FAME PROCESS: A DEDICATED DEVELOPMENT AND V&V PROCESS FOR FDIR

Andrea Guiotto[(1)], Regis De Ferluc[(2)], Marco Bozzano [(3)], Alessandro Cimatti[(3)], Marco Gario[(3)] Yuri Yushtein[(4)]

[(1)]Thales Alenia Space Italia, Strada Antica di Collegno 253, 10146 Torino, Italy, Andrea.guiotto@thalesaleniaspace.com
[(2)]Thales Alenia Space France, 5, allee des Gabians, 06156 Cannes la Bocca, France, Regis.Deferluc@thalesaleniaspace.com
[(3)]Fondazione Bruno Kessler, Via Sommarive 18, Povo, 38123 Trento Italy, {bozzano,cimatti.gario}@fbk.eu
[(4)]ESA - European Space Agency, Keplerlaan 1, PO Box 299, 2200AG Noordwijk, The Netherlands, yuri.yushtein@esa.int

ABSTRACT

In the frame of the European Space Agency (ESA) studies, Thales Alenia Space Italia has carryed out a research – FAME – in collaboration with Fondazione Bruno Kessler and Thales Alenia Space France. The objective of the FAME project was to define a dedicated FDIR development, verification and validation process that can address the issues and shortcomings of the current industrial FDIR development practices. The ultimate goal was to allow for the consistent and timely FDIR conception, development, and Verification & Validation. A parallel objective of the study was the development of a toolset supporting the Process and enabling a coherent definition, specification, development, and V&V of the FDIR functionalities. It started in September 2013 and ended in May 2014.

1. INTRODUCTION

Current practices of FDIR development utilize the results of System RAMS analyses (FMEA/FMECA, FTA, and HSIA). The main shortcoming of this approach is the conflict between FMEA (bottom-up approach) and FTA (top-down approach). Besides, data for these analyses becomes available later in the Systems Engineering process, when Software Development has passed its initial phases. There is also the necessity to write requirements to be more than "Do Fault Protection".

Another challenging domain is FDIR architecture. Often FDIR architecture is a heritage of previous project. This effectively brings to another shortcoming. There is the need to be supported in the architectural analysis and trade-offs phase based on the specific mission objectives, priorities, operations perspective, and concrete system architecture and design. It is necessary to better estimate and control the costs of the products and of the process.

Often, the high overall FDIR complexity limits possibility for the fault coverage analysis, and impairs effective V&V process. It is difficult to determine the propagation of failure in terms of time. These shortcomings could be improved with the following proposed solutions:

- Conflict between bottom-up approach and top-down approach for fault identification methods can be resolved starting from functional analysis that is available from early phases of project to identify a failures catalogue
- Value analysis is necessary to determine benefits of additional HW and SW
- a contract-based approach can be used to support the specification, verification and validation of the FDIR implementation with respect to the formal model of the design
- Requirements derived by FDIR Analysis are more than "Do FP" and must be inserted in specification at RB level. So it is needed to anticipate before SRR the FDIR Analysis
- Improve the generation of FDIR artifacts as FTA, FMEA tables, observability effectiveness analysis
- Timed Failure Propagation Graphs (TFPGs) can be used to determine the time evolution of failure in order to assure that recovery is performed in time.

A TFPG [1] is a directed graph model that represents temporal progression of failure in physical systems. Nodes of the graph represent either failure modes, which are fault causes, or discrepancies, which are off-nominal conditions that are the effects of failure modes; edges between nodes in the graph capture the effect of failure propagation over time in the underlying dynamic system, and specify timing constraints on fault propagation. TFPGs can be used for both fault diagnosis and fault prognosis.

In the next sections, the FAME process and FAME Environment developed to support it are described. They represent a possible answer to needs, challenges and shortcoming of current FDIR development approach.

2. THE FAME PROCESS

FAME process (figure 1) is composed by 6 activities and covers phase B, C and D. The FAME process is technology-independent and is described by using SPEM2.0 (System& Software Process Engineering Metamodel). In the SPEM 2.0 Meta-Model, processes are represented with activities, tasks, steps, artifacts, roles and milestones. At the beginning of the FAME process System Engineer performs Analyze User Requirements activity that is composed by the following tasks:

- Define RAMS and Autonomy Requirements
- Build Mission Phase/Spacecraft Operational Mode matrix

It starts at begin of Phase B and ends before System SRR.

