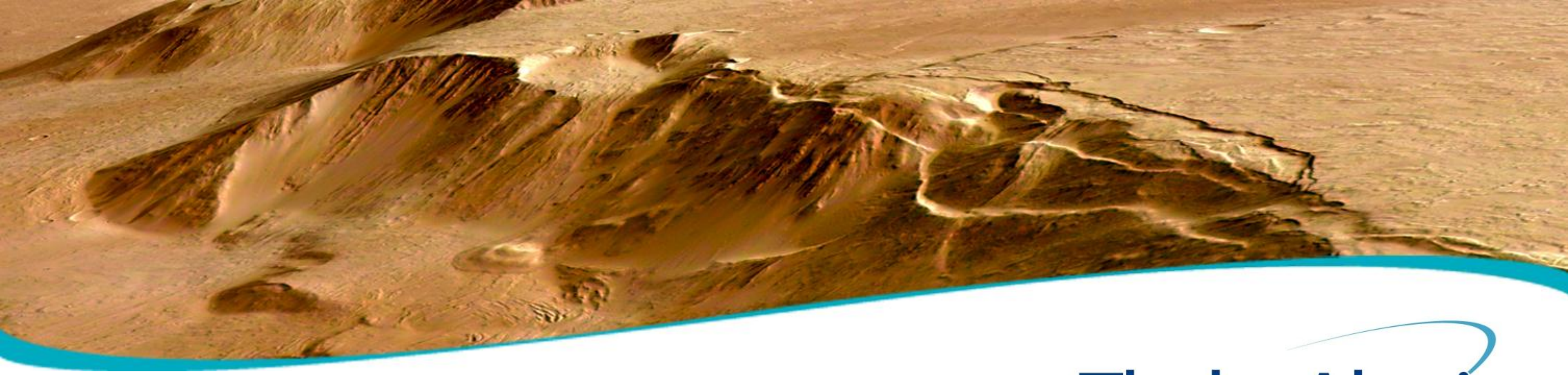
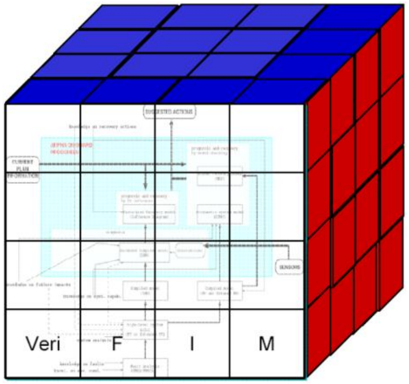


ThalesAlenia Space
A Thales / Finmeccanica Company



universitá degli studi del piemonte orientale AMEDEO AVOGADRO
Dipartimento di Informatica

Final Presentation

Verification of Failure Impact by Model Checking

Noordwijk, 16 December 2011

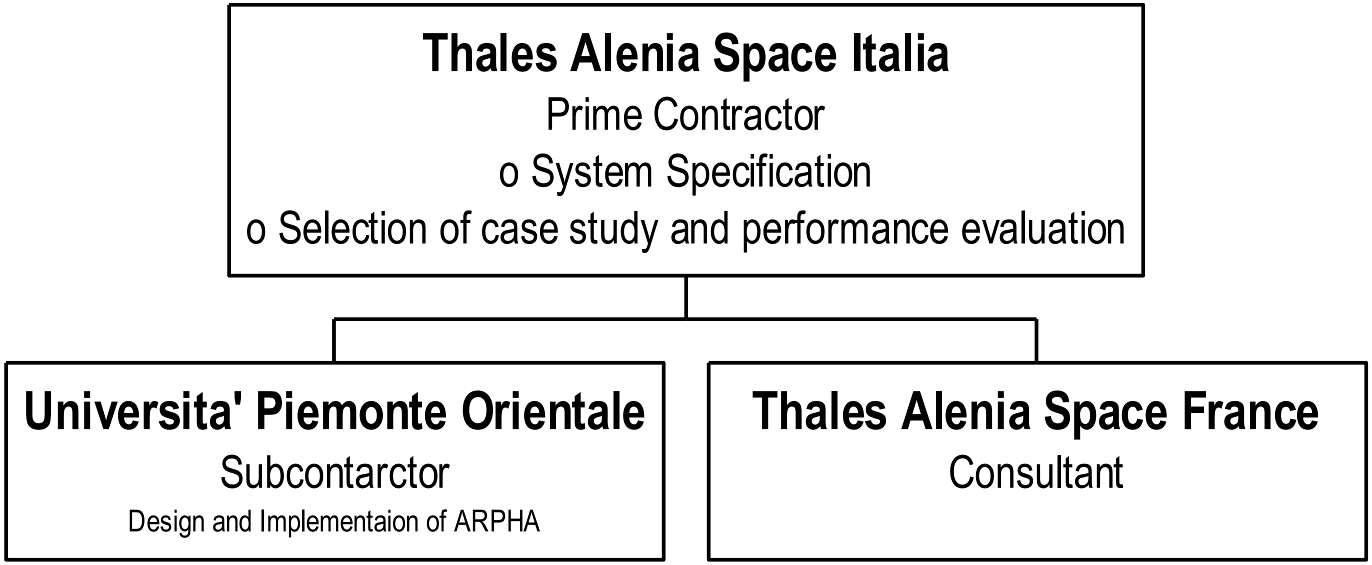
THALES

Agenda

- 14:45 15:00 Introduction
  - **Study Organization**
  - **Study Motivations**
  - **Study Objectives**
- 15:00 16:00 VeriFIM proposed solution
  - **Theoretical Aspect**
    - **Off-board process**
    - **On-board process**
  - **ARPHA implementation**
- 16:00 16:30 Evaluation of the approach
  - **Description of case study**
  - **Presentation of results**
- 16:30 16:45 Characterization of the approach
- 16:45 17:00 Industrial perspective
  - **Maturity of the approach**
  - **Pros, Cons and Gap to fill**
  - **HW/SW architecture**
- 17:00 17:15 Conclusion
- 17:15 17:30 Questions and Discussion

Introduction

Team Composition



Thales Alenia Space Italia

Prime Contractor

o System Specification

o Selection of case study and performance evaluation

Universita' Piemonte Orientale

Subcontarctor

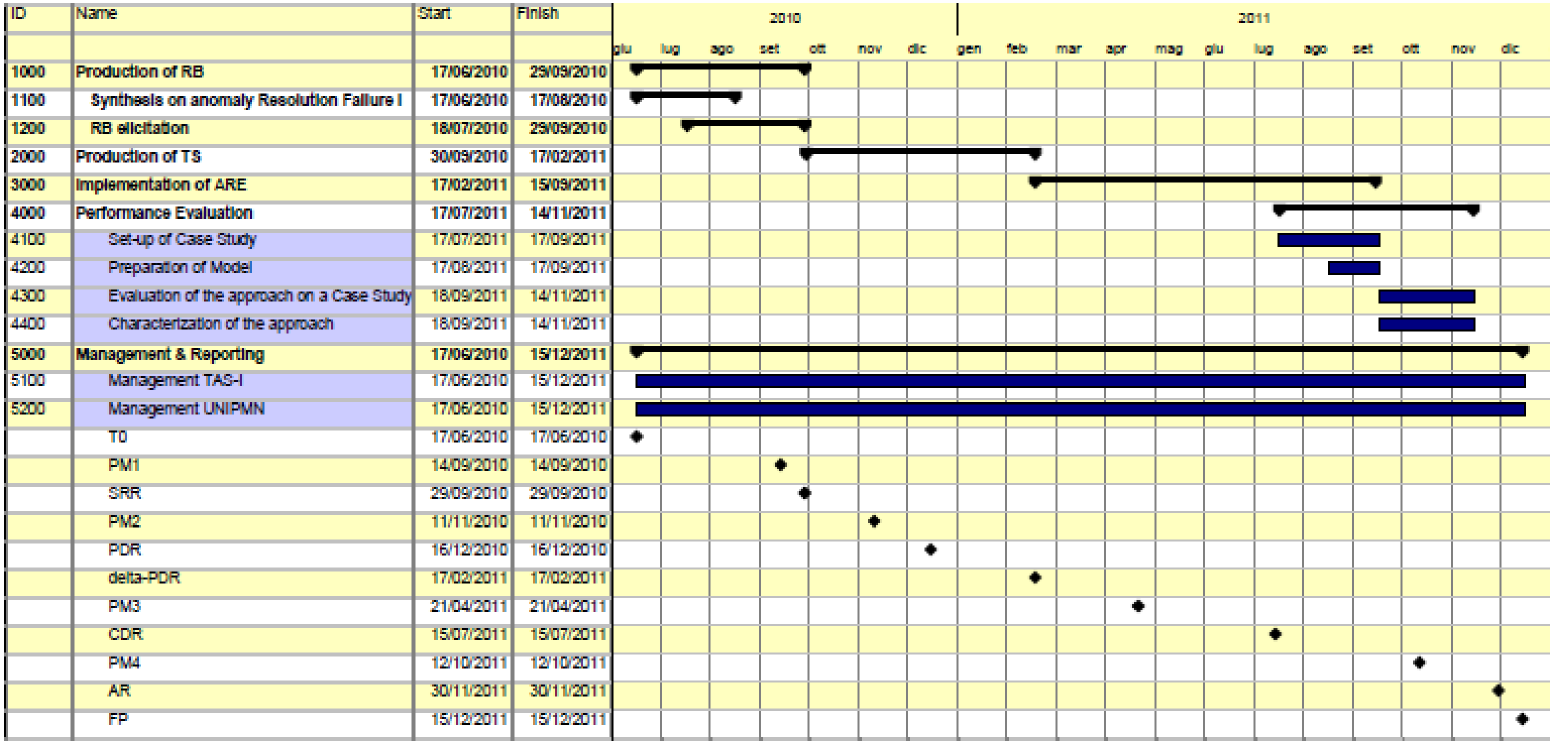
Design and Implementaion of ARPHA

Thales Alenia Space France

Consultant

Schedule

Start Date: 17/06/2010
End Date: 16/12/2011



ID	Name	Start	Finish
1000	Production of RB	17/06/2010	29/09/2010
1100	Synthesis on anomaly Resolution Failure I	17/06/2010	17/08/2010
1200	RB elicitation	18/07/2010	29/09/2010
2000	Production of TS	30/09/2010	17/02/2011
3000	Implementation of ARE	17/02/2011	15/09/2011
4000	Performance Evaluation	17/07/2011	14/11/2011
4100	Set-up of Case Study	17/07/2011	17/09/2011
4200	Preparation of Model	17/08/2011	17/09/2011
4300	Evaluation of the approach on a Case Study	18/09/2011	14/11/2011
4400	Characterization of the approach	18/09/2011	14/11/2011
5000	Management & Reporting	17/06/2010	15/12/2011
5100	Management TAS-I	17/06/2010	15/12/2011
5200	Management UNIPMN	17/06/2010	15/12/2011
	T0	17/06/2010	17/06/2010
	PM1	14/09/2010	14/09/2010
	SRR	29/09/2010	29/09/2010
	PM2	11/11/2010	11/11/2010
	PDR	16/12/2010	16/12/2010
	delta-PDR	17/02/2011	17/02/2011
	PM3	21/04/2011	21/04/2011
	CDR	15/07/2011	15/07/2011
	PM4	12/10/2011	12/10/2011
	AR	30/11/2011	30/11/2011
	FP	15/12/2011	15/12/2011

Motivations(1/2)

- **Currently employed FDIR operation is based on the design-time analysis of the faults and failure scenarios (e.g. FMEA, FTA) and run-time observation of the system operational status (health monitoring). It has the main objectives to timely detect the faults and to initiate the corresponding predefined recovery actions. If no corresponding action could be found, FDIR proceeds by executing the recovery actions to put the spacecraft into a known safe configuration and transfers control to the Ground operations for troubleshooting and planning the recovery actions.**
- **This approach is not always adequate for an autonomous system for the following reasons:**
  - Partial observability of system and environment does not allow for a certain identification of system status
  - Tradition FDIR cannot provide and utilize prognosis for the imminent failures
  - Automated FDIR procedures cannot leverage specific course of recovery based on the evaluation of causal knowledge of system and environment status
  - It is impossible to estimate the impact of the occurred faults and failures on the operational capabilities of the system
  - Reaction time does not always allow to wait a Ground recovery

Motivations(2/2)

- A new approach to on-board FDIR is needed which has the capability to reason about
  - anomalous observations based on the global knowledge of the system and its capabilities,
  - system environment,
  - and system-environment interaction in the presence of uncertainty.

- It has to provide the system with prognosis on the operational status to be taken into account for autonomous operational planning and to allow preventive recovery actions.

Study objectives

The global objective of this study is to demonstrate that integration of innovative technologies (i.e. model-based autonomy, model checking of stochastic hybrid models, run-time Dependability and Safety analysis, causal modelling, probabilistic calculus, Knowledge-Based Systems) in a unified modelling and autonomous reasoning framework may increase the achievable level of autonomy.

The main focus is on the autonomous anomaly resolution and prognostic pro-active FDIR capabilities.

Study objectives

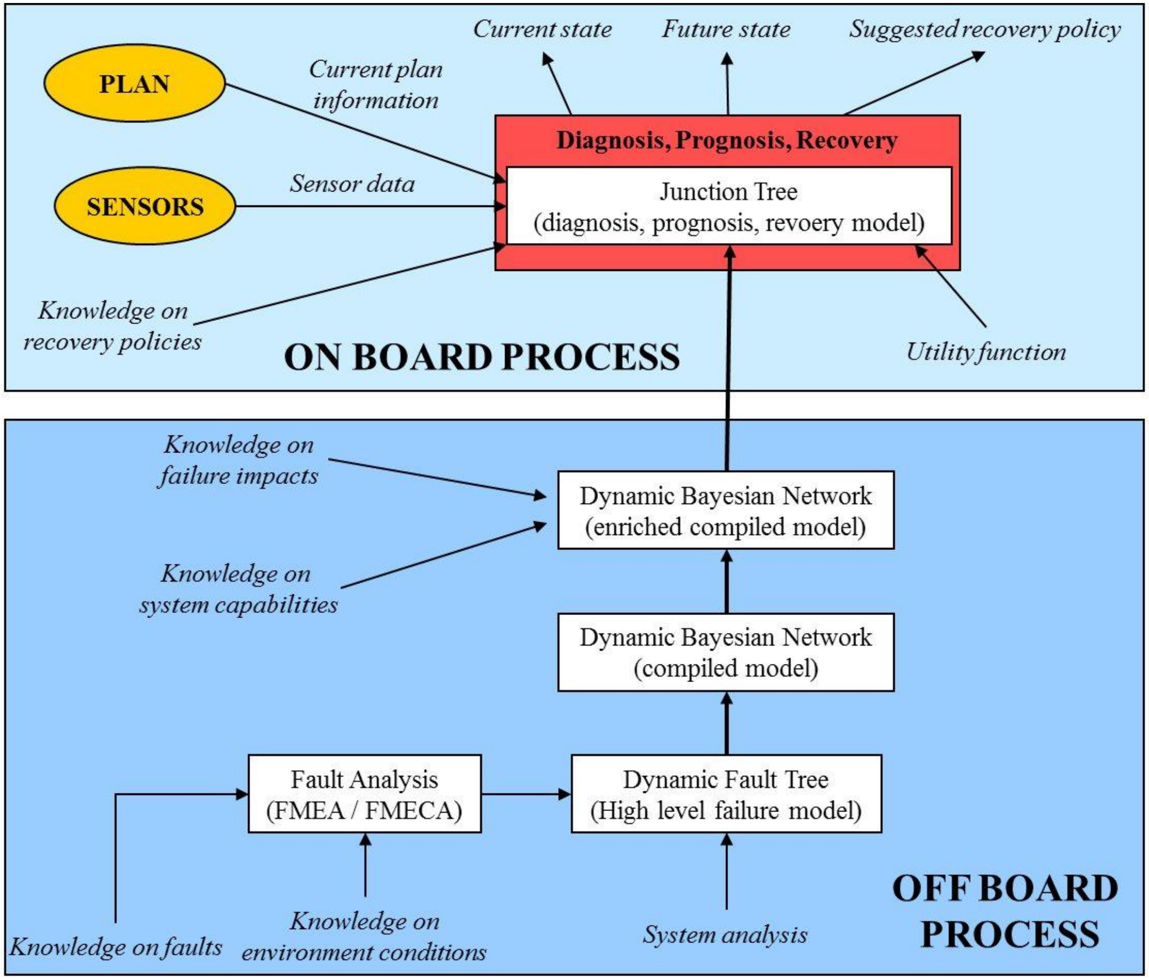
The global objective comprises the following sub-objectives:

- Evaluation and justification of an integrated and unified use of the stochastic hybrid model checking, causal probabilistic techniques and Knowledge-Based approaches, suited for on-board automated analysis, to increase the space systems level of autonomy in terms of anomaly resilience and autonomous recoverability;
- Definition of an integrated modelling framework for specification of the models suited for on-board autonomous reasoning to infer system Health, Dependability and Safety status and prognosis, and (preventive) anomaly resolution approaches;
- Development of an on-board software prototype, the Anomaly Resolution and Prognostic Health management for Autonomy (ARPHA), implementing the required autonomous reasoning and inference techniques, based on the use of stochastic hybrid model checking and probabilistic calculus approaches;
- Demonstration of the approach on case studies involving autonomous on-board systems and evaluation of the experimental results in terms of applicability, scalability, and performance;
- Evaluation of adequacy of the approach and developed technology for use in the context of critical on-board space systems

VeriFIM Proposed Solution

Off-board and on-board process

ARPHA



Theoretical aspects

Dynamic Fault Trees

As proposed by Joan Dugan et al. local dependencies can be included into a FT by defining a new class of gates, called Dynamic gates

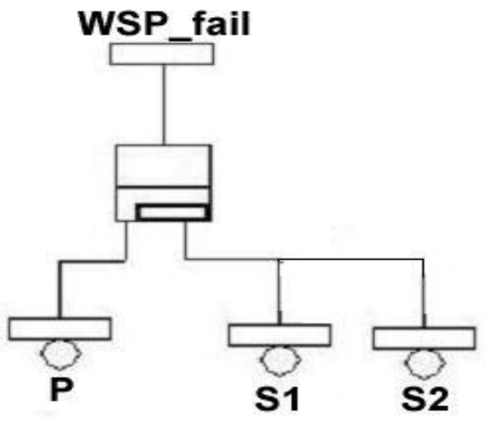
This extension has been called Dynamic Fault Tree (DFT)

- J. Bechta Dugan, S.J. Bavuso, and M.A. Boyd. Dynamic fault-tree models for fault-tolerant computer systems. *IEEE Trans Reliability*, 41:363.377, 1992.

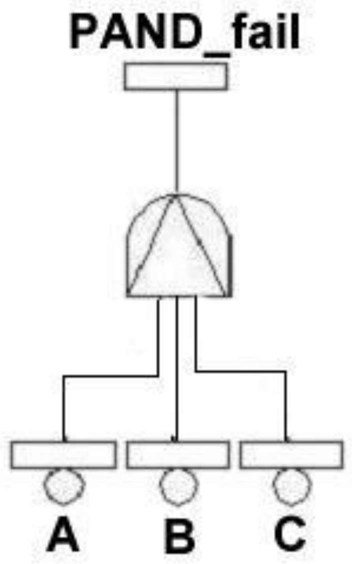
- J. Bechta Dugan, K.J. Sullivan, and D. Coppit. Developing a low-cost high quality software tool for dynamic fault-tree analysis. *IEEE Trans Reliability*, 49:49-59, 2000.

