ConcertoFLA-based Multi-concern Assurance for Space Systems

Zulqarnain Haider, Barbara Gallina

Mälardalen University, P.O. Box 883, SE- 721 23 Västerås, Sweden; zulqarnain.haider@mdh.se barbara.gallina@mdh.se

Anna Carlsson

OHB Sweden, P.O. Box 1269, SE- 16429 Kista, Sweden; anna.carlsson@ohb-sweden.se

Silvia Mazzini, Stefano Puri

Intecs, Italy; silvia.mazzini@intecs.it stefano.puri@intecs.it

Abstract

Space systems often need to be engineered in compliance with standards such as ECSS and need to ensure a certain degree of dependability. Given the multi-faceted nature of dependability (characterized by a set of concerns), assuring dependability implies multi-concern assurance, which requires the modelling of various system characteristics and their co-assessment and co-analysis, in order to enable the management of trade-offs between them. CHESS is a systems engineering methodology and an open source toolset, which includes ConcertoFLA. ConcertoFLA allows users (system architects and dependability engineers) to decorate component-based architectural models with dependability-related information, execute Failure Logic Analysis (FLA) techniques, and get the results back-propagated onto the original model. In this paper, we present the customization of the CHESS methodology and ConcertoFLA in the context of the ECSS standards to enable architects and dependability engineers to define a system and perform dependability-centered co-analysis for assuring the required non-functional properties of the system according to ECSS requirements. The proposed customization is then applied in the context of spacecraft Attitude Control Systems engineering, which is a part of satellite on-board software.

Keywords: Dependability analysis, Failure Logic Analysis, Multi-concern, Dependability assurance, ECSS standard series, CHESS toolset.

1 Introduction

Space systems such as satellites are often required to be engineered according to the standards such as European Cooperation for Space Standardization (ECSS) standards. The ECSS standards address different aspects of space project ranging from management, space system engineering and qualification. Due to the critical nature of the space systems, ECSS puts requirements on the assurance of the product and its software systems. In particular, ECSS has standards for software engineering ECSS-E-ST-40C [2], the assurance of dependability of product ECSS-Q-ST-30C [4], safety of product ECSS-Q-ST-40C [5], assurance of software ECSS-Q-ST-80 [3] and assurance of security of software ESSB-ST-E-008 [1]. To fulfil the requirements of the standards and provide assurance of dependability, safety and security, a systematic approach for co-assessment and co-analysis could have advantages on manifold. For example, modelling of various system characteristics and their co-assessment and co-analysis leads to reduction in cost as well enable the management of trade-offs between these properties.

CHESS [13] is a methodology and an open source supporting toolset based upon Papyrus UML [23]. CHESS is the result of several R&D projects, starting from the original CHESS (Composition with Guarantees for High integrity Embedded Software Components Assembly) ARTEMIS JU project [9] and continuing with CONCERTO (Guaranteed Component Assembly with Round Trip Analysis for Energy Efficient High Integrity Multicore Systems) ARTEMIS JU project [9]**,** to provide a model based solution to address the challenges of developing critical real time and embedded systems, by adopting a component based approach, across several domains of interest, including space.

The CHESS Modelling Language (CHESSML), part of the CHESS documentation [9], is based upon UML [22], SysML [19], MARTE [20] and includes also SafeConcert [14] as its base for the dependability profile. This profile enables a support of decorating the component based architectural models with dependability related information. ConcertoFLA [6], which is a part of CHESS toolset, utilizes the decorated components and calculates the failure behaviour of the composed system, representing the assembly of these components. The CHESS design modelling capabilities along with the analysis capabilities are well supportive and compliant with the ECSS standards addressing product and software engineering and assurance.

In this paper, we extend our previous work [21] and we customize the CHESS and ConcertoFLA methodologies in the context of ECSS. The approach, resulting from the customization, enables the co-analysis of reliability, safety and security concerns. Such co-analysis has the potential to contribute in the reduction of cost, complexity and in the management of trade-offs as well as compliance with the standards for qualification purposes.