Then FDIR Engineer performs ***Define Partitioning/allocation activity*** that is composed by the following tasks:

- Define Partitioning/allocation
- Define Architecture

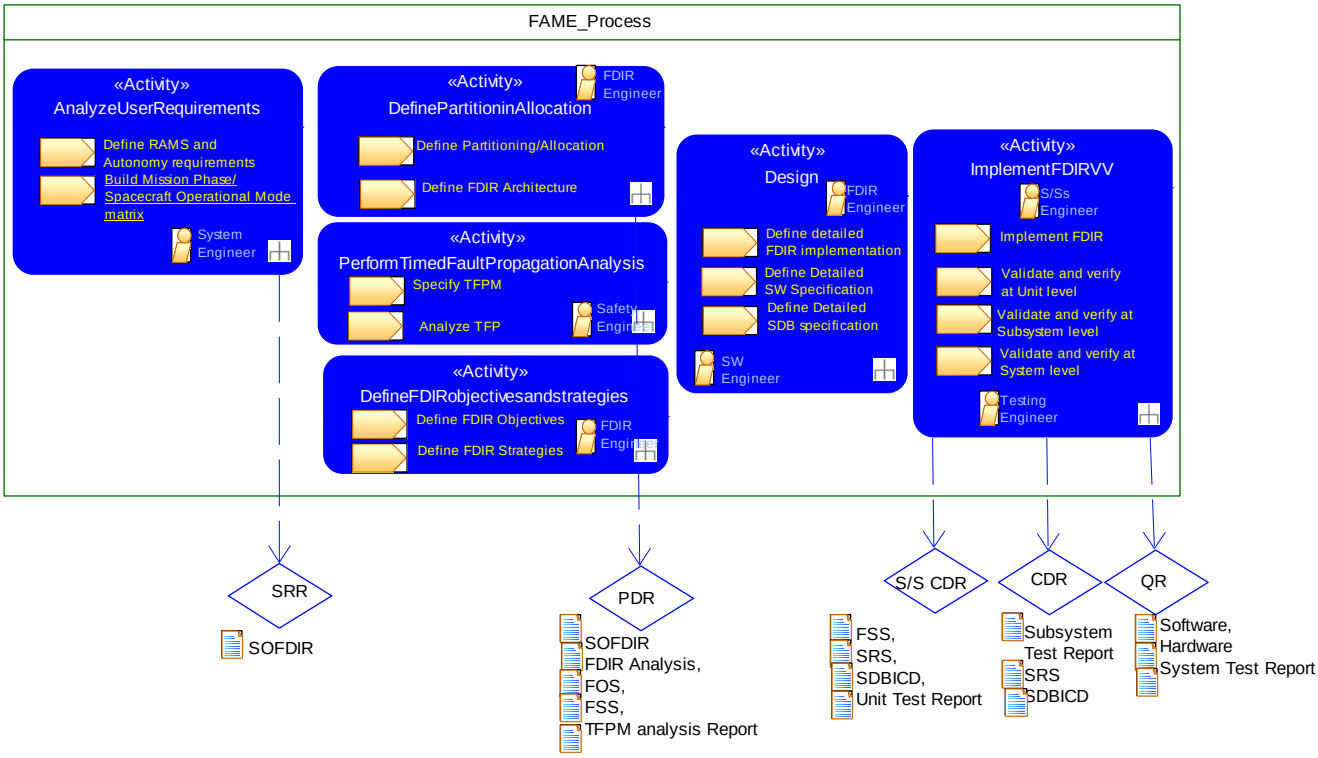

Figure 1. Overview of FAME Process

It starts ***after System SRR*** and ends ***System PDR.*** When defining the FDIR reference architecture, it is important to take into account the distribution of the FDIR functionalities among different components. FDIR components can be distributed, hierarchical or a combination of both, thus providing enough flexibility to cover a wide range of architectural solutions. Two activities can start in parallel. The first one is ***Define FDIR objectives and strategies***. It is performed by FDIR Engineer and it is composed by the following tasks:

- Define FDIR objectives (contained in FOS)
- Define FDIR strategies (contained FSS)

It starts ***after System SRR*** and ends at ***System PDR.*** The second one is ***Perform Timed Fault Propagation Analysis***. It is performed by Safety Engineer and it is composed by the following tasks: Specify TFPM and Analyze TFPM. TFPM stands for Timed Failure Propagation Model.

