FAME: A Model-Based Environment for FDIR Design in Aerospace

Benjamin Bittner[1], Marco Bozzano[1], Alessandro Cimatti[1], Regis De Ferluc[2], Marco Gario[1], Andrea Guiotto[3], and Yuri Yushtein[4]

[1] Fondazione Bruno Kessler, Trento, Italy
[2] Thales Alenia Space, France
[3] Thales Alenia Space, Italy
[4] European Space Agency (ESA), ESTEC, Noordwijk - The Netherlands

1 Introduction

The FAME environment is a model-based toolset that implements an integrated process for FDIR (Fault Detection, Isolation and Recovery) design, addressing the shortcomings of existing practices for FDIR development in aerospace [2]. It is built on top of COMPASS [4], a framework for model-based design and verification, that provides several verification capabilities, including simulation, property verification, RAMS analysis (FTA, FMEA), diagnosability and FDIR analysis. The FAME environment supports FDIR design by providing functionality to define mission and FDIR requirements, fault propagation modeling using TFPGs (Timed Fault Propagation Graphs), and automated synthesis of FDIR models from TFPGs and FDIR requirements. The FAME environment has been developed within an ESA-funded study, and has been thoroughly evaluated by the industrial partners on a case study derived from the ExoMars project.

2 The FAME Environment

We summarize the functionalities offered by the FAME environment, in particular focusing on those that are specific of FAME. Modeling, fault injection, and several formal analyses, including property verification, simulation, safety assessment (e.g., FTA, FMEA - both qualitative and quantitative), FDIR effectiveness analysis and performability analysis, are inherited from COMPASS [4].

Mission Specification The FAME environment provides a dedicated activity to specify the Mission Phase/Spacecraft Operational Mode matrix. It consists of the list of phases and operational modes, and S/C configurations. Phase/mode pairs can be tracked by FDIR to contextualize its strategies.

FDIR Requirements Specification FDIR requirements can be specified for the system level and the subsystem level, possibly linked to specific phase/-mode combinations. The FAME environment enables the user to specify S/C configurations as possible recovery targets, specify alarms that need to be fired with certain error states, and integrate existing FDIR components (if available).

Fault Propagation Analysis FAME uses TFPGs to model temporal interactions of failures and their effects [1]. It supports loading, syntactic verification, displaying, and editing of TFPGs. It furthermore supports checking whether the system exhibits failure propagations not captured by the TFPG (behavioral validation). The tool allows to check the TFPG's adequacy as a model for diagnosis, using diagnosability analysis [5]. Finally, a function based on FTA is available which allows to automatically synthesize a basic TFPG.

FDIR Synthesis The FAME environment supports fully automated synthesis of diagnosis (FD) and recovery (FR) components, starting from an extended SLIM model, a TFPG, and an FDIR specification. FD synthesis [3] creates a diagnoser that generates a set of alarms for each specified failure, by monitoring the available sensors. An FR model can be derived using conformant planning [6] - it provides for each specified alarm and phase/mode pair a recovery plan, i.e., a sequence of actions that guarantee to achieve the target.

2.1 Implementation and Distribution

The FAME toolset is implemented on top of COMPASS [4]. The GUI and most sub-components are implemented in Python, using the PyGTK library. Overall, the core consists of more than 100,000 lines of Python. The analysis capabilities of FAME are provided by the underlying model checking engines, in particular the NuSMV and the MRMC model checkers [4], that are written in C. The FAME toolset is distributed under the FAME Public License, a variant of GNU GPL, whereas some backends are licensed under FBK's Additional Components License. The tool is available for download to anyone within the ESA member states. Additional details may be found on the FAME Project web page [7].

References

1. S. Abdelwahed, G. Karsai, N. Mahadevan, and S.C. Ofsthun. Practical implementation of diagnosis systems using timed failure propagation graph models. *Instrumentation and Measurement, IEEE Transactions on*, 58(2):240–247, 2009.
2. B. Bittner, M. Bozzano, A. Cimatti, R. de Ferluc, M. Gario, A. Guiotto, and Y. Yushtein. An Integrated Process for FDIR Design in Aerospace. 2014. In Proc. IMBSA 2014.
3. M. Bozzano, A. Cimatti, M. Gario, and S. Tonetta. A formal framework for the specification, verification and synthesis of diagnosers. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.
4. 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*, doi: 10.1093/com, March 2010.
5. A. Cimatti, C. Pecheur, and R. Cavada. Formal Verification of Diagnosability via Symbolic Model Checking. In *International Joint Conference on Artificial Intelligence (IJCAI 2003)*, pages 363–369. Morgan Kaufmann, 2003.
6. A. Cimatti, M. Roveri, and P. Bertoli. Conformant planning via symbolic model checking and heuristic search. *Artificial Intelligence*, 159(1):127–206, 2004.
7. The FAME Project. `http://es.fbk.eu/projects/fame`.