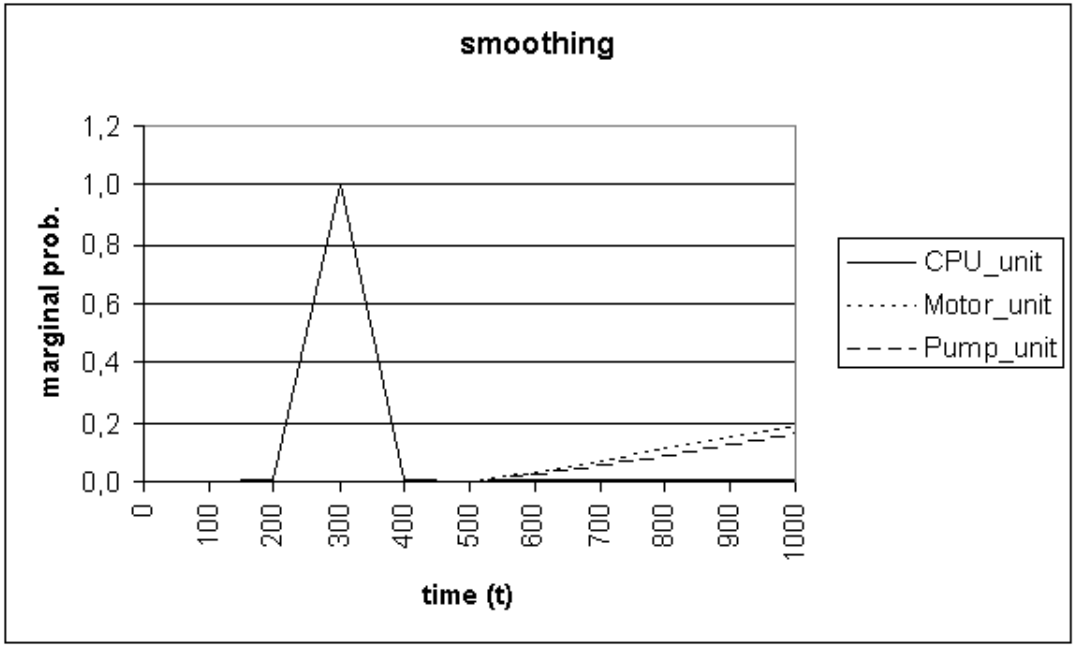
| Time (h) | *RADYBAN* ($k = 1$) | *RADYBAN* ($k = 0.1$) | *Galileo* |
|---|---|---|---|
| 100 | 0.045978 | 0.046026 | 0.0460314 |
| 200 | 0.103124 | 0.103214 | 0.103222 |
| 300 | 0.169204 | 0.169327 | 0.169336 |
| 400 | 0.241328 | 0.241474 | 0.241483 |
| 500 | 0.316482 | 0.316645 | 0.316651 |
| 600 | 0.391893 | 0.392060 | 0.392066 |
| 700 | 0.465241 | 0.465408 | 0.465411 |
| 800 | 0.534745 | 0.534908 | 0.534908 |
| 900 | 0.599169 | 0.599322 | 0.59932 |
| 1000 | 0.657763 | 0.657908 | 0.6579 |

| Time (h) | *RADYBAN* | | *DRPFTproc* | |
|---|---|---|---|---|
| | CPU repair | CPU + Trigger repair | CPU repair | CPU + Trigger repair |
| 100 | 0.044283796102 | 0.011243030429 | 0.0443301588 | 0.0112820476 |
| 200 | 0.096916869283 | 0.027566317469 | 0.0951982881 | 0.0276517226 |
| 300 | 0.156659856439 | 0.054836865515 | 0.155093539 | 0.0549629270 |
| 400 | 0.221550568938 | 0.091957211494 | 0.220137459 | 0.0921166438 |
| 500 | 0.289382189512 | 0.137252241373 | 0.288119742 | 0.137437204 |
| 600 | 0.358023554087 | 0.188778832555 | 0.356905021 | 0.188981668 |
| 700 | 0.425606846809 | 0.244557544589 | 0.424624354 | 0.244770740 |
| 800 | 0.490624904633 | 0.302729338408 | 0.489768367 | 0.302945892 |
| 900 | 0.551952958107 | 0.361649900675 | 0.551211316 | 0.361864672 |
| 1000 | 0.608829379082 | 0.419938921928 | 0.608191065 | 0.420148205 |

# Inference with observations




- P, B, CS, SS are repairable
- The system was observed
  - operational at $t_1$=100 h
  - failed at $t_2$=300 h
  - operational $t_3$=500 h