The safety engineer specifies a TFPG for the design starting from fault trees, FMEA tables and Hazard Analysis. Then ***Design activity*** can start. It is composed by the following tasks:

- Define detailed FDIR implementation performed by FDIR Engineer
- Define Detailed SW Specification performed by SW Engineer
- Define Detailed Spacecraft Data Base specification performed by SDB Engineer

It starts ***at System PDR*** and ends ***S/S CDR.*** At the end it is possible to accomplish ***Implement FDIR, V&Vactivity***. It is composed by the following tasks:

- Implement FDIR performed by Subsystem Engineer
- Validate and verify at Unit level performed by test engineer
- Validate and verify at Subsystem level performed by test engineer
- Validate and verify at System level performed by test engineer.

It starts ***at S/S PDR*** and ends ***System QR.***

3. THE FAME ENVIRONMENT

The design and implementation of the FAME environment is based on the COMPASS toolset [2,3]. The COMPASS toolset is based on formal methods, which offer a wide range of techniques for system verification and validation. An AADL-like language called SLIM language (System-Level Integrated Modelling Language) [2] is used within COMPASS for modelling the system architecture and behaviour. FAME also inherits ideas from AUTOGEF (Automated Generation of FDIR for the COMPASS integrated toolset) [4]. FAME relies on TFPG technology to specify fault propagation, and on the technologies for synthesis of FD from a TFPG, and for synthesis of FR using model-based planning.

The architecture of the FAME environment and its relationship with COMPASS are summarized in Figure 2.