Dynamic Gates

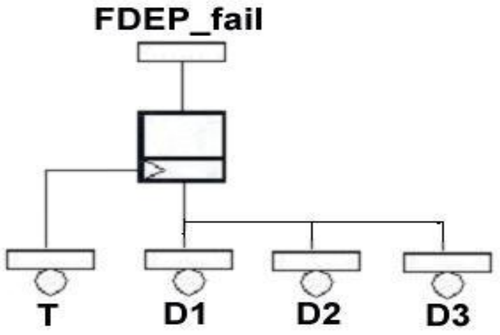
They model local dependencies among basic components or among their failure events.



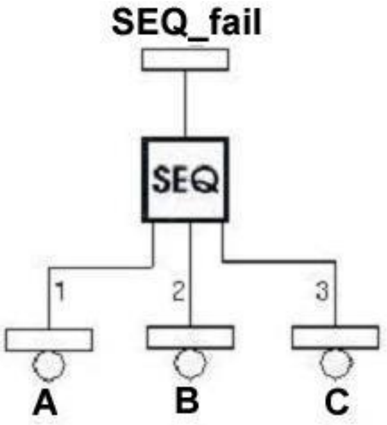
Warm Spare Gate



Priority And

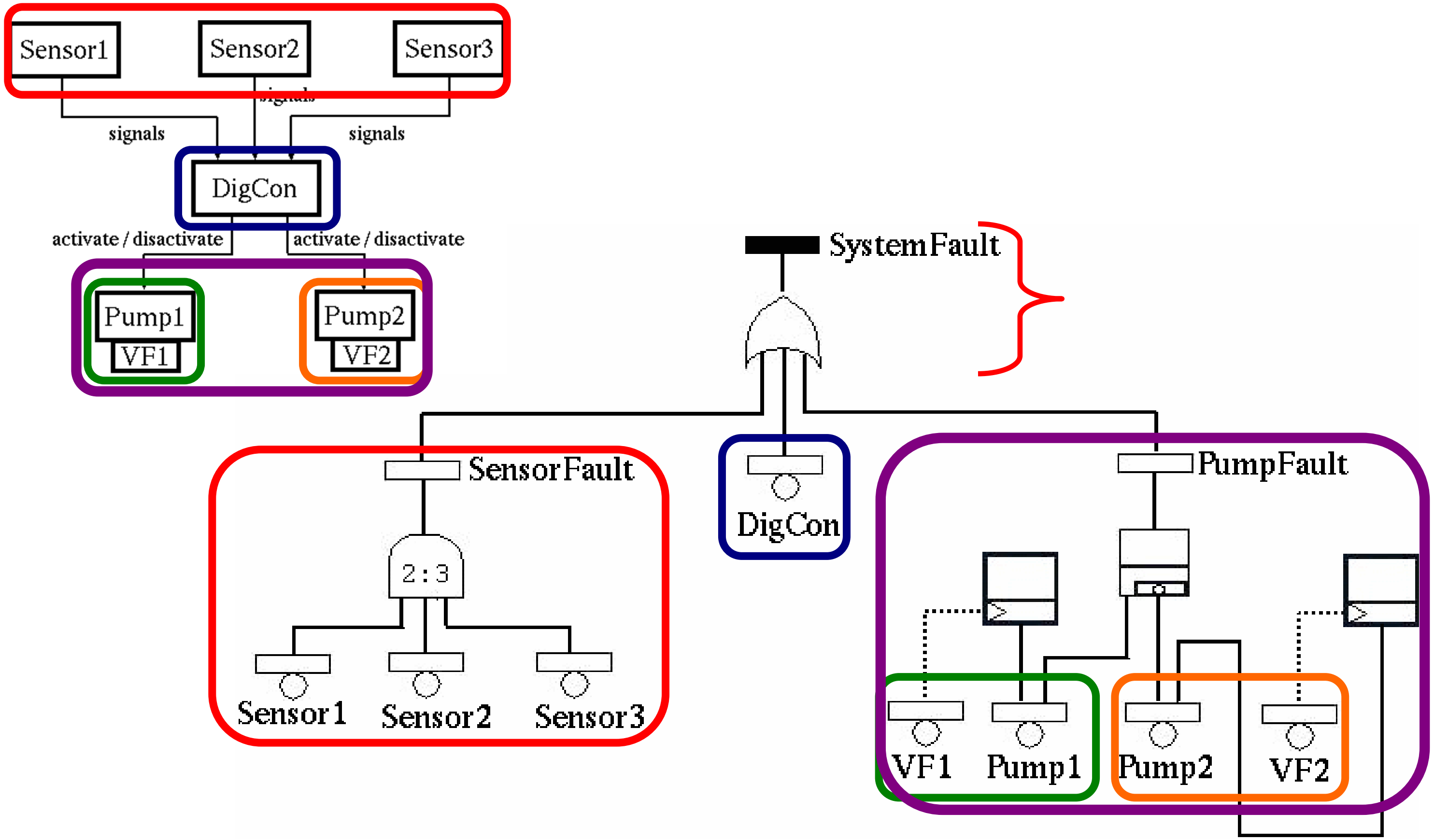


Functional Dependency Gate



Sequence Enforcing Gate

Example of DFT model



MESHKAT, L., DUGAN, J.B. and ANDREWS, J.D., 2000. Analysis of safety systems with on-demand and dynamic failure modes. In Proceedings of the Annual Reliability and Maintainability Symposium, Los Angeles, 24th-27th January, pp. 14-22 [DOI:10.1109/RAMS.2000.816277]

Static Models

- Bayesian Networks (aka Causal Networks, Probabilistic Networks, Belief Networks,…)
- Influence Diagrams (Decision Networks)

Dynamic Models

- Dynamic Bayesian Networks (2TBN)
- Dynamic Decision Networks

Emphasis: local dependencies

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** ***[J. Pearl, Probabilistic Reasoning in Inteligence Systems, Morgan Kaufmann, 1988]***
- **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**

BN: definition

A Bayesian Network is a pair <*G*,*P*> where

- ***G*** **is a Directed Acyclic Graph (DAG) with**
  - nodes representing (discrete) random variables
  - an oriented arc *X*→*Y* represents a dependency relation of *Y* from *X* (*X* influences *Y*, *Y* depends on *X*, *X* causes *Y*, etc…)
- ***P*** **is a probability distribution over the random variables represented by the nodes $X_1,\ldots X_n$ of the DAG such that**

$$P(X_1,\ldots X_n) = \prod_{i=1}^{n} P(X_i \mid Parent(X_i))$$

Specification of a CPT local to each node

Bayesian Networks

Diagnostic inference

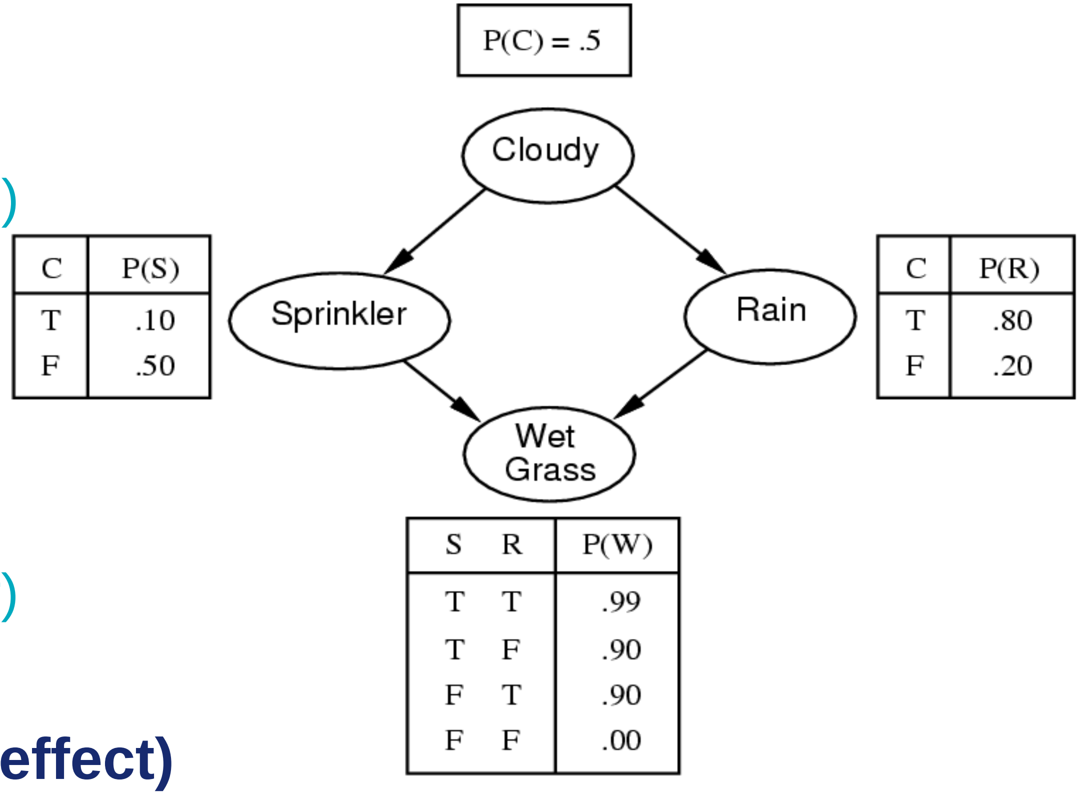
- **Pr(cause | effect)**
  - Pr(Sprinkler | Wet grass)
  - Pr(Cloudy | Wet grass)

Predictive inference

- **Pr(effect | cause)**
  - Pr(Wet grass | Cloudy)
  - Pr(Wet grass | Sprinkler)

Combined Inference

- **Pr(intermediate|cause, effect)**
  - Pr(Rain | Cloudy, Wet grass)



P(C) = .5

C	P(S)
T	.10
F	.50

C	P(R)
T	.80
F	.20

S	R	P(W)
T	T	.99
T	F	.90
F	T	.90
F	F	.00

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

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

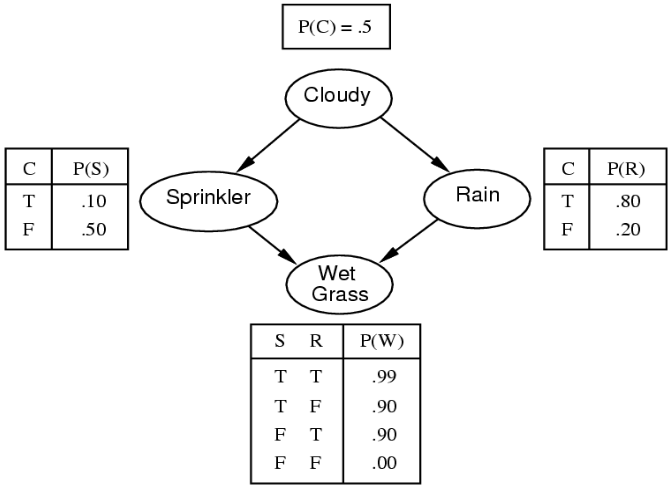
- X_1, …, X_n are network variables except Q, E.

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

- Y_1, …, Y_n are network variables except E.

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

Inference example



C	P(S)
T	.10
F	.50

C	P(R)
T	.80
F	.20

S	R	P(W)
T	T	.99
T	F	.90
F	T	.90
F	F	.00

Joint probability distribution

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

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

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

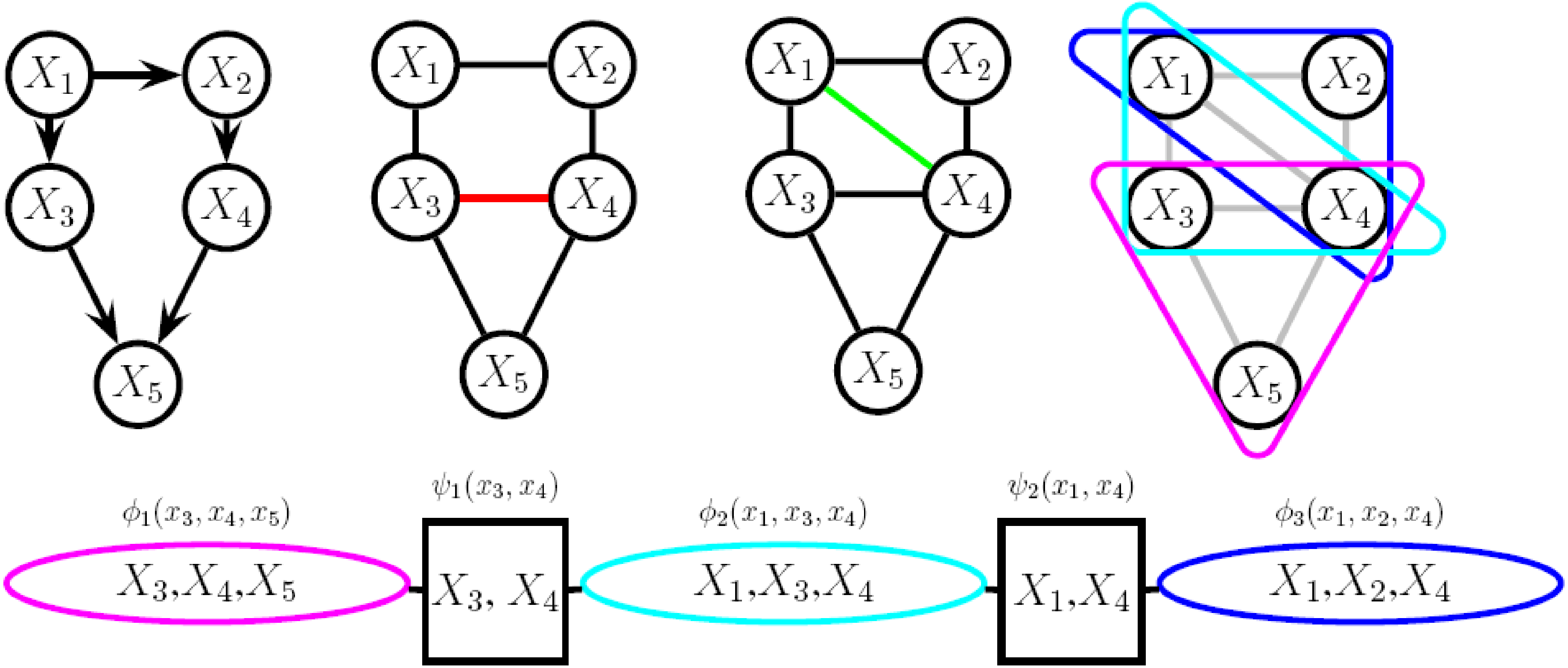
$$\vdots$$

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

Query:

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

Clustering Computation Scheme

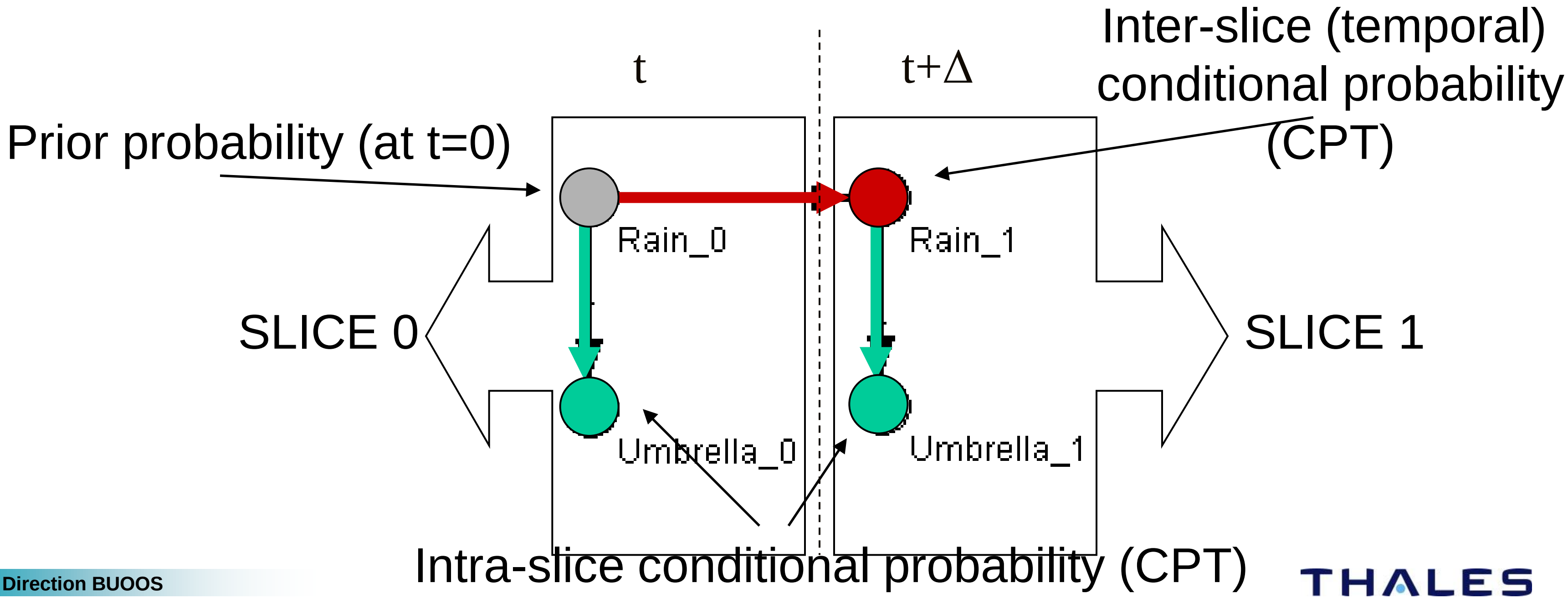


Junction (Join) tree

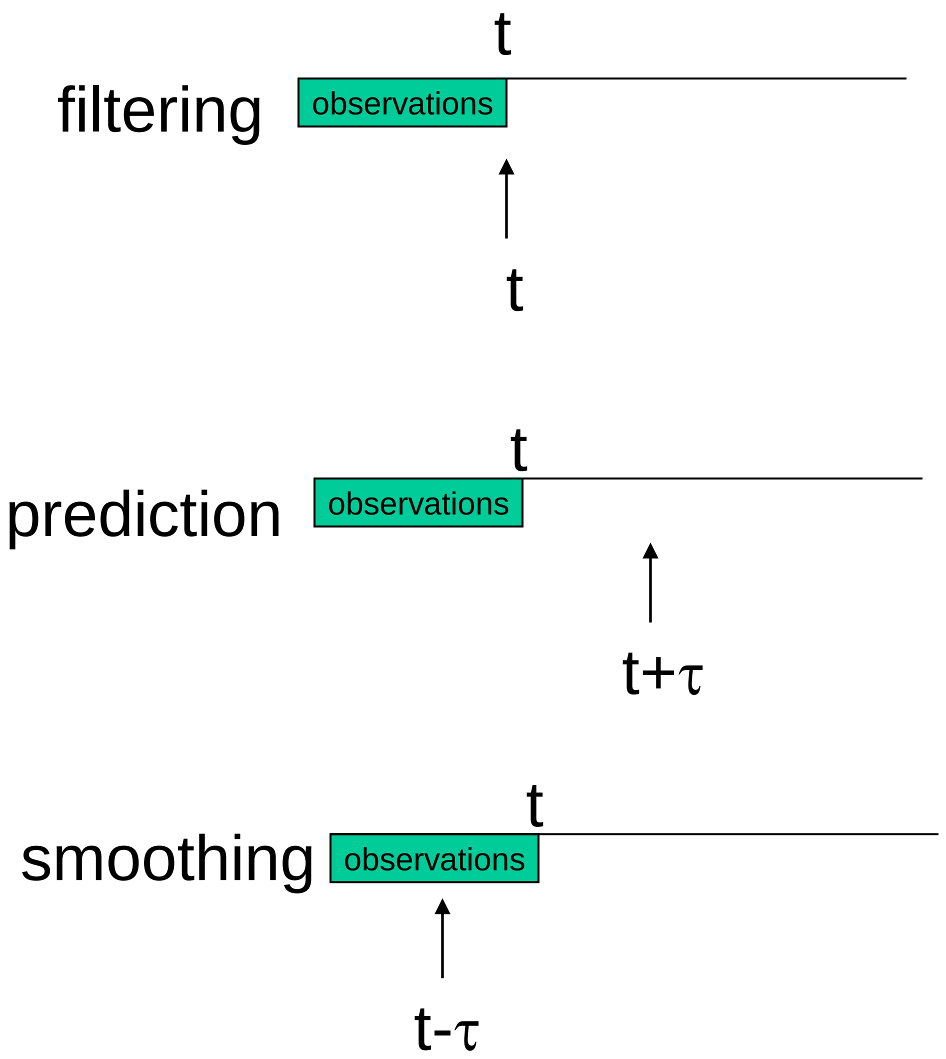
Advantage: dealing with 3 variables instead of 5