# Joint probabilities assuming observations

| Time (h) | 0,0,0 | 0,0,1 | 0,1,0 | 0,1,1 |
|---|---|---|---|---|
| 100 | **1.000000** | 0.000000 | 0.000000 | 0.000000 |
| 200 | **0.977576** | 0.003501 | 0.012862 | 0.000046 |
| 300 | 0.000000 | 0.228510 | **0.643095** | 0.007708 |
| 400 | 0.110162 | 0.224175 | **0.637081** | 0.022560 |
| 500 | **1.000000** | 0.000000 | 0.000000 | 0.000000 |
| 600 | **0.934621** | 0.024475 | 0.033999 | 0.000890 |
| 700 | **0.870357** | 0.051434 | 0.068166 | 0.004028 |
| 800 | **0.803337** | 0.079515 | 0.101124 | 0.010009 |
| 900 | **0.735453** | 0.107478 | 0.131794 | 0.019260 |
| 1000 | **0.668297** | 0.134277 | 0.159387 | 0.032024 |
| | 1,0,0 | 1,0,1 | 1,1,0 | 1,1,1 |
| 100 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 200 | 0.005916 | 0.000021 | 0.000078 | 0.000000 |
| 300 | 0.115366 | 0.001383 | 0.003891 | 0.000047 |
| 400 | 0.000673 | 0.001357 | 0.003855 | 0.000137 |
| 500 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 600 | 0.005655 | 0.000148 | 0.000206 | 0.000006 |
| 700 | 0.005267 | 0.000311 | 0.000413 | 0.000024 |
| 800 | 0.004861 | 0.000481 | 0.000612 | 0.000061 |
| 900 | 0.004450 | 0.000650 | 0.000798 | 0.000117 |
| 1000 | 0.004044 | 0.000813 | 0.000964 | 0.000194 |

# Conclusions

- Several classes of model can be applied in the Dependability/Reliability field
- DBN extend BN introducing a temporal dimension
  - Several kinds of inference: filtering, prediction, smoothing
- RADYBAN allows to
  - Edit and analyze a DBN
  - Edit and analyze a DFT through conversion into DBN
    - Possibility of computing measures conditioned by observations
    - DBN avoid the state space generation
    - DFT can be an high-level formalism to generate DBN

# Future work

- Implementing in RADYBAN the possibility of modeling the repair of subsystems in the DFT
  - Several repair policies are already defined, together with the corresponding conversion rules DFT→DBN

- VeriFIM project: collaboration with Alenia, ESA
  - Development of a DBN analyzer based on Junction Tree (JT) algorithm, to be run on board of a rover
    - Goals: diagnosis, prediction and recovery of the rover according to the observations coming from the rover's sensors
  - The DBN (and the JT) will be derived from the DFT of the rover by means of RADYBAN (off board)

# REFERENCES


DBN P. Weber, L. Jouffe, “Reliability modelling with dynamic Bayesian networks”, *Proc. SAFEPROCESS 2003*, Washington D.C., 2003.

DBN Murphy K. Dynamic Bayesian networks: representation, inference and learning. PhD thesis, UC Berkley; 2002.

DFT J. B. Dugan, S. J. Bavuso, M. A. Boyd, “Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems”, IEEE Trans. on Reliability, vol 41, 1992, pp 363-377.

DFT R. Manian, D. W. Coppit, K. J. Sullivan, J. B. Dugan, “Bridging the Gap Between Systems and Dynamic Fault Tree Models”, Proc. Annual Reliability and Maintainability Symposium, 1999, pp 105-111.

~ ~ ~

DFT→ DBN S. Montani, L. Portinale, A. Bobbio, D. Codetta Raiteri, *“RADYBAN: a tool for Reliability Analysis of Dynamic Fault Trees through Conversion into Dynamic Bayesian Networks”*, Reliability Engineering and System Safety, vol. 93(7), pages 922-932, Elsevier, July 2008

DFT→ DBN L. Portinale, D. Codetta-Raiteri, S. Montani, *"Supporting Reliability Engineers in Exploiting the Power of Dynamic Bayesian Networks”*, International Journal of Approximate Reasoning, vol. 51(2), pages 179-195, Elsevier, Jan. 2010.

DBN applications D. Codetta-Raiteri, A. Bobbio, S. Montani, L. Portinale, *"A dynamic Bayesian network based framework to evaluate cascading effects in a power grid“,* Engineering Applications of Artificial Intelligence, Elsevier (to appear).

# REFERENCES (free download)


DFT→ DBN L. Portinale, A. Bobbio, D. Codetta-Raiteri, S. Montani, *"Compiling Dynamic Fault Trees into Dynamic Bayesian Networks: the RADYBAN tool"*, CEUR Workshop Proceedings, vol. 268, August 2007. Proceedings of the Bayesian Modeling Applications Workshop, Vancouver, Canada, July 2007, **http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-268/**

DFT→ DBN S. Montani, L. Portinale, A. Bobbio, M. Varesio, D. Codetta-Raiteri, *"DBNet, a tool to convert Dynamic Fault Trees into Dynamic Bayesian Networks"*, Dip. di Informatica, Univ. del Piemonte Orientale, TR-INF-2005-08-02-UNIPMN, August 2005, **http://www.di.unipmn.it/index.php?page=technical_reports&year=2005**

DFT→ DBN S. Montani, L.. Portinale, A. Bobbio, *"Dynamic Bayesian Networks for Modeling Advanced Fault Tree Features in Dependability Analisys"*, Dip. di Informatica, Univ. del Piemonte Orientale, TR-INF-2004-03-04-UNIPMN, March 2004, **http://www.di.unipmn.it/index.php?page=technical_reports&year=2004**

BN Luigi Portinale Helge Langseth, *"Bayesian Networks in Reliability"*, Dip. di Informatica, Univ. del Piemonte Orientale, TR-INF-2005-04-01-UNIPMN, April 2005, **http://www.di.unipmn.it/index.php?page=technical_reports&year=2005**