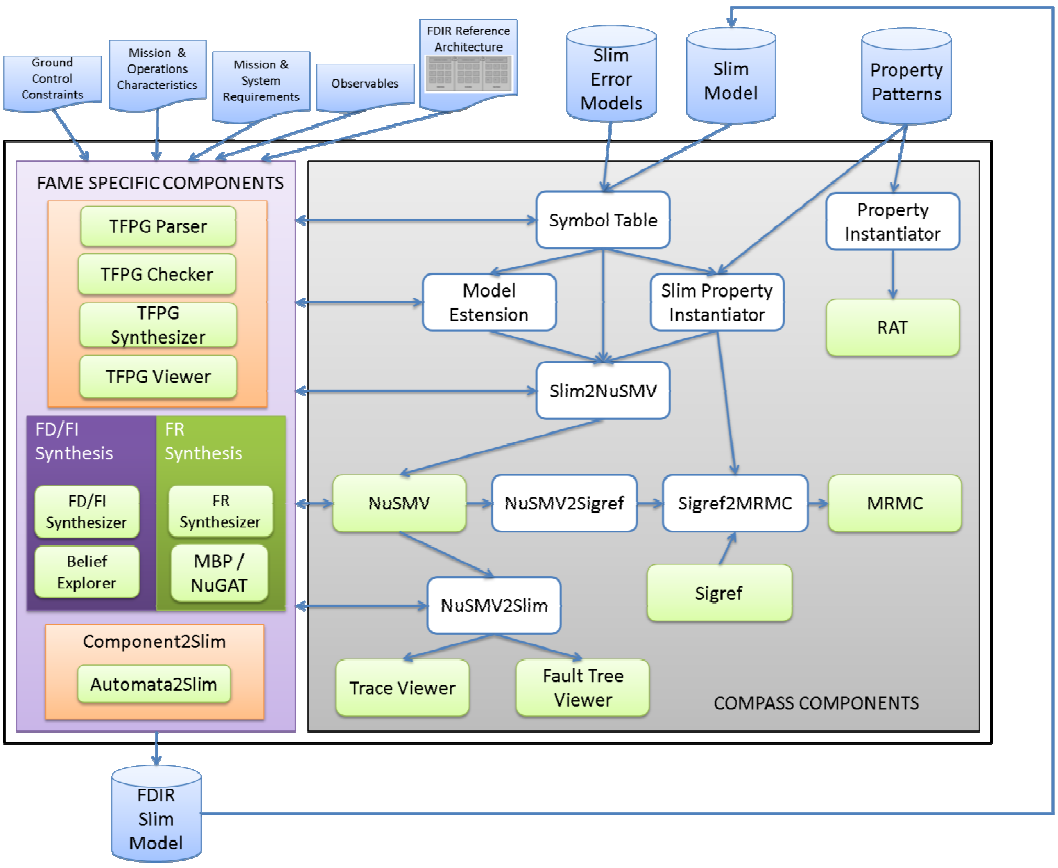

Figure 2. Architecture of FAME Environment

The FAME environment takes several inputs, including functional and error models, mission characteristics, mission and system requirements, and property patterns; it generates as output an FDIR model, which can be combined with the original system models. Several components of the COMPASS toolset (right part of Figure 2) are re-used, with some adaptations, in FAME. FAME specific components, illustrated on the left of Figure 2, can be grouped in the following categories:

- Components for TFPG management, namely those that allow to parse an input TFPG, visualize a TFPG in a graphical form, check the syntax and behavior of a TFPG against the original model, and automatically synthesize a TFPG
- Components that implement automatic synthesis of an FD model (automaton) starting from a TFPG, using techniques based on belief space exploration, and automatic synthesis of an FR model (automaton), using model-based planning techniques
- Components that enable the translation of the automata for FD and/or FR synthesized in the previous step, into SLIM

The main capabilities (figure 3) provided by the FAME environment are:

- ***System Modeling*** and verification framework inherited from COMPASS. This includes: use of formal models, written in the SLIM language, for nominal models, error models and FDIR models; model extension and ***fault injection*** to automatically extend the nominal models with error specification; definition of properties using property patterns; formal verification techniques, based on model checking, that cover a broad range of activities (functional verification, safety assessment, FDIR effectiveness analysis, performability analysis, etc.)
- ***Mission Modeling***: Definition of mission phases and operational modes, mission requirements and FDIR requirements, to specify the desired requirements on the FDIR
- ***TFPG Modeling***: Modeling of fault propagation using TFPGs (Timed Failure Propagation Graphs) and Automatic synthesis of a TFPG from a model.
- **TFPG Analysis**: Analyses of TFPGs
    - Behavioral validation, to check compliance of a TFPG with respect to a SLIM model

- Effectiveness validation, to check suitability of a TFPG for implementing a diagnoser

- ***FDIR Synthesis***: Automated synthesis of an FDIR model
  - Synthesis of an FD model from a TFPG
  - Synthesis of an FR model, using conformant planning routines

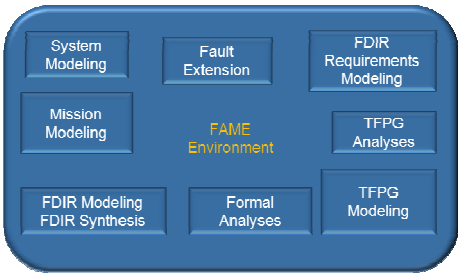

Figure 3. Capabilities of FAME Environment

FAME process can be supported by FAME environment capabilities as described in the following table:

FAME activity	**FAME Environment capabilities**
Analyze User Requirements	System Modeling Mission Modeling Formal Analysis Fault Extension
Define Partitioning/allocation activity	System Modeling Formal Analysis
Define FDIR objectives and strategies	FDIR Requirements Modeling
Perform Timed Fault Propagation Analysis	TFPG Analysis TFPG Modeling Formal Analysis
Design activity	FDIR Modeling FDIR Synthesis Formal Analysis
Implement FDIR, V&Vactivity.	N/A

Table 1. Process vs. Environment

4. FAME PROCESS APPLIED TO A CASE STUDY

The FAME process and environment is evaluated trough a case-study derived from the EXOMARS mission, focusing on the Trace Gas Orbiter (TGO) system which is to be launched in 2015 towards the Mars planet. An example of feared event for the considered system is a "TGO erroneous attitude". The case-study involves the following components:.