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


t
t+Δ
Inter-slice (temporal) conditional probability (CPT)
Prior probability (at t=0)
Rain_0
Rain_1
SLICE 0
SLICE 1
Umbrella_0
Umbrella_1
Intra-slice conditional probability (CPT)

Inference on DBN



Inference algorithms:

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

DFT conversion into DBN

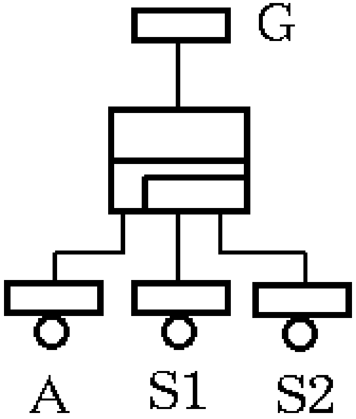
Modular approach:

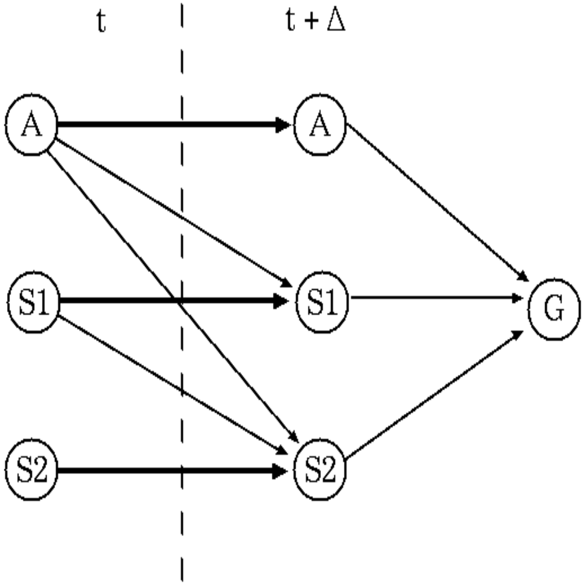
- First, every single gate is converted into DBN
- Then, the resulting DBNs are connected together in correspondance to the nodes they share.
- An adjustment to the CPT of a node is required when new arcs enter the node, due to the connection of two DBNs
  - add all the parents derived from DBN1 and DBN2 as columns in the new CPT;
  - in every entry of the table, set the probability of failure of the node using some interaction rules (Noisy-Or, MSP,…)

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: λ
- S1, S2 are the warm spare components
  - stand by → αλ   α is the dormancy factor (0<α<1)
  - working → λ



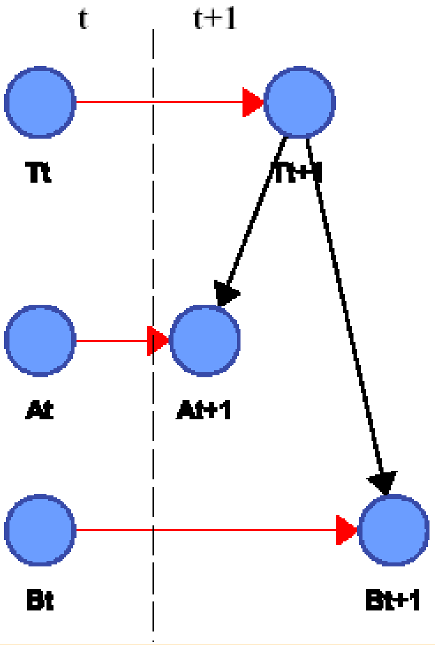


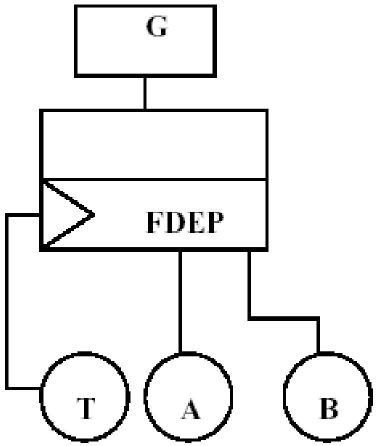
$$Pr\{A(t+\Delta)=1|A(t)=1\}=1$$
$$Pr\{A(t+\Delta)=1|A(t)=0\}=1-e^{-\lambda_A\Delta}$$

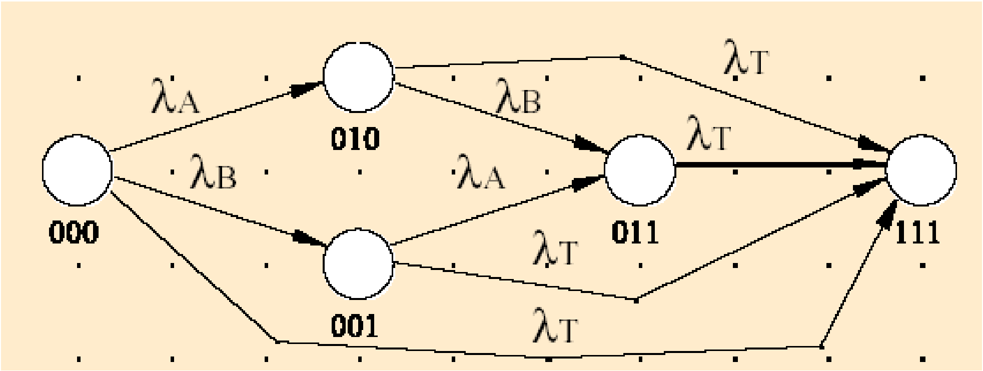
$$Pr\{S1(t+\Delta)=1|S1(t)=1\}=1$$
$$Pr\{S1(t+\Delta)=1|A(t)=0,S1(t)=0\}=1-e^{-\alpha\lambda_{S_1}\Delta}$$
$$Pr\{S1(t+\Delta)=1|A(t)=1,S1(t)=0\}=1-e^{-\lambda_{S_1}\Delta}$$

$$Pr\{S2(t+\Delta)=1|S2(t)=1\}=1$$
$$Pr\{S2(t+\Delta)=1|A(t)=0,S1(t)=0,S2(t)=0\}=1-e^{-\alpha\lambda_{S_2}\Delta}$$
$$Pr\{S2(t+\Delta)=1|A(t)=0,S1(t)=1,S2(t)=0\}=1-e^{-\alpha\lambda_{S_2}\Delta}$$
$$Pr\{S2(t+\Delta)=1|A(t)=1,S1(t)=0,S2(t)=0\}=1-e^{-\alpha\lambda_{S_2}\Delta}$$
$$Pr\{S2(t+\Delta)=1|A(t)=1,S1(t)=1,S2(t)=0\}=1-e^{-\lambda_{S_2}\Delta}$$

Functional Dependency gate







$$\Pr\{T(t+\Delta)=1|T(t)=1\}=1$$

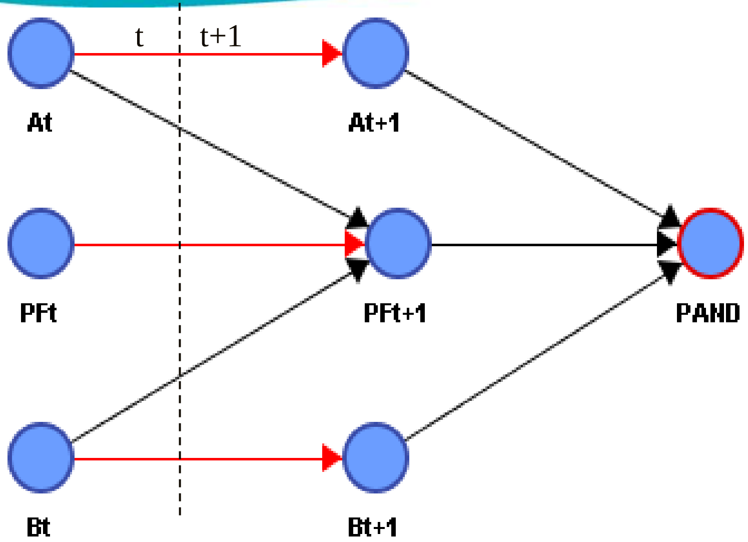
$$\Pr\{T(t+\Delta)=1|T(t)=0\}=1-e^{-\lambda_T \Delta t}$$

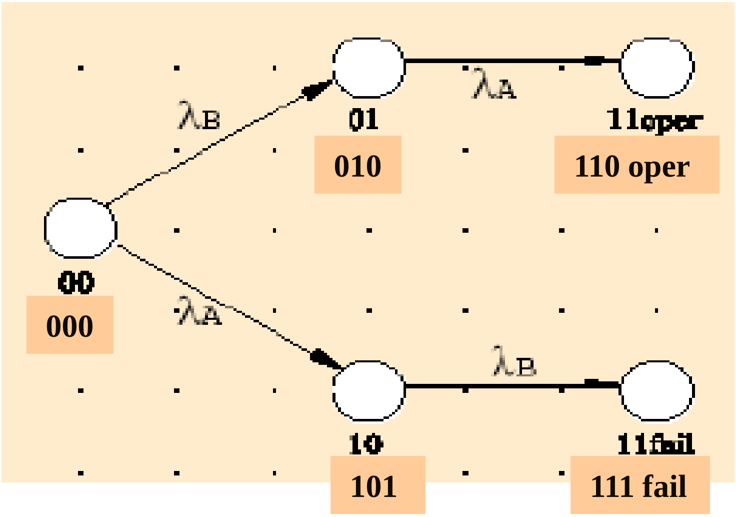
$$\Pr\{A(t+\Delta)=1|A(t)=1\}=1$$

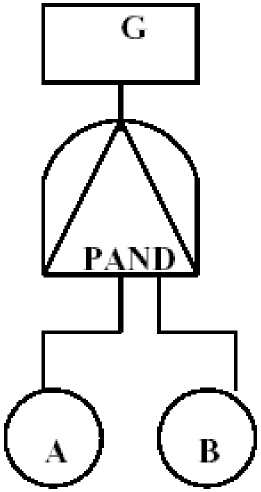
$$\Pr\{A(t+\Delta)=1|A(t)=0,T(t+\Delta)=0\}=1-e^{-\lambda_A \Delta t}$$

$$\Pr\{A(t+\Delta)=1|A(t)=0,T(t+\Delta)=1\}=p_{dep}(=1)$$

Priority AND gate







$\Pr\{A(t+\Delta)=1|A(t)=1\}=1$

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

$\Pr\{B(t+\Delta)=1|B(t)=1\}=1$

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

$\Pr\{PF(t+\Delta)=1|*,PF(t)=1\}=0$

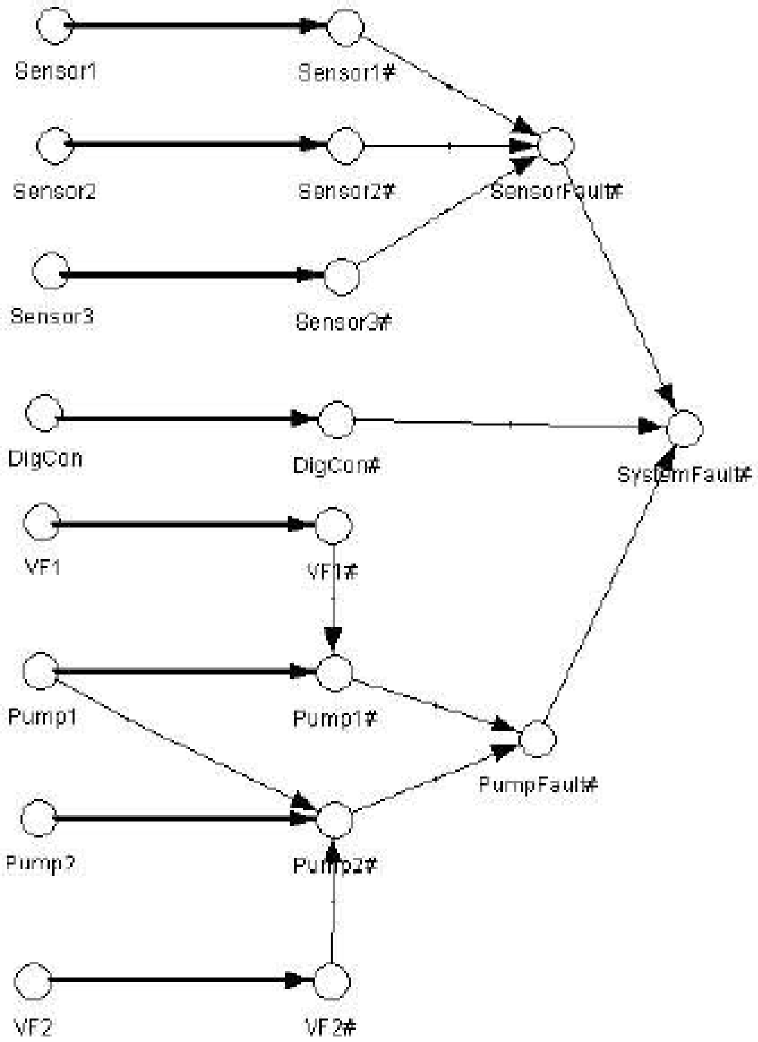
$\Pr\{PF(t+\Delta)=1|\ A(t)=0,\ B(t)=0,PF(t)=0\}=0$

$\Pr\{PF(t+\Delta)=1|\ A(t)=1,\ B(t)=0,PF(t)=0\}=1$

$\Pr\{PF(t+\Delta)=1|\ A(t)=0,\ B(t)=1,PF(t)=0\}=0$

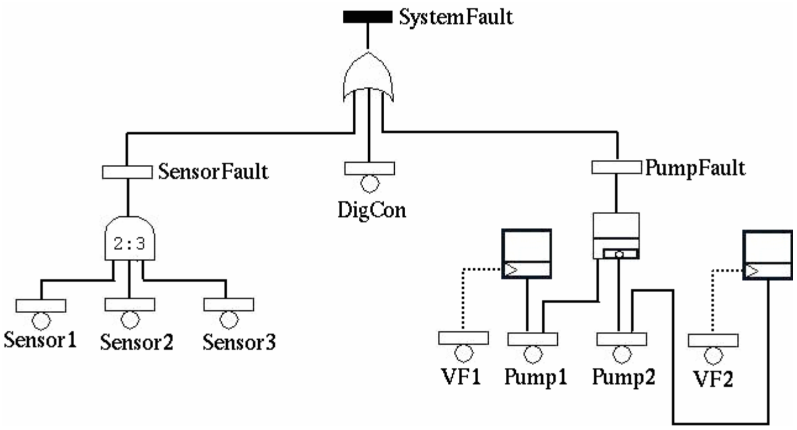
$\Pr\{PF(t+\Delta)=1|\ A(t)=1,\ B(t)=1,PF(t)=0\}=1$

HSS – DBN representation



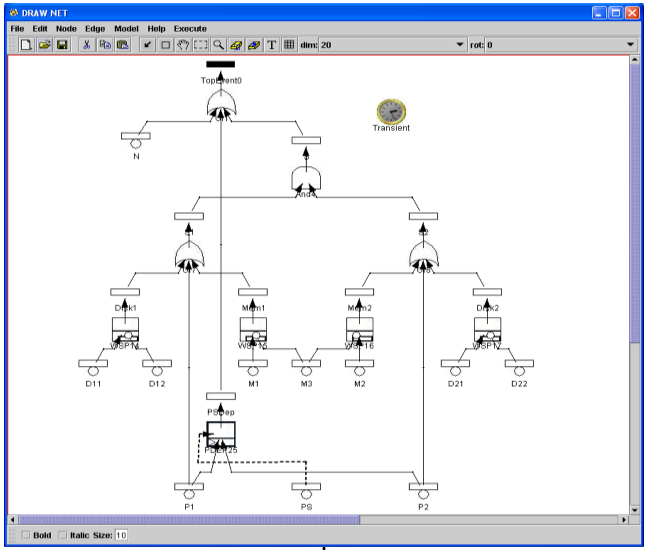
time (h)	RADYBAN	DRPFTPROC	GALILEO
200	0.0013651	0.0013651	0.0013651
400	0.0049082	0.0049083	0.0049083
600	0.0104139	0.0104139	0.0104140
800	0.0176824	0.0176826	0.0176826
1000	0.0265295	0.0265295	0.0265295

Comparison of results with different tools



Off-board tools suite

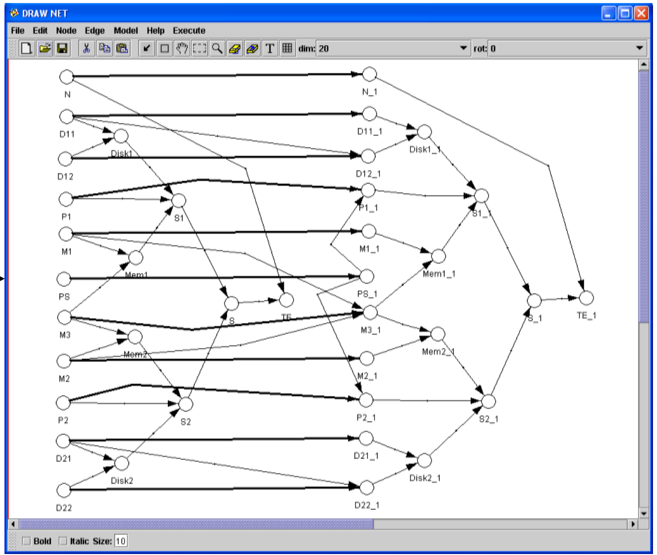
Off-board process



DFT.xml

DFT2DBN

DBN.xml



Enriched DBN.xml

DBN2JT

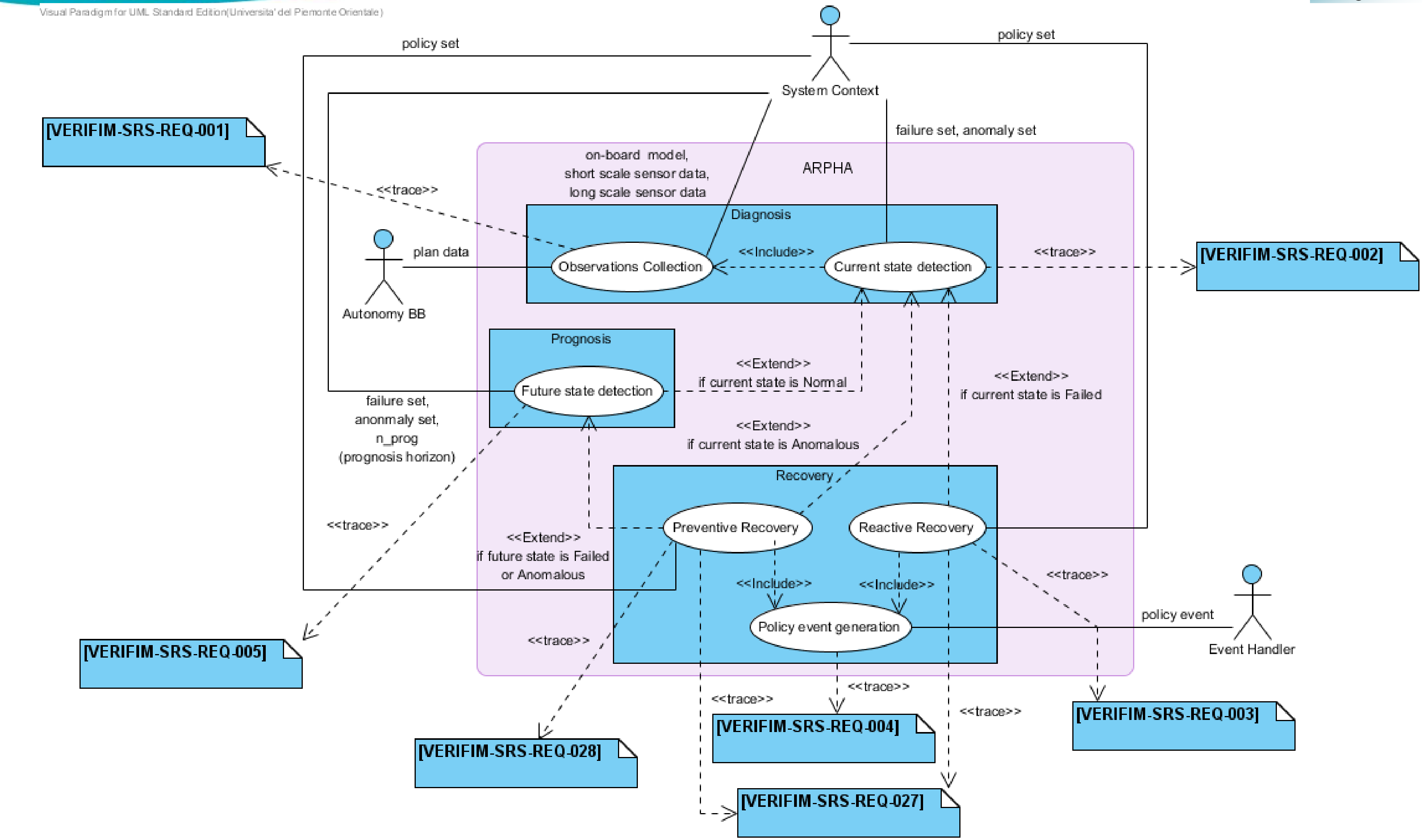
Junction Tree (JT.xml)

ARPHA

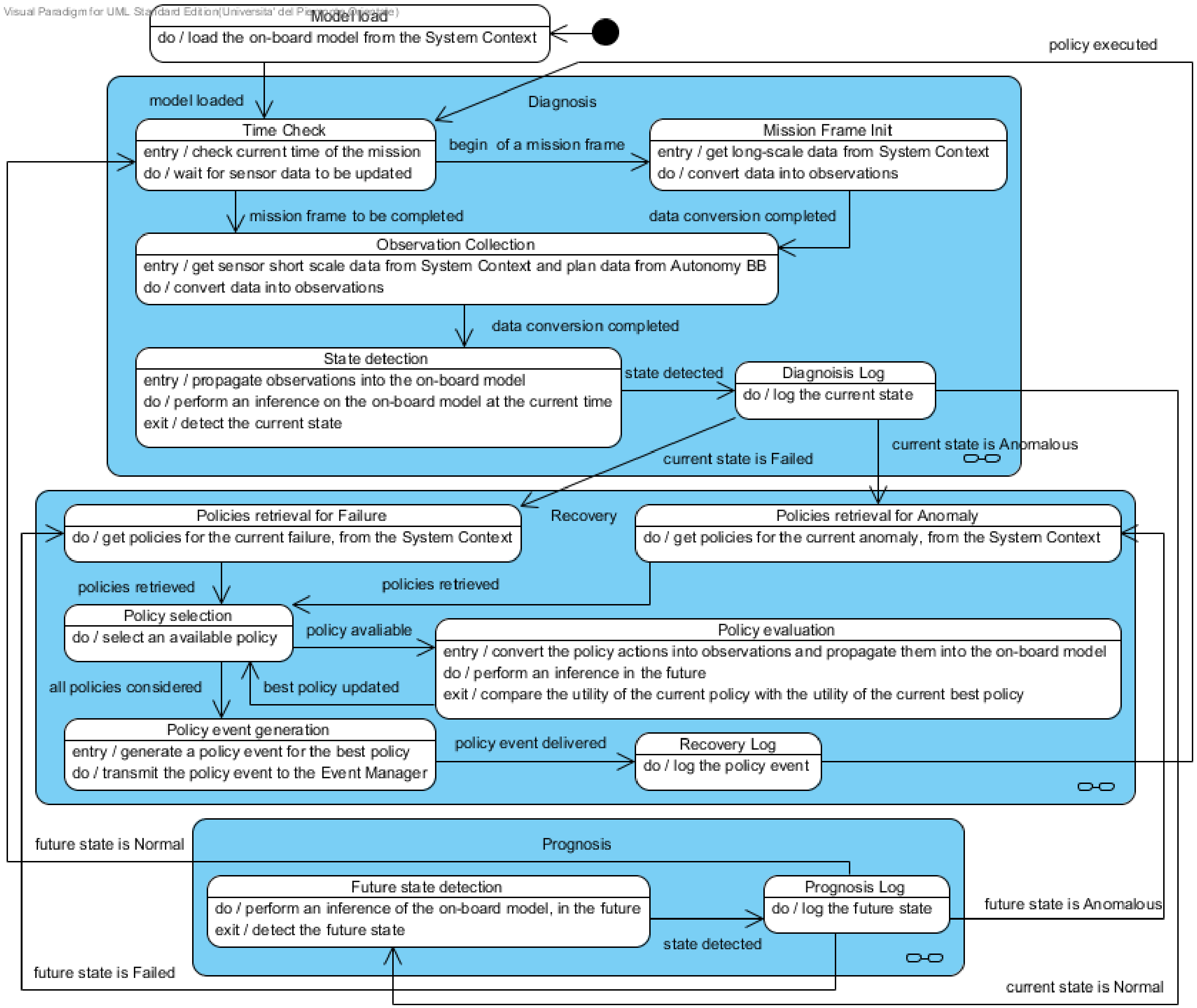
ARPHA

ON-BOARD ANOMALY RESOLUTION AND PROGNOSTIC HEALTH MANAGEMENT FOR AUTONOMY

On-board process Specification (1/2)



On-board process Specification (2/2)



Model load
do / load the on-board model from the System Context

model loaded

Diagnosis

Time Check
entry / check current time of the mission
do / wait for sensor data to be updated

begin of a mission frame

Mission Frame Init
entry / get long-scale data from System Context
do / convert data into observations

mission frame to be completed

data conversion completed

Observation Collection
entry / get sensor short scale data from System Context and plan data from Autonomy BB
do / convert data into observations

data conversion completed

State detection
entry / propagate observations into the on-board model
do / perform an inference on the on-board model at the current time
exit / detect the current state

state detected

Diagnoisis Log
do / log the current state

current state is Failed

current state is Anomalous

policy executed

Recovery

Policies retrieval for Failure
do / get policies for the current failure, from the System Context

Policies retrieval for Anomaly
do / get policies for the current anomaly, from the System Context

policies retrieved

policies retrieved

Policy selection
do / select an available policy

policy avaliable

Policy evaluation
entry / convert the policy actions into observations and propagate them into the on-board model
do / perform an inference in the future
exit / compare the utility of the current policy with the utility of the current best policy

all policies considered

best policy updated

Policy event generation
entry / generate a policy event for the best policy
do / transmit the policy event to the Event Manager

policy event delivered

Recovery Log
do / log the policy event

Prognosis

future state is Normal

Future state detection
do / perform an inference of the on-board model, in the future
exit / detect the future state

state detected

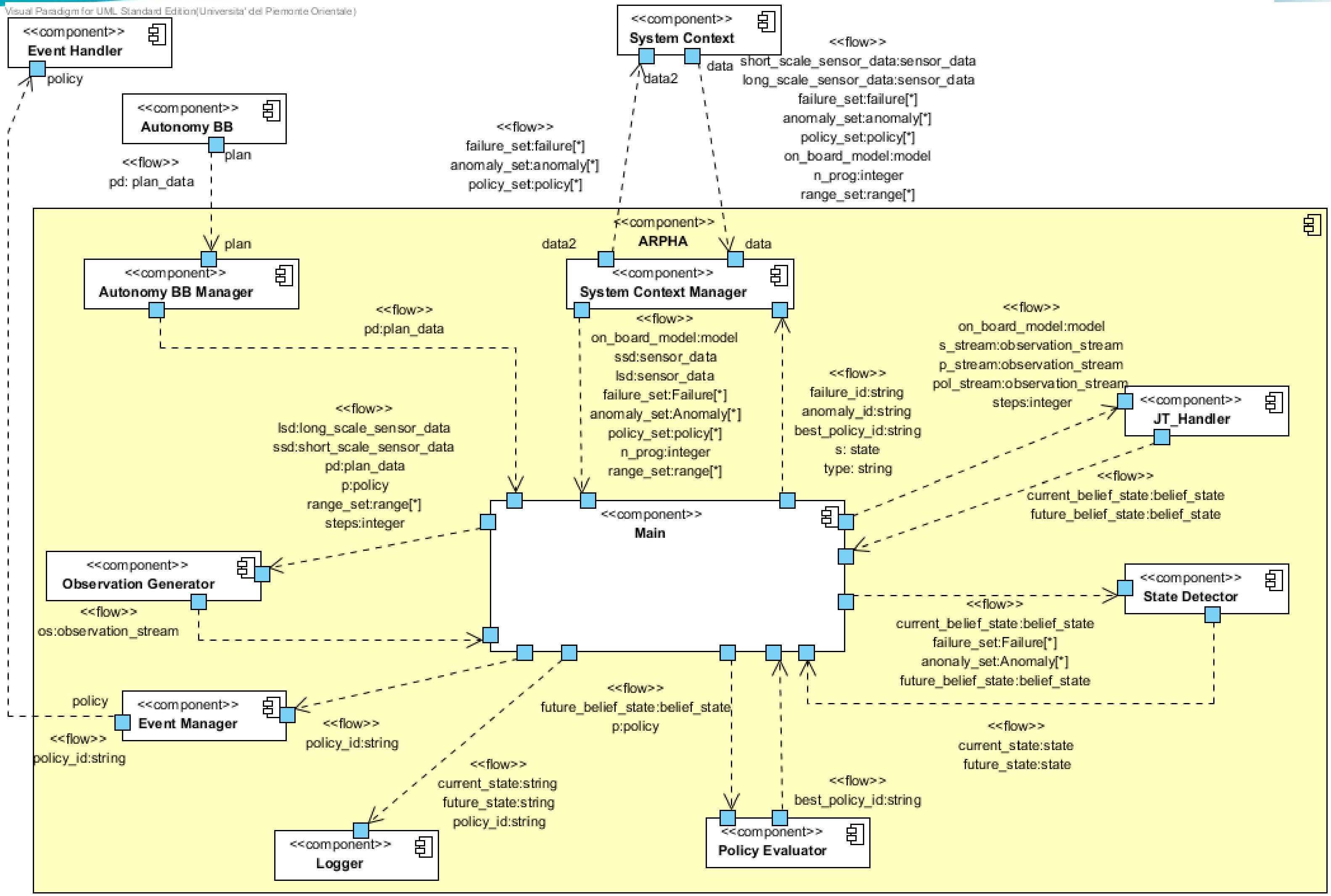
Prognosis Log
do / log the future state

future state is Anomalous

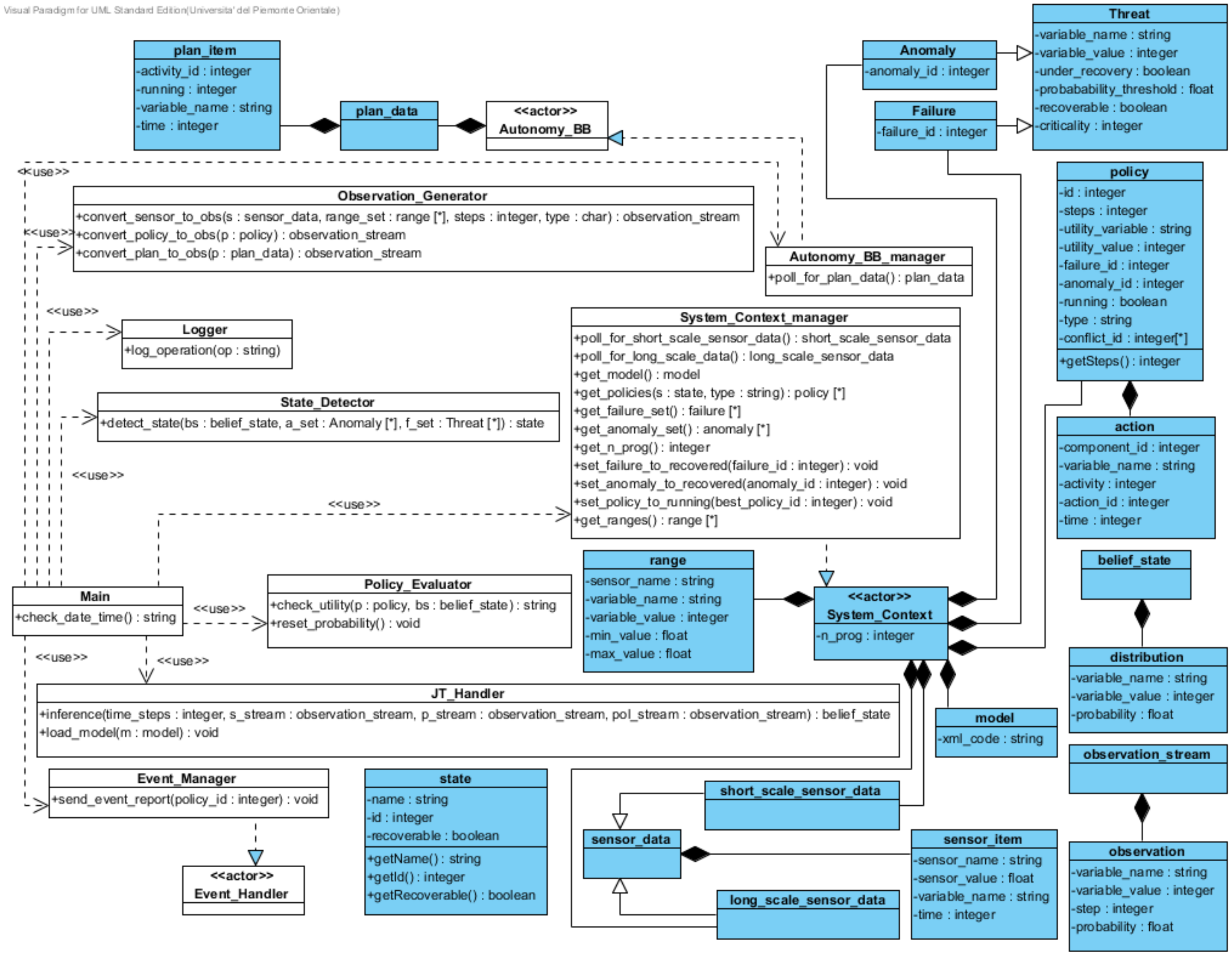
future state is Failed

current state is Normal

ARPHA Design: components



ARPHA Design: components and data structures



Visual Paradigm for UML Standard Edition(Universita' del Piemonte Orientale)

plan_item
-activity_id : integer
-running : integer
-variable_name : string
-time : integer

plan_data

Autonomy_BB

Anomaly
-anomaly_id : integer

Failure
-failure_id : integer

Threat
-variable_name : string
-variable_value : integer
-under_recovery : boolean
-probabability_threshold : float
-recoverable : boolean
-criticality : integer

Observation_Generator
+convert_sensor_to_obs(s : sensor_data, range_set : range [*], steps : integer, type : char) : observation_stream
+convert_policy_to_obs(p : policy) : observation_stream
+convert_plan_to_obs(p : plan_data) : observation_stream

Autonomy_BB_manager
+poll_for_plan_data() : plan_data

policy
-id : integer
-steps : integer
-utility_variable : string
-utility_value : integer
-failure_id : integer
-anomaly_id : integer
-running : boolean
-type : string
-conflict_id : integer[*]
+getSteps() : integer

Logger
+log_operation(op : string)

System_Context_manager
+poll_for_short_scale_sensor_data() : short_scale_sensor_data
+poll_for_long_scale_data() : long_scale_sensor_data
+get_model() : model
+get_policies(s : state, type : string) : policy [*]
+get_failure_set() : failure [*]
+get_anomaly_set() : anomaly [*]
+get_n_prog() : integer
+set_failure_to_recovered(failure_id : integer) : void
+set_anomaly_to_recovered(anomaly_id : integer) : void
+set_policy_to_running(best_policy_id : integer) : void
+get_ranges() : range [*]

State_Detector
+detect_state(bs : belief_state, a_set : Anomaly [*], f_set : Threat [*]) : state

action
-component_id : integer
-variable_name : string
-activity : integer
-action_id : integer
-time : integer

belief_state

range
-sensor_name : string
-variable_name : string
-variable_value : integer
-min_value : float
-max_value : float

System_Context
-n_prog : integer

Main
+check_date_time() : string

Policy_Evaluator
+check_utility(p : policy, bs : belief_state) : string
+reset_probability() : void

distribution
-variable_name : string
-variable_value : integer
-probability : float

JT_Handler
+inference(time_steps : integer, s_stream : observation_stream, p_stream : observation_stream, pol_stream : observation_stream) : belief_state
+load_model(m : model) : void

model
-xml_code : string

observation_stream

Event_Manager
+send_event_report(policy_id : integer) : void

state
-name : string
-id : integer
-recoverable : boolean
+getName() : string
+getId() : integer
+getRecoverable() : boolean

short_scale_sensor_data

sensor_data

sensor_item
-sensor_name : string
-sensor_value : float
-variable_name : string
-time : integer

long_scale_sensor_data

observation
-variable_name : string
-variable_value : integer
-step : integer
-probability : float

Event_Handler

ARPHA

IMPLEMENTATION

ARPHA software is composed by the following files :

- Main.c — represents Main object in Object Diagram
- Autonomy_BB_manager.c — represents Autonomy_BB_manager object in Object Diagram
- Event_manager.c — represents Event_manager object in Object Diagram
- Jt_handler.c — represents Jt_Handler object in Object Diagram
- Logger.c — represents Logger object in Object Diagram
- Observation_generator.c — represents Observation_generator object in Object Diagram
- Policy_evaluator.c — represents Policy_evaluator object in Object Diagram
- State_detector.c — represents State_detector object in Object Diagram
- System_context_manager.c — represents System_context_manager object in Object Diagram

ARPHA CODE ARCHITECTURE

- Model_definition.h contains data structures to represent the model in memory
- Recovery_definition.h contains data structures to do the Arpha process: Diagnosis-Prognosis-Recovery
- Global.h contains some global variables
- Config.h allows to set the number of slices to be considered in the prognosis;
  To configure sensors;
  To define anomalies;
  To define failures;
  To define plan and recovery actions;
  To define policies of recovery;
  To define plans.
- XML_JTS.h contains the XML of the model
- ROSEX_OUTPUT.h contains the sensor and plan data (file used n Off-mode by Arpha)
- Definition.h contains the definition of all the constants used by Arpha to be set in order to optimize memory utilization of ARPHA

Note that at all file c, exception to Main.c, corresponding a header file where are declared the prototipies of functions implemented in the c file and used by another modules.