- 2 Processor Modules (PM) in cold or hot redundancy with Central Software. that includes Guidance and Navigation Control Application .
- Inertial Management Unit (IMU): 2 Gyroscopes in cold or hot redundancy.

For the perimeter of the case study, it has been addressed one mission phase: Mars Orbit Insertion (MOI) phase.
In this phase, the TGO has several operational modes. For the case-study, the following operational modes have been considered:

- Routine – (ROUT)
- Manoeuvre in critical conditions – (MAN-C)
- SAFE-1
- SAFE-2

In the following sections, it is described the use of main capabilities of FAME following the FAME process. In this way, the applicability of the process itself to industrial projects has been evaluated, and also its compliance with applicable standards.
in addition, this case study has allowed to experiment once again on formal methods applied for Failure Management analysis and FDIR synthesis, an specifically to evaluate the novel contributions of the project addressing the analysis of timed fault propagation in the system (using TFPG) and the corresponding diagnose synthesis.

4.1 System Modeling and Fault Extension

The first activity has been to insert in FAME Environment the nominal model and error model of system as SLIM files. For the evaluation, the feared events generated coming from the IMU realizing the acquisition of the spacecraft attitude have been analyzed. Using documentation and FMECA from IMU equipment supplier, the IMU FMECA Items have been analysed and those having impact on the system have been selected: Sensor Signal is too low, Sensor output is biased, Sensor output is erroneous. SLIM model have been refined with the error models related to the IMU Acquire Attitude function. The FMECA items have been translated to error events, the failure modes have been translated to error states, and the local / system effects have been used to define the fault injection. The IMU failure propagation has been described

4.2 FDIR Requirement Modeling

Then a set of requirements coming from the Exomars TGO project have been analysed and derived to produce FDIR objectives, strategies, and specification

Req. ID	**FDIR requirements**
FAME-SUB-CASE-STUDY-FDIR-REQ-010	Mission shall be ensured for any single failure
FAME-SUB-CASE-STUDY-	TGO shall be able to achieve its manoeuvres of Mars Orbit Insertion

FDIR-REQ-020	even in case of single failure.

Table 2. FDIR Requirements

Then a set of objectives has been derived

Req. ID	FDIR requirements
[FAME-SUB-CASE-STUDY-FDIR-OBJ-010]	If IMU failure item "FAME_IMU_001" occures during phase "MOI" and mode "MAN_C", TGO shall be able to carry on the manoeuvre.
[FAME-SUB-CASE-STUDY-FDIR-OBJ-020	If IMU failure item "FAME_IMU_001" occures during phase "MOI" and mode "ROUT", TGO shall not start the manoeuvre and go to SAFE mode.

Table 3. FDIR Requirements

To reach the previously defined FDIR objectives, FDIR strategy is defined, using information from Feared Event Table and FMECA analysis.

Req. ID	FDIR requirements
[FAME-SUB-CASE-STUDY-FDIR-STR-010]	If a failure occures on the nominal IMU during phase "MOI" and mode "MAN_C", the TGO system shall autonomously switch to redundant unit.
[FAME-SUB-CASE-STUDY-FDIR-STR-011]	If a failure occures on the redundant IMU during phase "MOI" and mode "MAN_C", the TGO system shall reset this redundant unit and try to carry on the manoeuvre.

Table 4. FDIR Strategies

4.3 Mission Modeling

The Mission Phase and Operational modes selected for the case study are inserted in FAME Environment. Then Spacecraft configuration are specified and associated to Operational modes.

Figure 4. Mission Specification in FAME Environment

4.4 FDIR Specification

FDIR specification output from previous activities is entered in the FAME environement.

4.5 TFPG Modeling and TFPG Analysis

From the SLIM model enhanced with timing aspects and the error model, the TFPG model for IMU_1 failures is defined manually or using the TFPG synthesizer tool (see figure 6). The TFPG should be consistent with the system model behavior The behavioural validation allows to identify wrong values for timing on the TFPG edges. The effectiveness validation allows to identify the failure modes that are not diagnosable in the different modes

4.6 FDIR Modeling and FDIR Synthesis

The fault detection synthesis is run based on the fault detection specification (figure 6) and the fault recovery synthesis is run based on the fault recovery specification (figure 7).