ARPHA mode

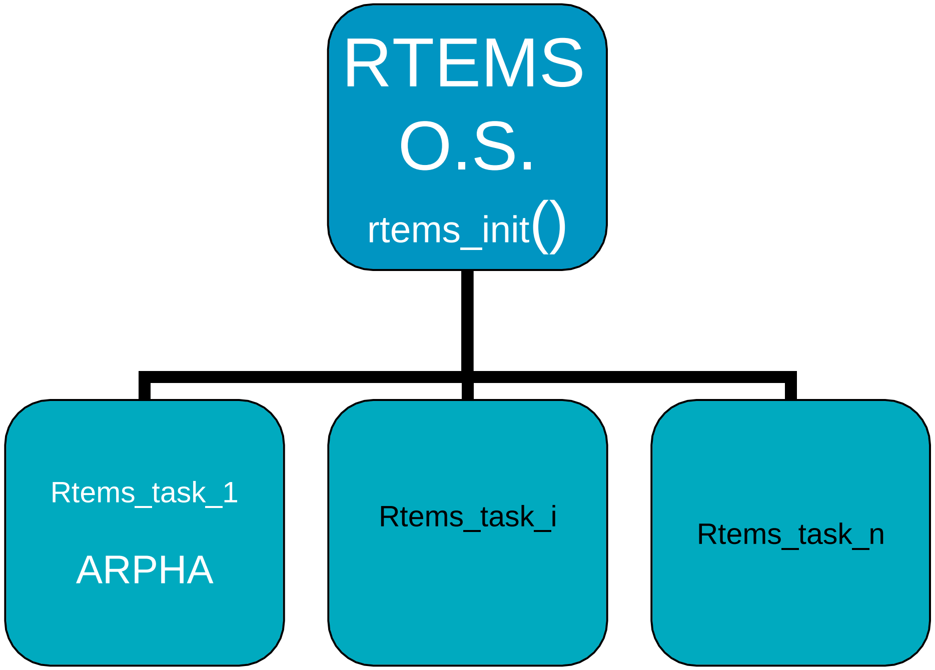
- **ARPHA can be executed in two ways:**
  - On-line Mode
    - used to run ARPHA in the evaluation platform
  - Off-line Mode
    - used to perform validation w.r.t. TS and RB
    - the values of sensors and the current plan and action must be written in ROSEX_OUTPUT.h,

ARPHA RELEASES

ARPHA source software is zip-file with the following content:

- ***Eclipse project***
- ***Visual Studio C++ project***
- A directory that containing Source code to be compiled with ***gcc*** (Version : 4.3.4)
- A directory that containing Source code to be compiled with ***sparc-rtems-gcc***

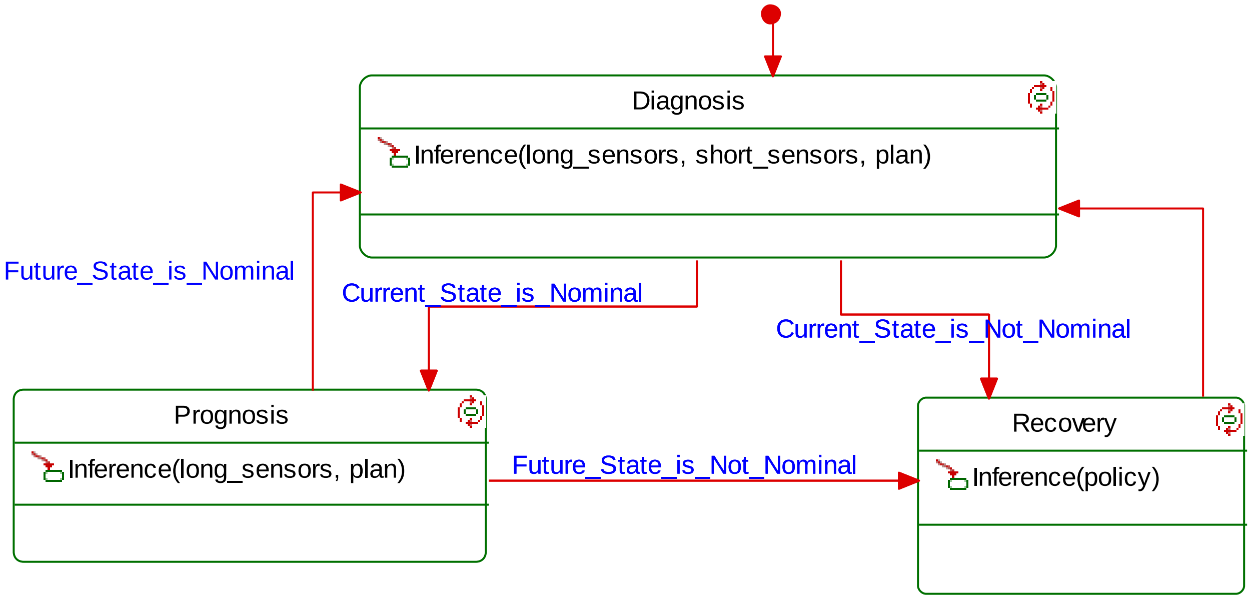
ARPHA in RTEMS



RTEMS pseudo code:

```
Rtems_task_init()
{
        New_RTEMS_Task Arpha  =  rtems_task_create();
        rtems_task_start(Arpha);
}
```

ARPHA Cycle


Diagnosis
Inference(long_sensors, short_sensors, plan)
Future_State_is_Nominal
Current_State_is_Nominal
Current_State_is_Not_Nominal
Prognosis
Inference(long_sensors, plan)
Future_State_is_Not_Nominal
Recovery
Inference(policy)

Evaluation of the approach

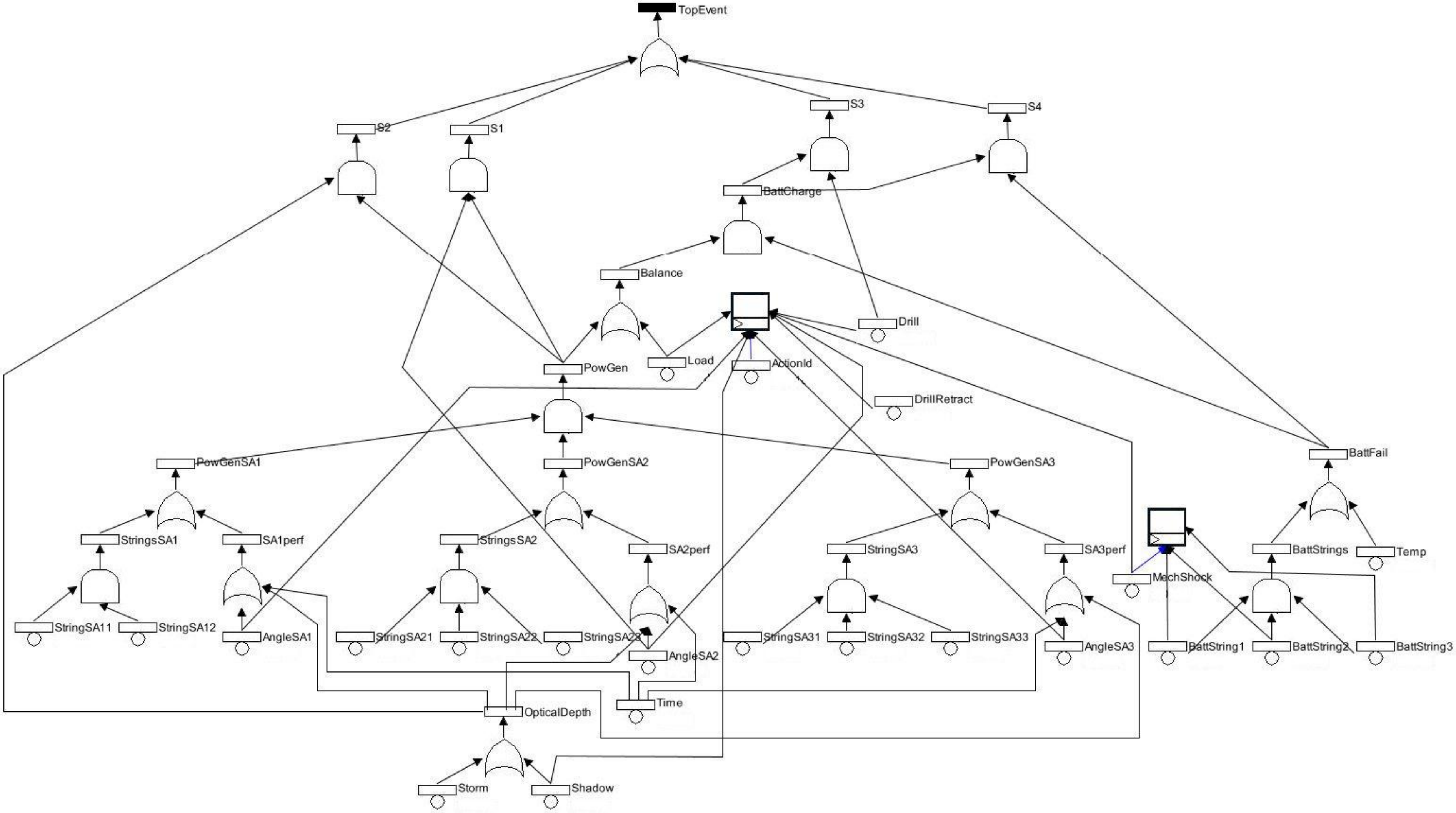
Case Study Description

Case study: planetary rover

The case study deals with the power supply system of the rover, with a particular attention to the following aspects and their combinations:

- **the power supply by the solar arrays: 3 solar arrays, namely SA1, SA2, SA3. Each solar array can generate power if two conditions hold:**
  - at least one string is not failed;
  - the combination of sun aspect angle, optical depth, and local time (day or night) is suitable.
- **the load: The amount of load depends on the current action performed by the rover**
- **the power supply by the battery composed by three redundant strings:**
  - The charge of the battery may be steady, decreasing or increasing according to the current levels of load and generation by the solar arrays
  - The charge of the battery may be compromized by the damage of the battery occurring in two situations:
  1. all the strings are failed,
  2. or the temperature of the battery is low.
- **Scenarios:**
  - Slope of terrain (S1): the presence of a terrain slope increases sun aspect angle by causing lower power generation of solar array
  - Presence of dusty (S2) : the presence of dust increases optical depth and reduces power generated by solar arrays.
  - Problem during drilling (S3) : we simulate an unexpected high request of energy by drill.
  - Damage to battery system (S4): we simulate a damage to battery that reduces battery charge level.

Dynamic Fault Tree model



Scenarios and policies (1/2)

We are interested in 4 failure or anomaly scenarios. Each scenario can be recovered by specific policies:

Scenario 1: slope of terrain.

Recovery policies:

P1) suspension of the plan in order to reduce the load

P2) change of inclination of SA2 and SA3 in order to try to improve the sun aspect angle and consequently the power generation (the tilting system can not act on SA1)

Scenario 2: presence of dusty.

Recovery policies:

P3) movement of the rover into another position in order to try to avoid a shadowed area and improve the power generation as a consequence

P4) modication of the inclination of SA2 and SA3, retraction of the drill, and suspension of the plan

Scenarios and policies (2/2)

Scenario 3: problem during drilling.

Recovery policies:

P4) as in scenario 2

P5) retraction of the drill, suspension of the plan

Scenario 4: damage to battery system.

Recovery policies:

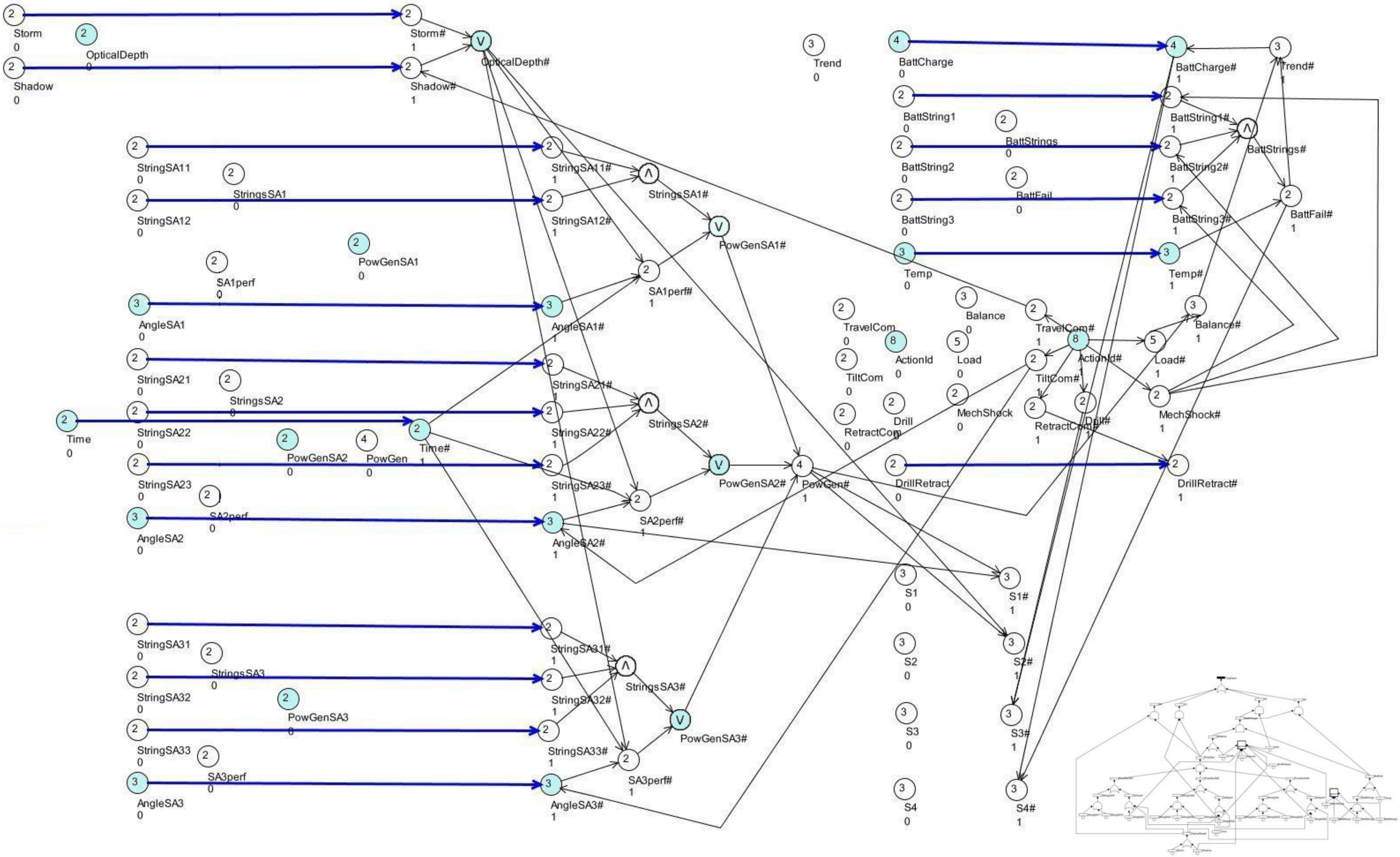
P4) as in scenario 2 and scenario 3

From DFT to DBN

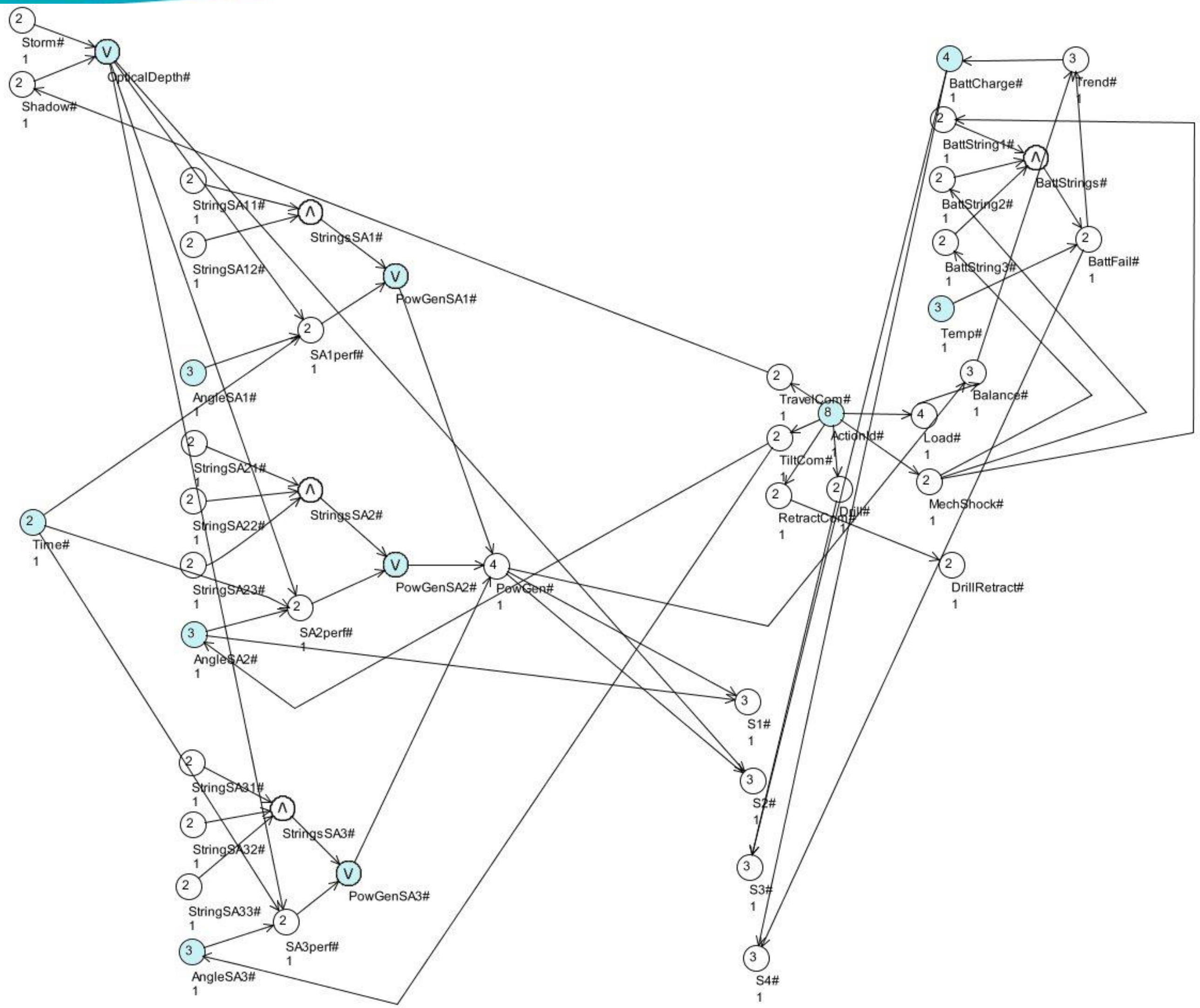
- Conversion of the DFT into DBN
  - DFT events become DBN variables
  - DFT gates determine the CPTs of the DBN variables

- Enrichment of the resulting DBN
  - Increase of the size of several variables
    - Multi-state components, conditions, events, levels, ...
  - Update of conditional probability tables (CPT)
    - Non binary variables
    - Non Boolean relations among variables
  - Addition of support variables in order to reduce the number of CPT entries
    - «divorcing» technique

Dynamic Bayesian Network

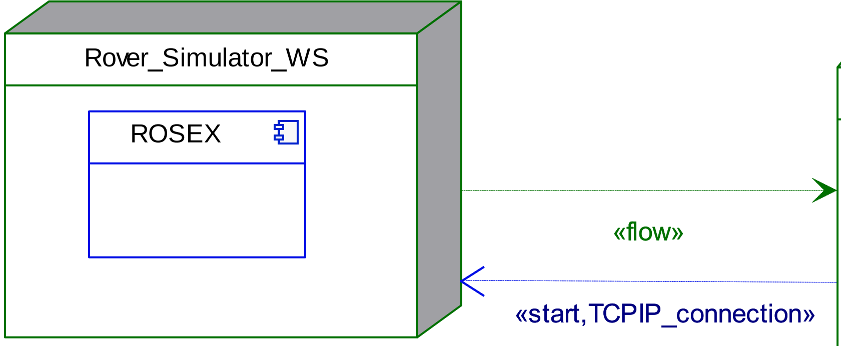


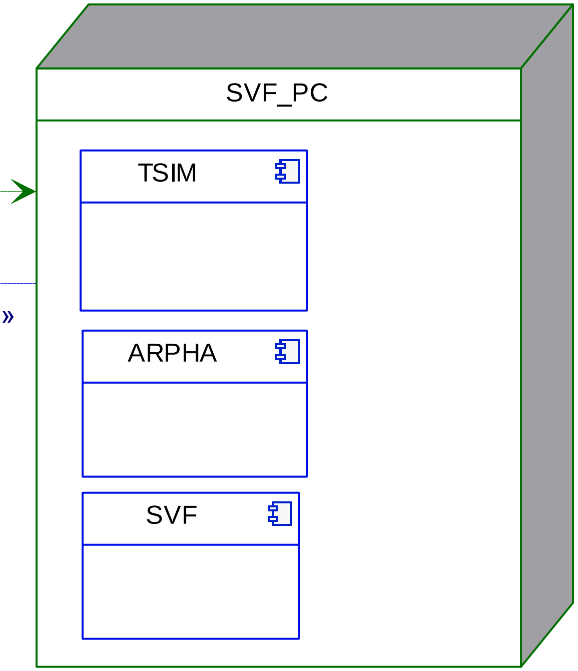
Dynamic Bayesian Network (t+Δ slice only)

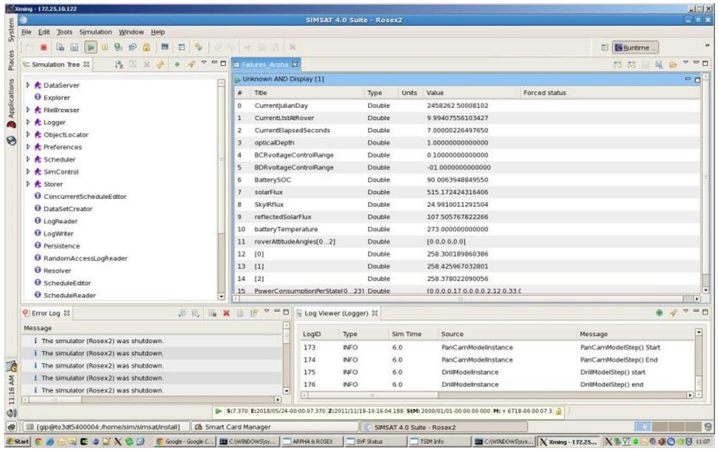


Evaluation results

Evaluation Platform







- DataServer
- Explorer
- FileBrowser
- Logger
- ObjectLocator
- Preferences
- Scheduler
- SimControl
- Storer
- ConcurrentScheduleEditor
- DataSetCreator
- LogReader
- LogWriter
- Persistence
- RandomAccessLogReader
- Resolver
- ScheduleEditor
- ScheduleReader

Unknown AND Display (1)

#	Title	Type	Units	Value	Forced status
0	CurrentJulianDay	Double		2458262.50008102	
1	CurrentLtstAtRover	Double		9.99407556103427	
2	CurrentElapsedSeconds	Double		7.00000226497650	
3	opticalDepth	Double		1.00000000000000	
4	BCRvoltageControlRange	Double		0.10000000000000	
5	BDRvoltageControlRange	Double		-01.0000000000000	
6	BatterySOC	Double		90.0063948849550	
7	solarFlux	Double		515.172424316406	
8	SkyIRflux	Double		24.9910011291504	
9	reflectedSolarFlux	Double		107.505767822266	
10	batteryTemperature	Double		273.000000000000	
11	roverAttitudeAngles[0...2]	Double		[0.0,0.0,0.0]	
12	[0]	Double		258.300189860386	
13	[1]	Double		258.425967032801	
14	[2]	Double		258.378022090056	
15	PowerConsumptionPerState[0...23]	Double		[0.0,0.0,17.0,0.0,0.0,2.12,0.33,(	

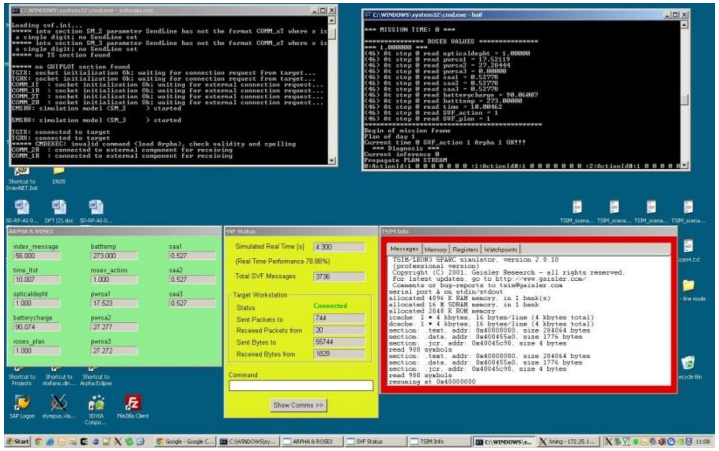
Error Log

Message
The simulator (Rosex2) was shutdown.
The simulator (Rosex2) was shutdown.
The simulator (Rosex2) was shutdown.
The simulator (Rosex2) was shutdown.
The simulator (Rosex2) was shutdown.

Log Viewer (Logger)

LogID	Type	Sim Time	Source	Message
173	INFO	6.0	PanCamModelInstance	PanCamModelStep() Start
174	INFO	6.0	PanCamModelInstance	PanCamModelStep() End
175	INFO	6.0	DrillModelInstance	DrillModelStep() start
176	INFO	6.0	DrillModelInstance	DrillModelStep() end

```
S:7.370 E:2018/05/24-00:00:07.370 Z:2011/11/18-10:16:04.188 StM: 2000/01/01-00:00:00.000 M: + 6718-00:00:07.3
```



```
Loading svf.ini...
***** into section SM_2 parameter SendLine has not the format COMM_xT where x is
 a single digit; no SendLine set
***** into section SM_3 parameter SendLine has not the format COMM_xT where x is
 a single digit; no SendLine set
***** no TS section found

***** no GUIPLOT section found
TGTX: socket initialization Ok; waiting for connection request from target...
TGRX: socket initialization Ok; waiting for connection request from target...
COMM_1T  : socket initialization Ok; waiting for external connection request...
COMM_1R  : socket initialization Ok; waiting for external connection request...
COMM_2T  : socket initialization Ok; waiting for external connection request...
COMM_2R  : socket initialization Ok; waiting for external connection request...
SMSRV: simulation model <SM_2      > started

SMSRV: simulation model <SM_3      > started

TGTX: connected to target
TGRX: connected to target
***** CMDEXEC: invalid command <load Arpha>, check validity and spelling
COMM_2R  : connected to external component for receiving
COMM_1R  : connected to external component for receiving
```

```
*** MISSION TIME: 0 ***

*************** ROSEX VALUES ***************
*** 1.000000 ***
<46> At step 0 read opticaldepht = 1.000000
<46> At step 0 read pwrsa1 = 17.52119
<46> At step 0 read pwrsa2 = 27.20444
<46> At step 0 read pwrsa3 = 0.00000
<46> At step 0 read saa1 = 0.52778
<46> At step 0 read saa2 = 0.52778
<46> At step 0 read saa3 = 0.52778
<46> At step 0 read batterycharge = 90.06007
<46> At step 0 read batttemp = 273.00000
<46> At step 0 read time = 10.00462
<46> At step 0 read SVF_action = 1
<46> At step 0 read SVF_plan = 1
*******************************************
Begin of mission frame
Plan of day 1
Current time 0 SVF_action 1 Arpha 1 OK!!!
  *** Diagnosis ***
Current inference 0
Propagate PLAN STREAM
0:ActionId:1 0 0 0 0 0 0 0 0 :1:ActionId#:1 0 0 0 0 0 0 0 0 :2:ActionId#:1 0 0 0 0 0
```

ARPHA & ROSEX

index_message 56.000	batttemp 273.000	saa1 0.527
time_ltst 10.007	rosex_action 1.000	saa2 0.527
opticaldepht 1.000	pwrsa1 17.523	saa3 0.527
batterycharge 90.074	pwrsa2 27.277	
rosex_plan 1.000	pwrsa3 27.272	

SVF Status

Simulated Real Time [s] 4.300

(Real Time Performance 78.88%)

Total SVF Messages 3736

Target Workstation

Status	Connected
Sent Packets to	744
Received Packets from	20
Sent Bytes to	65744
Received Bytes from	1829

Command

Show Comms >>

TSIM Info

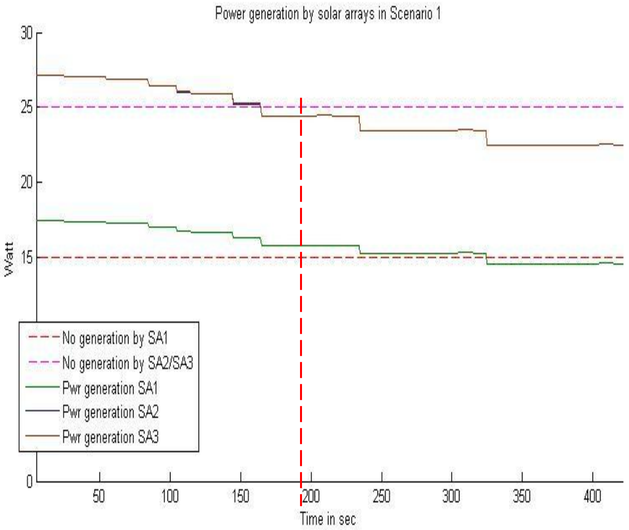
Messages | Memory | Registers | Watchpoints

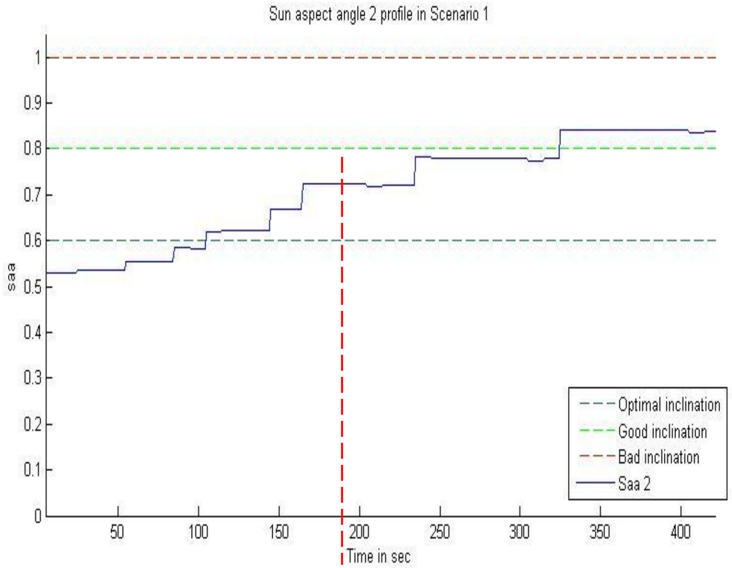
```
 TSIM/LEON3 SPARC simulator, version 2.0.10
 (professional version)
 Copyright (C) 2001, Gaisler Research - all rights reserved.
 For latest updates, go to http://www.gaisler.com/
 Comments or bug-reports to tsim@gaisler.com
serial port A on stdin/stdout
allocated 4096 K RAM memory, in 1 bank(s)
allocated 16 M SDRAM memory, in 1 bank
allocated 2048 K ROM memory
icache: 1 * 4 kbytes, 16 bytes/line (4 kbytes total)
dcache: 1 * 4 kbytes, 16 bytes/line (4 kbytes total)
section: .text, addr: 0x40000000, size 284064 bytes
section: .data, addr: 0x400455a0, size 1776 bytes
section: .jcr, addr: 0x40045c90, size 4 bytes
read 980 symbols
section: .text, addr: 0x40000000, size 284064 bytes
section: .data, addr: 0x400455a0, size 1776 bytes
section: .jcr, addr: 0x40045c90, size 4 bytes
read 980 symbols
resuming at 0x40000000
```

Scenario 1: Slope of terrain

```
*** MISSION TIME: 3 ***
 *************** ROSEX VALUES ***************
(187) At step 3 read pwrsa1 = 15.71248
(187) At step 3 read pwrsa2 = 24.40224
(187) At step 3 read pwrsa3 = 24.41719
(187) At step 3 read saa2 = 0.72340
**********************************************
*** Diagnosis ***
NO FAILURE Pr{S1#=2} = 0.00000000 (0.99000000)
NO FAILURE Pr{S2#=2} = 0.00000000 (0.59000000)
NO FAILURE Pr{S3#=2} = 0.00000000 (0.99000000)
NO FAILURE Pr{S4#=2} = 0.00000000 (0.99000000)
(Criticality level 2) Anomaly 1 save because more important
[Pr{S1#=1} = 1.00000000] >= 0.99000000
Anomaly 2 excluded because under recovery or minor criticaly (no check)
NO ANOMALY Pr{S3#=1} = 0.00000000 (0.99000000)
NO ANOMALY Pr{S4#=1} = 0.00000000 (0.99000000)
      STATE SYSTEM "A" (1)
Elapsed Time for diagnosis: 8.690000 sec
  ## Preventive Recovery ##
Policy 7 discard (policy 6 has utility 0.47361844)
Best policy for Preventive Recovery is the 6
BEST POLICY for the anomaly 1 is:      6
Anomaly 1 under recovery
Policy 6 running
Elapsed Time Prognosis and or Recovery: 175.370000 sec
```

the presence of a terrain slope increases sun aspect angle by causing lower power generation of solar array.





Scenario 2: Presence of Dust

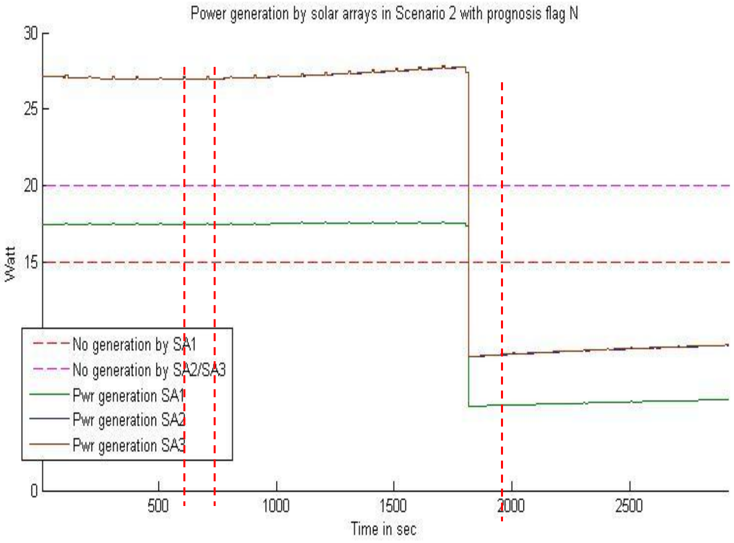
Arpha output at **mission time 9 (663) and 10 (728)**:

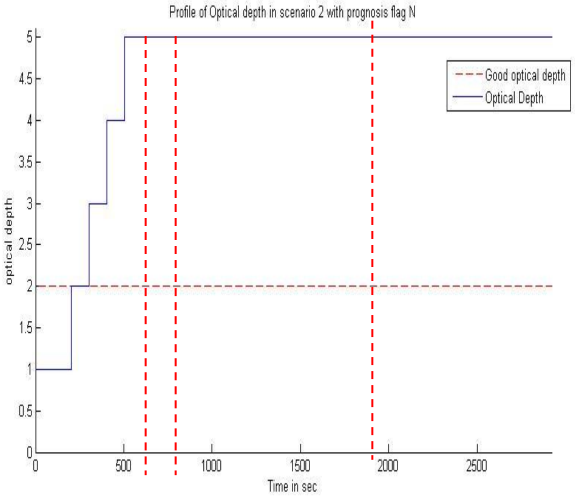
```
*** Diagnosis *** STATE SYSTEM "N“
 ## Prognosis ## FUTURE STATE SYSTEM "F" (2)
Future System state anomalous/failied but prognosis flag set to 'N'
Elapsed Time Prognosis and or Recovery: 43.500000 sec

*** MISSION TIME: 27 ***

*************** ROSEX VALUES ***************
(1835) At step 27 read opticaldepht = 5.00000
(1835) At step 27 read pwrsa1 = 5.53303
(1835) At step 27 read pwrsa2 = 8.77638
(1835) At step 27 read pwrsa3 = 8.79704
**********************************************
*** Diagnosis ***

NO FAILURE Pr{S1#=2} = 0.00000000 (0.99000000)
(Criticality level 3) Failure 2 save
[Pr{S2#=2} = 1.00000000] >= 0.59000000
NO FAILURE Pr{S3#=2} = 0.00000000 (0.99000000)
NO FAILURE Pr{S4#=2} = 0.00000000 (0.99000000)
Anomaly 1 excluded because under recovery or minor criticaly (no check)
Anomaly 2 excluded because under recovery or minor criticaly (no check)
NO ANOMALY Pr{S3#=1} = 0.00000000 (0.99000000)
NO ANOMALY Pr{S4#=1} = 0.00000000 (0.99000000)
      STATE SYSTEM "F" (2)
Elapsed Time for diagnosis: 8.630000 sec
  ## Reactive Recovery ##
Policy 3 save as best (0.09183467)
Policy 4:
Utility Function= 0.8773
Policy 4 save as best (previous 0.09183467)
Best policy for Reactive Recovery is the 4
BEST POLICY for the failure 2 is:       4
Failure 2 under recovery
Policy 4 running
Elapsed Time Prognosis and or Recovery: 174.320000 sec
```



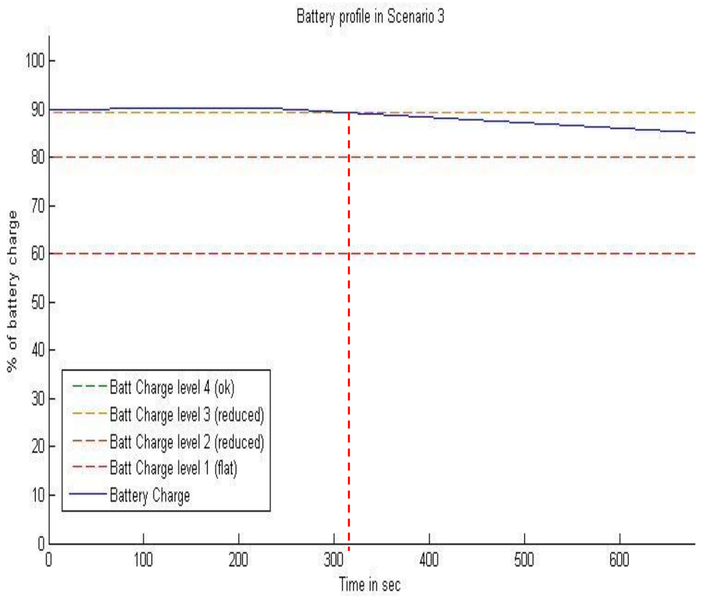


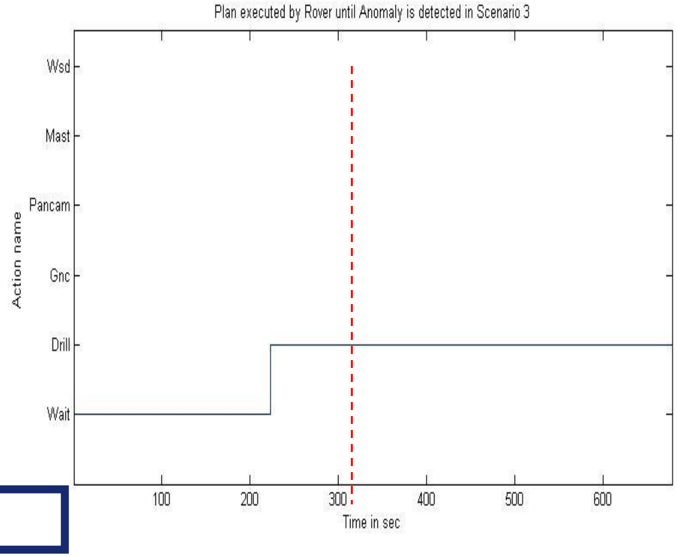
presence of dust increases optical depth and reduces power generated by solar arrays.