5. APPROACH CHARACTERIZATION

The approach has been evaluated in terms of adequacy, effectiveness and usability with respect to FAME process, FAME Methodology and FAME Environment. FAME process is adequate because it is compliant with the current project life cycle in terms of respect of phases and reviews and compliant with applicable standards. It is independent from any tools. FAME process is effective in the initial phases where FDIR is not yet defined and a clear definition of check point guarantees an optimization of time spent for each activity by avoiding to waste time and effort to accomplish premature tasks. FAME process is usable because it can be inserted easily in the current industrial process, but the use of TFPG requires a training of users in order to learn the methodology. FAME methodology is adequate because TFPG is based on the identification of failure mode and discrepancies, and transitions between discrepancies, but TFPG complexity can be critical since it depends on number of nodes and edges and by temporal constants in use. In any case, Slim generated by synthesis can be analyzed by using COMPASS features as correctness. FAME methodology is more effective if SLIM models used in the FAME process are not created from scratch, but are derived from existing models of the system. However, the application of the FAME methodology to the space domain may be limited by the state-space explosion when introducing time on complex models. FAME methodology to be usable must be adopted in an incremental way, considering small subset of failures, and taking into account the assumptions related to these failures. At the end, all the results should be combined in order to generate FD and a FR modules that covers the entire set of FDIR specification for the entire set of failures in the system, and therefore taking into consideration all the TFPGs. FAME environment is not yet adequate to manage several input and outputs files and elaboration time depends too much on complexity of TFPG for what concerns the synthesis of detection. FAME environment is effective only with simple TFPG. The usability of tool is good. Changes on TFPG textual file are reflected in graphical view (roundtrip is good).

6. CONCLUSIONS

FAME process foresees Functional Analysis that can be used early in the process with a positive effect on the eventual FDIR maturity. Failure propagation can be analyzed with TFPG. FAME process is phased and can be

Component	Error State	Failure Mode	Generated alarm	Predefined alarm	Enabled
▼ TGO.AcquireAttitude_Block.AcquireAttitude1					
	MEASURES_NONE	FM_AcquireAttitude_MEASURES_NONE_1	NO_MEAS_1		☑
	MEASURES_ERRONEOUS	FM_AcquireAttitude_MEASURES_ERRONEOUS_1	ERR_MEAS_1		☑
	MEASURES_BIASED	FM_AcquireAttitude_MEASURES_BIASED_1	BIASED_MEAS_1		☑
▼ TGO.AcquireAttitude_Block.AcquireAttitude2					
	MEASURES_NONE	FM_AcquireAttitude_MEASURES_NONE_2	NO_MEAS_2		☑
	MEASURES_ERRONEOUS	FM_AcquireAttitude_MEASURES_ERRONEOUS_2	ERR_MEAS_2		☑
	MEASURES_BIASED	FM_AcquireAttitude_MEASURES_BIASED_2	BIASED_MEAS_2		☑

Figure 5. Fault Detection Specification

FDIR Specification | Synthesis

Targets
Controllables
Patterns
FD Table
FR table

FR Table

Alarm	Phase	Op-mode	Severity	Target mode	Target conf	Target constraints	Allowed Recovery Actions	Predefined recovery
▼ NO_MEAS_1								
▼	MOI							
		MAN_C	2 (Critical)	MAN_C	IMU_2		all	
		ROUT	1 (Catastrophic)	SAFE	IMU_2		all	
		SAFE	1 (Catastrophic)	SAFE	IMU_2		all	
▼ ERR_MEAS_1								
▼	MOI							
		MAN_C	2 (Critical)	MAN_C	IMU_2		all	
		ROUT	1 (Catastrophic)	SAFE	IMU_2		all	
		SAFE	1 (Catastrophic)	SAFE	IMU_2		all	
▼ BIASED_MEAS_1								
▼	MOI							
		MAN_C	2 (Critical)	MAN_C	IMU_2		all	
		ROUT	1 (Catastrophic)	SAFE	IMU_2		all	
		SAFE	1 (Catastrophic)	SAFE	IMU_2		all	