Scenario 3: Drill

```
*** MISSION TIME: 5 ***

*************** ROSEX VALUES ***************
(338) At step 5 read batterycharge = 89.04301
(338) At step 5 read SVF_action = 2
**********************************************
*** Diagnosis ***
NO FAILURE Pr{S1#=2} = 0.00000000 (0.99000000)
NO FAILURE Pr{S2#=2} = 0.00000000 (0.59000000)
NO FAILURE Pr{S3#=2} = 0.00000000 (0.99000000)
NO FAILURE Pr{S4#=2} = 0.00000000 (0.99000000)
NO ANOMALY Pr{S1#=1} = 0.00000000 (0.99000000)
NO ANOMALY Pr{S2#=1} = 0.00000000 (0.59000000)
(Criticality level 3) Anomaly 3 save because more important
[Pr{S3#=1} = 1.00000000] >= 0.99000000
NO ANOMALY Pr{S4#=1} = 0.21929918 (0.99000000)
       STATE SYSTEM "A" (3)
Elapsed Time for diagnosis: 8.630000 sec
  ## Preventive Recovery ##
Policy 9 save as best (0.88224292)
Policy 10 save as best (previous 0.88224292)
Best policy for Preventive Recovery is the 10
BEST POLICY for the anomaly 3 is:      10
Elapsed Time Prognosis and or Recovery: 173.930000 sec
```



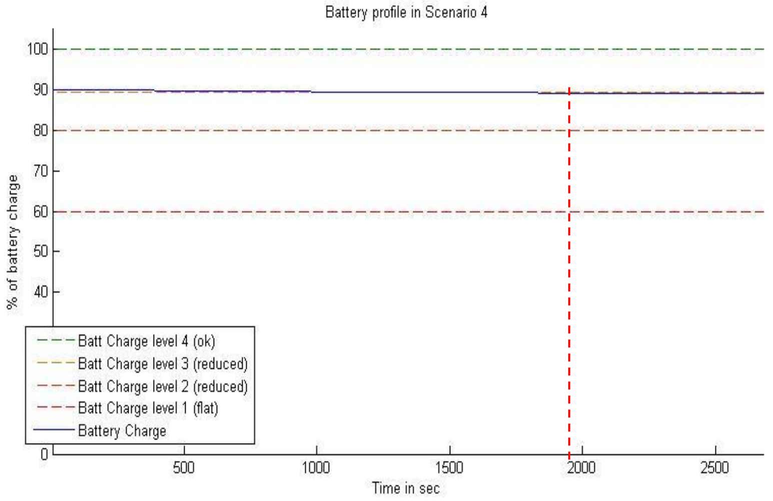


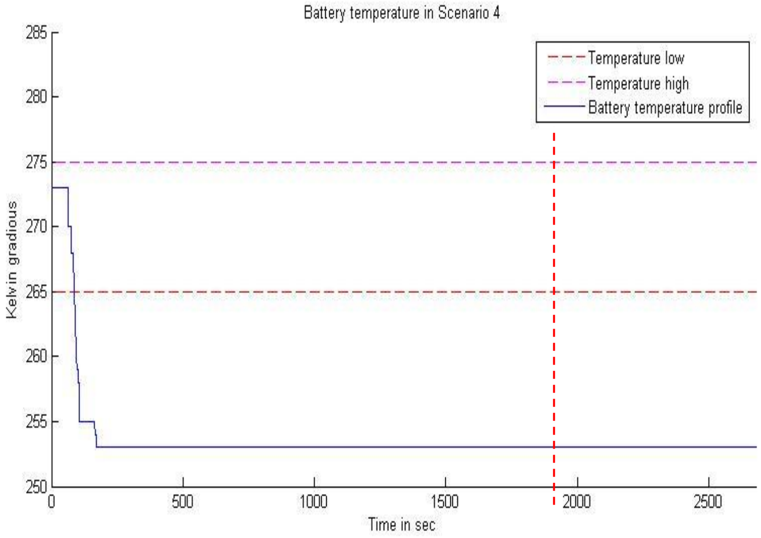
unexpected high request of energy by drill

Scenario 4: Damage to battery

```
*** MISSION TIME: 27 ***

*************** ROSEX VALUES ***************
(1816) At step 27 read batterycharge = 89.34396
(1816) At step 27 read batttemp = 253.00000
**********************************************
*** Diagnosis ***
 NO FAILURE Pr{S1#=2} = 0.00000000 (0.99000000)
NO FAILURE Pr{S2#=2} = 0.00000000 (0.59000000)
NO FAILURE Pr{S3#=2} = 0.00000000 (0.99000000)
NO FAILURE Pr{S4#=2} = 0.00000000 (0.99000000)
NO ANOMALY Pr{S1#=1} = 0.00000000 (0.99000000)
NO ANOMALY Pr{S2#=1} = 0.00000000 (0.59000000)
NO ANOMALY Pr{S3#=1} = 0.00000000 (0.99000000)
(Criticality level 4) Anomaly 4 save because more importa
[Pr{S4#=1} = 1.00000000] >= 0.99000000
       STATE SYSTEM "A" (4)
Elapsed Time for diagnosis: 8.630000 sec
  ## Preventive Recovery ##
Policy 9 save as best (0.88270821)
Best policy for Preventive Recovery is the 9
BEST POLICY for the anomaly 4 is:      9
Anomaly 4 under recovery
Policy 9 running
Elapsed Time Prognosis and or Recovery: 87.080000 sec
```





a damage to battery reduces battery charge level

Characterization of the approach

Reliability

- ARPHA prototype has satisfied requirements of SSS and SRS .
- ARPHA is able to perform diagnosis of the system and eventually to detect in the current state a failure or an anomaly and to select the policy recovery.
- ARPHA is able to perform prognosis of the system and eventually to foresee in the future state a failure or an anomaly and to select a preventive recovery.

Adequacy (1/2)

- ARPHA is able to verify the failure impact on the future state of the system.
- Environmental aspect of space mission can be modelized in the DBN used by ARPHA to perform inference.
- It is possible to take into account the failure causes by inserting them in utility functions used to select recovery.
- It receives the current action under execution in the autonomy building block, and so it is possible to evaluate failure impact on the currently executing plan.
- Critical on-board system requirements (as robustness to stack Overflow, not use of unbounded depth recursive functions, not use of dynamic memory allocation) are respected.
- Parameters used by ARPHA can be changed via TC from ground.

Adequacy (2/2)

- ARPHA is capable of on-board decision making to appropriately react to system faults and failure.
- ARPHA increases autonomy of spacecraft, by introducing the prognosis on-board with two purposes:
  - **detecting the future belief state,**
  - **and evaluating the future effects of recovery policies in order to select the most suitable to deal with anomalies or failures.**
- Diagnosis, prognosis and recovery are not completely based on the probability parameters present in the model, but they take into account information coming from the system and environment monitoring. This guarantees an adequate resolution of the encountered anomalies.
- ARPHA is complementary to a goal-oriented autonomy Building Block. ARPHA does not perform planning/re-planning, but it takes advantage of knowledge about the current plan under execution. It could suggest re-planning in case a recovery action has to be performed. ARPHA can force the system to go into a safe state, from which a new planning activity has to start.

Effectiveness

- ARPHA is able to take into account of component fault rate.
- Correctness of diagnosis, prognosis and recovery depends on CPT.
- ARPHA is more effective by observing phenomena that have a modification rate comparable with time spent for ARPHA cycle.

Availability (1/2)

- **Frequency of observations collection to perform diagnosis of current state and prognosis of future state depends on duration of an ARPHA cycle. It is possible that sensors values are updated by system with an higher frequency respect to ARPHA observations collection.**
- **Software is schedulable: A schedulability analysis approach has been proposed to understand the deadlines to be respected, in order to ensure stability and convergence of the control loop formed by the ARPHA and the system it is observing and controlling.**

Availability (2/2)

- ARPHA worst case execution time (C) is composed by Diagnosis/Prognosis and/or Recovery time. Duration of these processes depends on model complexity:
  - ARPHA Diagnosis time depends on model complexity,
  - ARPHA Prognosis time depends on n_prog and model complexity,
  - ARPHA Recovery time depends on time steps of recovery and model complexity),
- ARPHA deadline depends on the system under control.
- ARPHA deadline can be computed by considering the maximum length of the ARPHA cycle for the observed/controlled system.

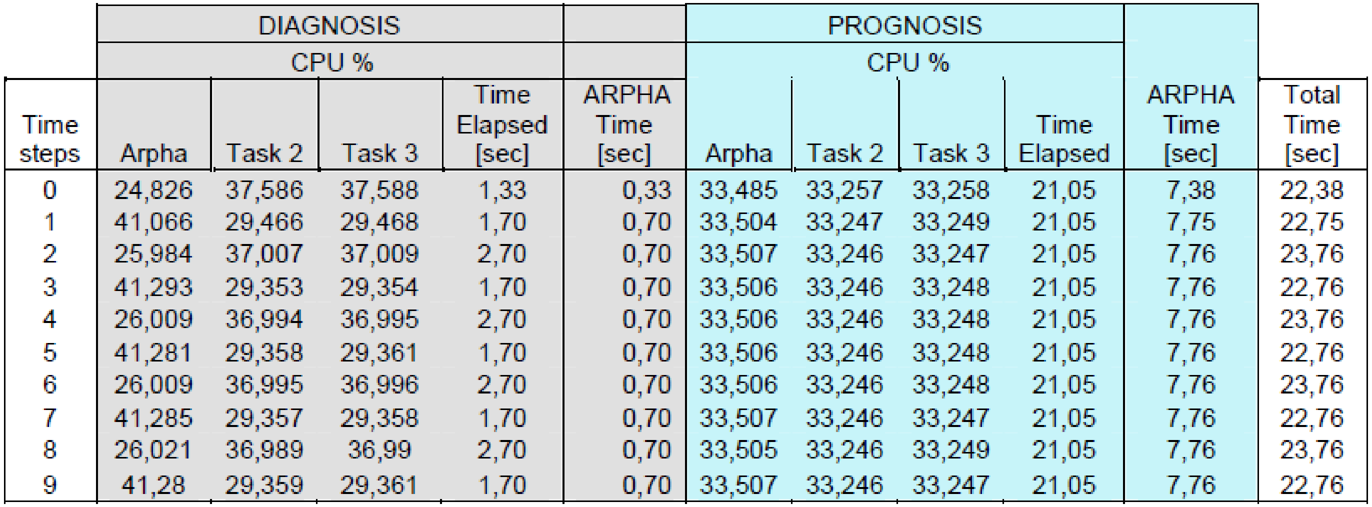
With model of case study. Deadline is given by 8.68 sec (diagnosis) + 43.77 sec (prognosis) + 175.37sec (recovery) = 3 minute and 47 sec
The simpler approach is to use ARPHA in a control loop that allows to perform ARPHA reasoning in 4 minutes. In case diagnosis/recovery is sufficient to guarantee the system safety, ARPHA reasoning could take less then 1 minute.

Processing Power (1/4)

- Inference step affects CPU usage on the base of complexity of the model.
- ARPHA cycle duration depends on complexity of model, number of prognosis steps, duration of recovery policy and type of microprocessor used to run ARPHA
- Using case study model, an ARPHA cycle (diagnosis and prognosis) takes about 1 minute. A single inference cycle takes 9 seconds. In case of recovery the ARPHA cycle can take 4 minutes.

Processing Power (2/4)

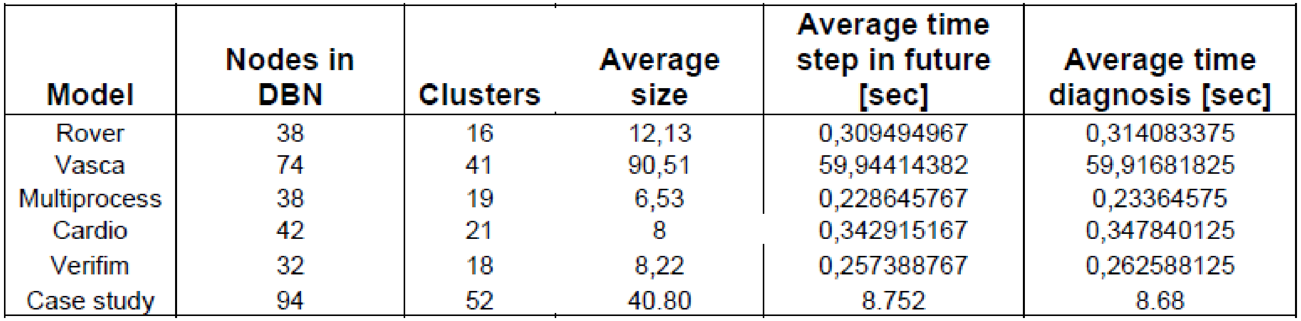
- **CPU usage has been computed by running 2 dummy processes in parallel to ARPHA at the same priority.**



	DIAGNOSIS					PROGNOSIS					
	CPU %					CPU %					
Time steps	Arpha	Task 2	Task 3	Time Elapsed [sec]	ARPHA Time [sec]	Arpha	Task 2	Task 3	Time Elapsed	ARPHA Time [sec]	Total Time [sec]
0	24,826	37,586	37,588	1,33	0,33	33,485	33,257	33,258	21,05	7,38	22,38
1	41,066	29,466	29,468	1,70	0,70	33,504	33,247	33,249	21,05	7,75	22,75
2	25,984	37,007	37,009	2,70	0,70	33,507	33,246	33,247	21,05	7,76	23,76
3	41,293	29,353	29,354	1,70	0,70	33,506	33,246	33,248	21,05	7,76	22,76
4	26,009	36,994	36,995	2,70	0,70	33,506	33,246	33,248	21,05	7,76	23,76
5	41,281	29,358	29,361	1,70	0,70	33,506	33,246	33,248	21,05	7,76	22,76
6	26,009	36,995	36,996	2,70	0,70	33,506	33,246	33,248	21,05	7,76	23,76
7	41,285	29,357	29,358	1,70	0,70	33,507	33,246	33,247	21,05	7,76	22,76
8	26,021	36,989	36,99	2,70	0,70	33,505	33,246	33,249	21,05	7,76	23,76
9	41,28	29,359	29,361	1,70	0,70	33,507	33,246	33,247	21,05	7,76	22,76

Processing Power (3/4)

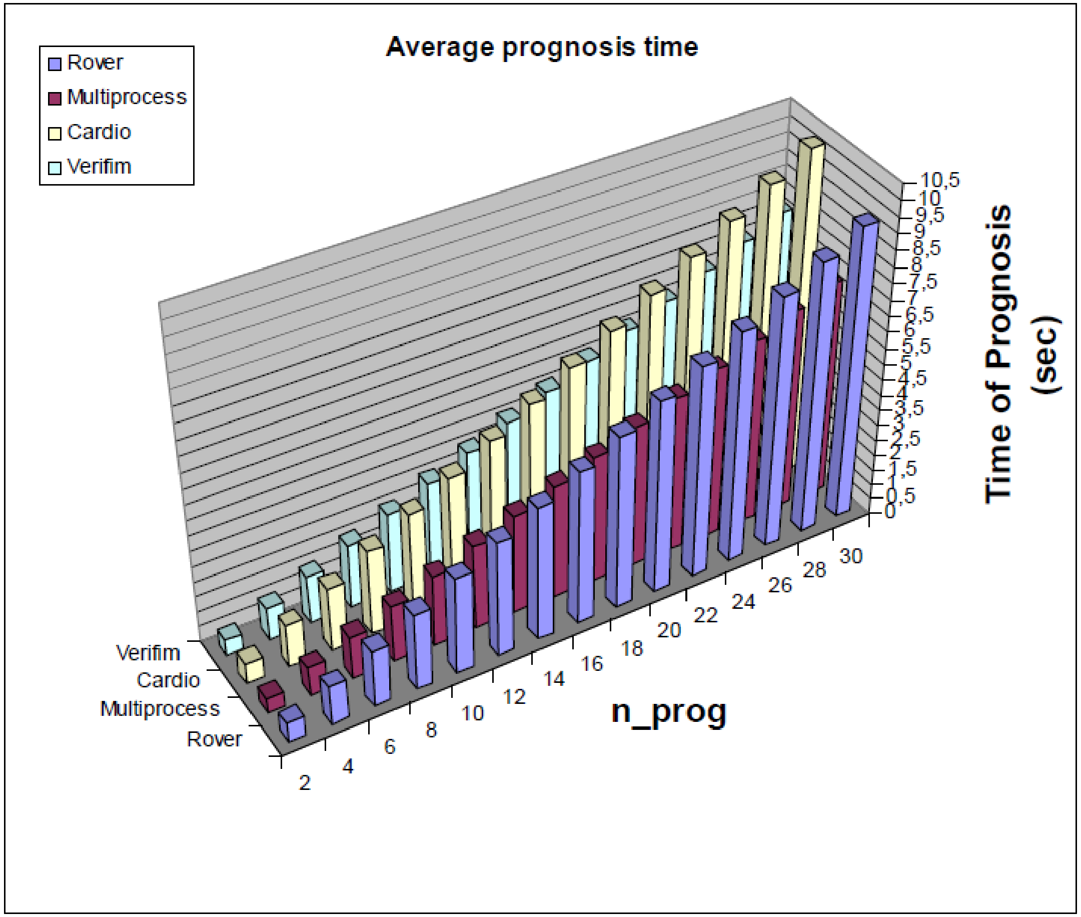
- **In order to evaluate how CPU budget is affected by model size, ARPHA has been run by using 5 models.**