Figure 6. Fault Recovery Table

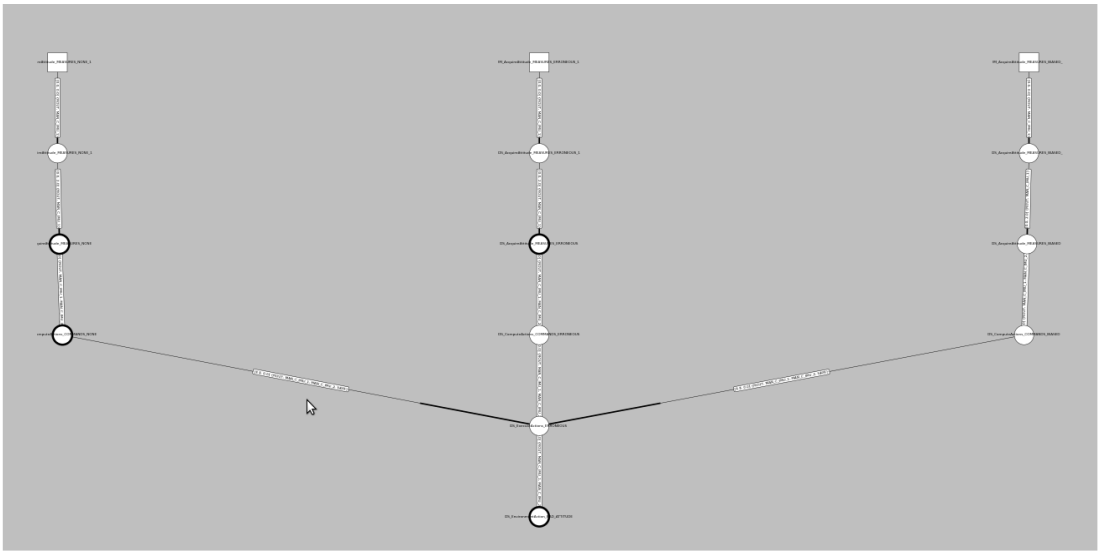

Figure 7. TFPG of Case Study

employed starting from the early system development phases, and which is able to take into account the design and RAMS data from both, Software and System perspective. Since FAME process includes list of checkpoints, list of roles, list of artifacts and rules to checking consistency of FAME process, it guarantees an optimization of time spent for each activity by avoiding to waste time and effort to accomplish premature tasks.

Several interesting aspects related to FDIR architectural specification could be considered in future extensions of the FAME environment/toolset. These extensions can be applied either at the FDIR Level (FDIR Architecture - Centralized/Distributed/Decentralized and Hierarchy Levels), or at the level of each single Requirement (Scope, Context).

System hazards should be considered when modeling fault propagation using TFPGs. System hazards could be inserted in the TFPG using an additional node category. It has to be investigated the relationships between hazards and failure modes/discrepancies: a hazard may cause a failure mode, and may propagate by activating discrepancies.

It is possible to extend the current TFPG synthesis algorithm with a dedicated procedure specifically for synthesizing the timing bounds and the enabling system modes of edges. Another idea of extension is to support a Contract-Based Design (CBD) flow integrating contract-based specification and verification techniques.

7. REFERENCES

1. Sherif Abdelwahed and Gabor Karsai. Failure Prognosis Using Timed Failure Propagation Graphs. In Electrical Engineering, 2007.
2. M.Bozzano, A.Cimatti, J.-P.Katoen, V. Y.Nguyen, T.Noll and M.Roveri. Safety, Dependability, and Performance Analysis of Extended AADL Models. The Computer Journal, 54(5):754-775, 2011.
3. COMPASS project and Integrated Tool-set. Webpage: http://compass.informatik.rwth-aachen.de
4. E. Alaña, H. Naranjo, Y. Yushtein, M. Bozzano, A. Cimatti, M. Gario, R. de Ferluc, G. Garcia. Automated generation of FDIR for the compass integrated toolset (AUTOGEF). In Proceedings of DASIA 2012. Dubrovnik, Croatia. May 14-16, 2012.