Model	Nodes in DBN	Clusters	Average size	Average time step in future [sec]	Average time diagnosis [sec]
Rover	38	16	12,13	0,309494967	0,314083375
Vasca	74	41	90,51	59,94414382	59,91681825
Multiprocess	38	19	6,53	0,228645767	0,23364575
Cardio	42	21	8	0,342915167	0,347840125
Verifim	32	18	8,22	0,257388767	0,262588125
Case study	94	52	40.80	8.752	8.68

Processing Power (4/4)

Average prognosis time vs n-prog

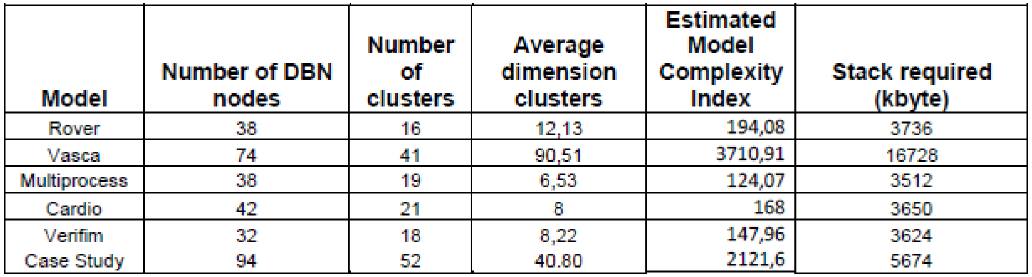


Memory Requirement (1/2)

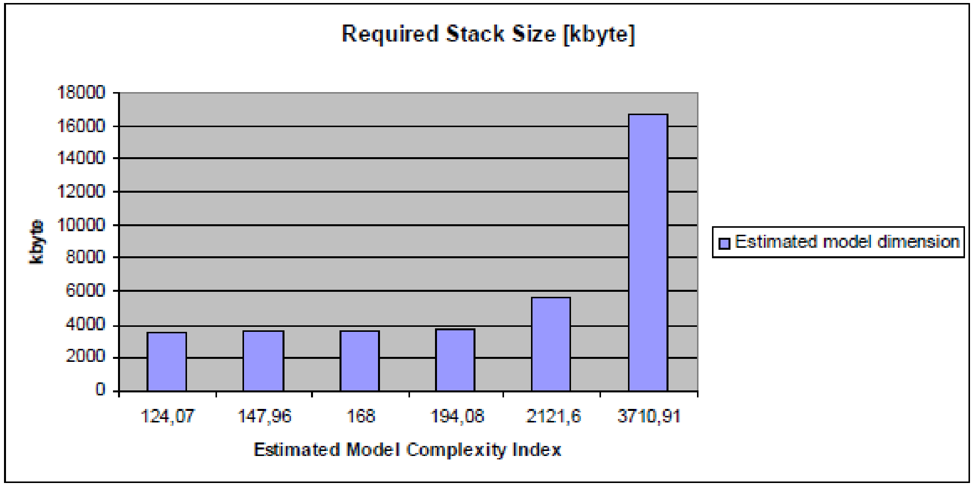
- **For the case study, ARPHA performs inference within 32MB of RAM .**
- **The amount of RAM required by the ARPHA software depended on several factors whose estimation a priori is not easy. There are several parameters that can affect this. For example:**
  - The size and the nature of the model of the DBN considered.
  - The number of activities and the number of observations
- **Code size is 320384 bytes**

Memory Requirement (2/2)

- **ARPHA has been used with 5 models in order to evaluate how stack size changes on the base**
  - Number of nodes in DBN
  - Model complexity



Model	Number of DBN nodes	Number of clusters	Average dimension clusters	Estimated Model Complexity Index	Stack required (kbyte)
Rover	38	16	12,13	194,08	3736
Vasca	74	41	90,51	3710,91	16728
Multiprocess	38	19	6,53	124,07	3512
Cardio	42	21	8	168	3650
Verifim	32	18	8,22	147,96	3624
Case Study	94	52	40.80	2121,6	5674



Industrial prospective

(maturity, pros, cons, gaps to fill)

Maturity of ARPHA (1/2)

- **The approach integrates several modeling formalisms**
  - representing the system behavior at different levels of abstraction
  - performing the system analysis.

- **DFT formalism**
  - system representation in a language familiar to the reliability engineers.
  - not able to capture all the possible kinds of dependency
    - DFT can only capture the aspects represented by its nodes.

- **ARPHA is suitable to be embedded in current on-board computers.**
  - The use of ARPHA could require a microprocessors with higher performance respect to Leon3.
  - ARPHA can be integrated also to run under an operating system different from RTEMS

Maturity of ARPHA (2/2)

- ARPHA approach is suitable w.r.t. mission characteristics as unpredictable local conditions.

- ARPHA could be used by performing a sort of “pipeline” of prognosis:
  - parallel execution of more prognosis inference processes by using different n_prog.

- The use of ARPHA has also an impact on development process.
  - It is necessary to foresee the development of Dynamic Bayesian Network.

Pros (1/2)

- **PROGNOSIS. Besides diagnosis, ARPHA can perform prognosis, with two purposes:**
  - detecting the future belief state,
  - evaluating the future effects of recovery policies
    - in order to select the most suitable to deal with anomalies or failures.

- **RECOVERY.**
  - In traditional on-board FDIR systems, an action to be executed is associated with each anomalous event or failure.
  - In ARPHA, several recovery policies are associated with a certain anomaly or failure
    - each policy consists of a set of actions which may be executed at different times.
  - ARPHA deals with policies instead of single actions
  - The most suitable policy has to be selected to take care of the current anomaly or failure.
    - The policies are evaluated according to their future effect on the system state.

Pros (2/2)

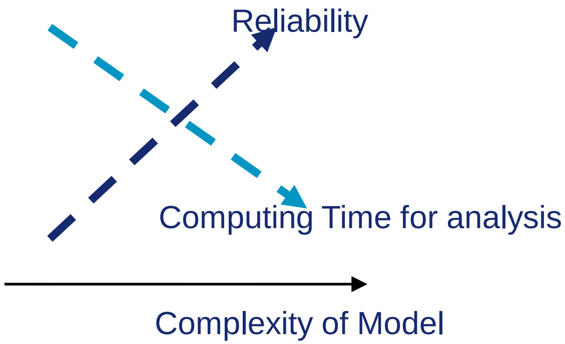
- **MODELING.**
  - DFT formalism allows the system representation in a language familiar to the reliability engineers. (basic events, Boolean gates, dynamic gates).
  - The same stochastic process is represented by the DBN automatically compiled from the DFT.
  - The enrichment of such DBN allows to model other aspects of the system, such as
    - particular variables representing the system state (the intermediate levels of power generation, load, battery charge, battery temperature, etc.)
    - the effect of actions on the state of components (the effect of tilting on the sun aspect angle)
    - probabilistic dependencies (the failure of the battery strings because of mechanical shock)
    - the dynamic behaviour of the system (the load changing according to the current action).

- **OBSERVATIONS CONDITIONED ANALYSIS. The on-board analysis of the model is conditioned by observations.**
  - Diagnosis, prognosis and recovery they take into account information coming from the system and environment monitoring.

- **DIAGNOSABILITY ANALYSIS.**
  - ARPHA can be used as ground support tool for diagnosability analysis.
  - ARPHA avoids the involvement of the ground control thanks to automatic recovery

Cons (1/3)

- **COMPUTING TIME OF ANALYSIS. The time necessary to analyze the model is influenced by the model size.**
  - The complexity of the DBN model depends on the number of entries in the CPTs of variables. The size of CPT depends on
    - the size of variables (number of possible values)
    - the number of parents of the variables.
  - prognosis requires several inferences of the model, for a number of time steps corresponding to the prognosis time horizon.
  - The diagnosis instead, requires a single inference of the model at the current time.



Cons (2/3)

- **MODELLING COMPLEXITY.**
  - The DFT formalism is rather simple (no particular skill in stochastic modelling are required).
  - The DBN can be obtained in automatic way, but its enrichment actually requires a modeller with a specific experience in Bayesian modelling (editing of CPTs in particular)
    - considering any possible case
    - avoiding cases not compatible with observations.

- **SENSOR DATA COLLECTION.**
  - Sensor data are produced by sensors with a high frequency in time (for instance, each second)
  - the ARPHA cycle may take a longer time
  - some sensor data may not be collected and ignored by ARPHA during diagnosis, as a consequence.

Cons (3/3)

- **DISCRETIZATION. Particular measures (power generation level) need to be discretized in order to be modelled by the DBN variables.**
  - The number of discrete intermediate levels must not be high to avoid too many entries in the CPTs.
    - The discretization of variables may lead to some approximation in the model.

- **PROBABILITY PARAMETERS. Component failure rates or other probability parameters may not be immediately available.**
  - they must be estimated or investigated.

- **PROBABILISTIC DETECTION OF STATES. Diagnosis and prognosis in ARPHA provide the current and future belief state of the system respectively.**
  - A belief state is a state detected with a certain probability
  - the state detection and the policy evaluation by ARPHA are not certain.

Gap to fill (1/2)

- **USE OF EDFT INSTEAD OF DFT. The limited modeling power of DFT requires the enrichment of the DBN compiled from the DFT.**
  - During the VeriFIM study, the DFT formalism has been extended into the EDFT formalism
    - Multi-state basic events
    - External actions and control actions
    - SDEP gate (inflencing state transition rates)
  - If an automatic translator from EDFT to DBN was developed, the effort to enrich the DBN would be less relevant

- **IMPROVEMENT OF THE ON-BOARD COMPUTING POWER. The computing time required by ARPHA to perform its complete cycle puts in evidence the necessity to improve the on-board computing power.**
  - especially in the prognosis functions.
  - it could be interesting to run ARPHA on a co-processor

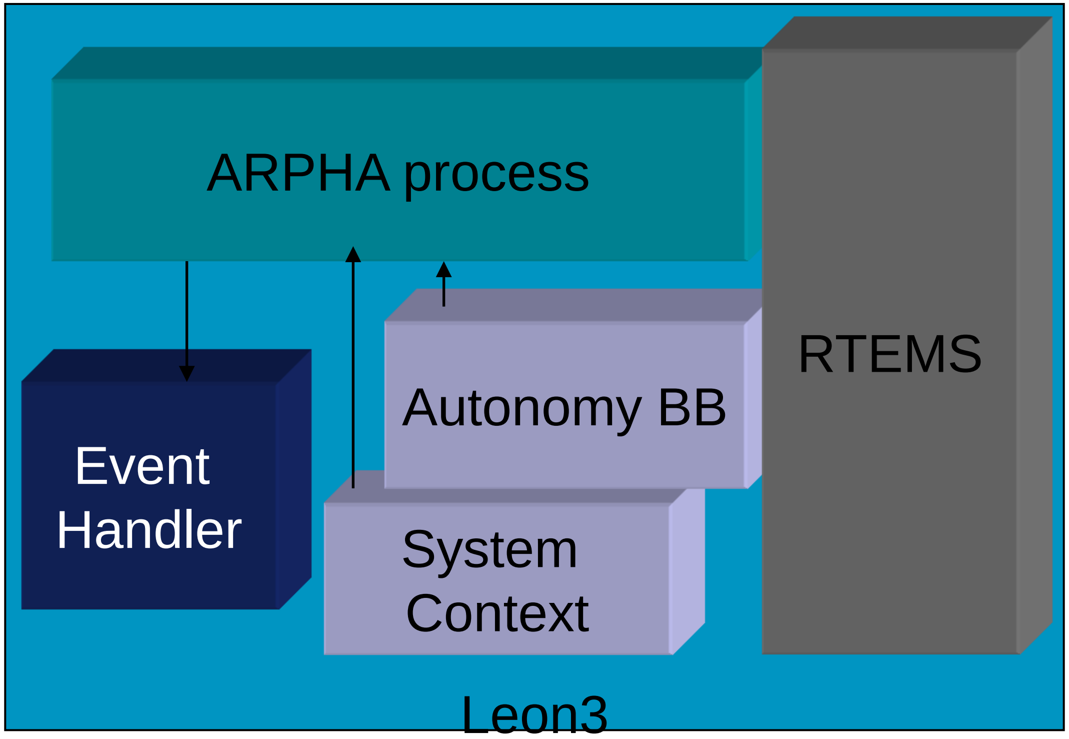
Gap to fill (2/2)

- **KNOWLEDGE ON PROBABILITY PARAMETERS. In order to design an accurate stochastic model, knowledge about probability parameters has to be provided.**
  - Such data may be derived from documentation, experiments, fault injections, testbeds, etc.

- **TRADE-OFF BETWEEN THE MODEL ACCURACY AND THE COMPUTING TIME.**
  - The analysis of an accurate model may take a relatively long time
  - a simpler model may be based on not realistic assumptions
    - its analysis time may be compatible with the current on-board hardware.
  - If ARPHA had to be applied in a real mission, a good trade-off between the model accuracy and the computing time has to be researched.

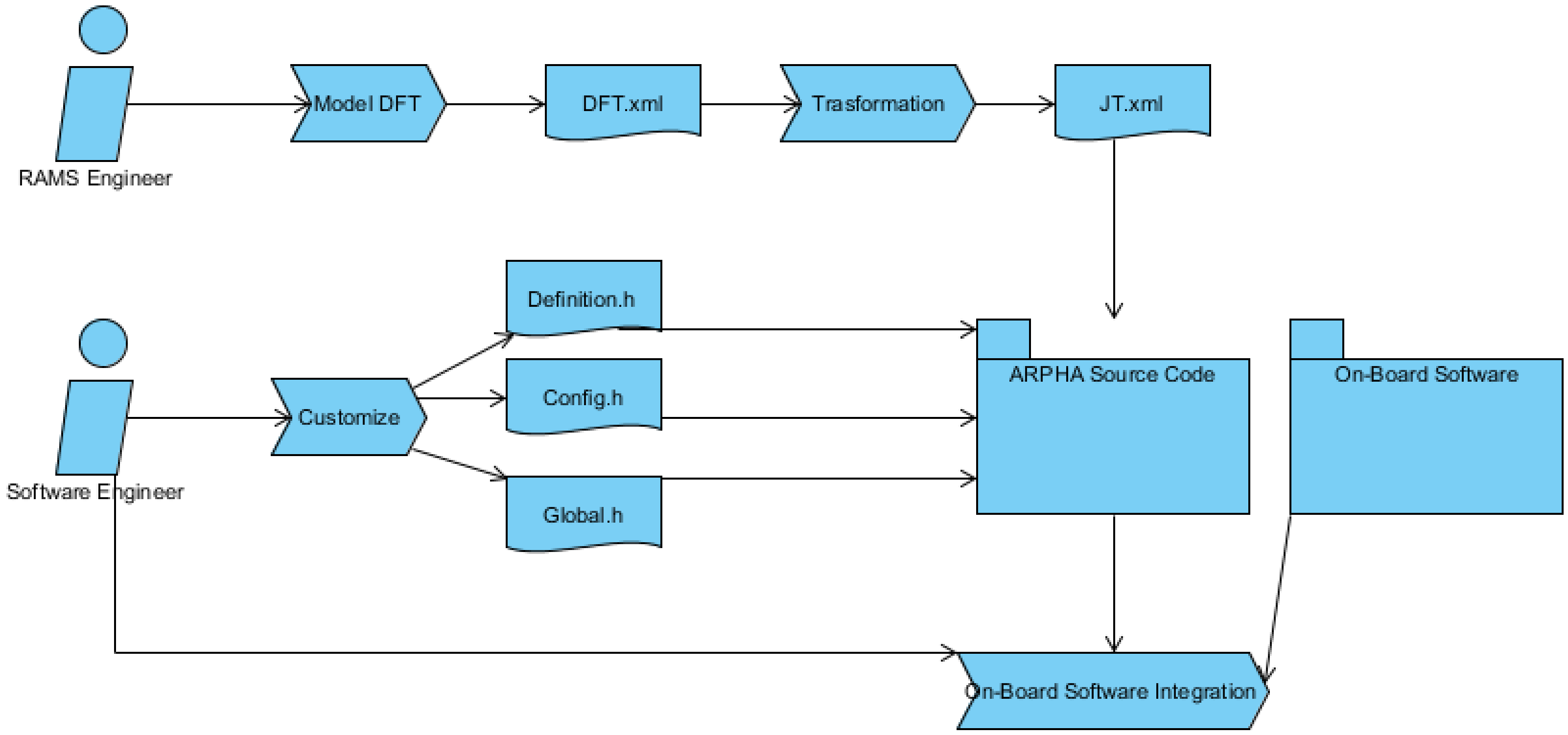
Proposed Software and Hardware architecture

ARPHA will run in parallel to other processes of on-board software.



•policy event triggered by recovery is managed by Event Handler.
•ARPHA has not to wait the conclusion of a recovery policy to perform a new diagnosis or prognosis on the system,
•ARPHA can consider also the changes performed during execution of recovery policy to perform a new diagnosis or recovery.

Development Process



Conclusion (1/2)

- The developed approach provides a unified modeling and autonomous framework that integrates
  - an high level modeling formalism (DFT),
  - a low level modeling formalism (DBN)
  - and an inference oriented formalism (JT).
- The on-board analysis of the JT conditioned by the sensors data and the recovery actions, allows to evaluate the system current and future state, and the recovery policies if necessary, in automatic way, without the assistance of the ground control.
- This approach increases the achievable level of autonomy.
- The developed prototype ARPHA represents an on-board software FDIR component suited for use in the existing spacecraft system architectures. It can perform on-board diagnosis, prognosis and recovery inference.

Conclusion (2/2)

- ARPHA is able to verify the failure impact on the future state of the system
- Environmental aspect of space mission can be modeled in the DBN used by ARPHA to perform inference.
- It is possible to take in account the failure causes, by inserting them in the utility function used to select recovery.
- ARPHA can evaluate the failure impact on the currently executing plan as well.
- The developed ARPHA prototype has been evaluated on the space embedded target (running under RTEMS on the LEON3 processor). The obtained performance data shows ARPHA usability in the context of the current space applications and available on-board computers.

Dissemination

L. Portinale, D. Codetta-Raiteri, "Using Dynamic Decision Networks and Extended Fault Trees for Autonomous FDIR", Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI), IEEE, Boca Raton, USA, November 2011.

L. Portinale, D. Codetta-Raiteri, "ARPHA: an FDIR architecture for autonomous spacecrafts based on Dynamic Probabilitstic Graphical Models ", Proceedings of the AI in Space Workshop @ IJCAI, Barcelona, Spain, July 2011.

D. Codetta-Raiteri, L. Portinale, S. Di Nolfo, A. Guiotto, "ARPHA: a software prototype for fault identification, detection and recovery in autonomous spacecrafts", paper submitted to ACTA Futura journal, published by ESA.

D. Codetta-Raiteri, L. Portinale, S. Di Nolfo, A. Guiotto"Innovative fault identification, detection and recovery in autonomous spacecrafts", abstract submitted to Infotech@Aerospace, Garden Grove, CA USA, 2012.

Special issue on "Application of Bayesian Networks", International Journal of Approximate Reasoning, Elsevier.