2 Background

In this section, we describe the background concepts. In particular, Section 2.1 provides the details of ECSS standards. Section 2.2 describes the ConcertoFLA analysis process.

2.1 European Cooperation for Space Standardization (ECSS) standards

ECSS standards cover all the aspects of a space system project spanning to the management of the project, engineering space system and its qualification. Assurance of different properties is an essential part of system engineering. ECSS provides standards for assurance of dependability, safety of the system and the software product as well as security of software. ECSS- E-ST-40C standard is focused on software part of the space system. The standard covers all the phases of the development of the software and puts requirements and principles for software design. For the assurance of software, the standard refers to the ECSS-Q-ST-80C.

Following are the ECSS standards related to the system and the software product assurance, in particular assurance of dependability, safety and security.

- ECSS-Q-ST-30C, defines the dependability requirements on space product assurance. In ECSS scope, the notion of dependability embraces reliability, maintainability and availability. Unlike, the academic dependability literature [12], where dependability also includes safety and security. The standard puts requirements over dependability analysis and states "*dependability analysis shall be conducted on all levels of the space system and be performed in respect of the level that is being assessed i.e., System, Subsystem and Equipment levels*".
- ECSS-Q-ST-40C, defines the requirements on space product assurance focused on Safety. The standard requires that hazard analysis shall be conducted to identify the hazards. Also, it states that "*The fault tree analysis shall be used to establish the systematic link between the system level hazard and the contributing hazardous events and subsystems, equipment or piece part failure*".
- ESSB-ST-E-008, defines the requirements for secure engineering of the space software product. The standard is focused on the security of software product and states that "*The supplier shall perform a cyber-security risk assessment of the software products in order to determine the security sensitivity of the individual software components*".
- ECSS-Q-ST-80C, lists the requirements for software product assurance with emphasise on dependability and safety. The standard state "*The supplier shall perform a software dependability and safety analysis of the software products, in accordance with the requirements of ECSS-Q-ST-30 and ECSS-Q-ST-40 and using the results of system level safety and dependability analysis, in order to determine the criticality of the individual software components*".

2.2 ConcertoFLA

ConcertoFLA is a tool-supported methodology for the compositional calculation of the failure behaviour of component-based systems, based on the failure behaviour of individual components. The failure behaviour is specified using an adaptation [8] in the CHESS context of Failure Propagation Transform Calculus (FPTC) [7] rules. Each FPTC rule defines the input/output behaviour of a specific component using a combination of the port name and the guide-word/failure mode. ConcertoFLA supports three types of failure modes with two specializations for each – the failure modes are value (coarse/subtle), timing (early, late), provision (omission, commission). Using the FPTC rules, four different behaviours of a component can be defined, which are as following:

- Propagator, a component propagates the fault it received on its input port to the output port without changing the type of the fault.
- Transformer, component transform the fault received on its input port into another type of the fault.
- Sink, component sinks the fault it receives on its input port and produces no fault on its output port.
- Source, component is the source of the fault on its output port and received no fault on its input port.

3 ECSS-compliant Multi-concern assurance approach

As recalled in Section 2.1, ECSS standards require the assurance and analysis of several non-functional properties of the system. The CHESS methodology and ConcertoFLA, recalled in Section 1, are customized for performing multi-concern assurance, focusing on three concerns, i.e., safety, security, and reliability. The overall approach, resulting from the customization, consists of five activities, as the activity diagrams, depicted in Figure 1, shows. These activities are:

1. System design- The system architecture is specified using CHESSML. First, all the components in isolation are specified and then assembled.
2. Individual component failure behavior specification using FPTC rules. As stated in Section 2.2, the failure modes used are of high abstraction. The advantage of this abstraction is the support for the assembly of heterogeneous components e.g., developed in different domains with different specialized terminology. In this paper, the above-mentioned abstraction facilitates the interpretation of the failure modes for different concerns.
3. Behaviour injection and ConcertoFLA execution to calculate the failure behavior at system level. The

analysis generates failure propagation paths, which consist of the sequences of the possible events leading to the system level failures, as a consequence of the injected behavior (including fault(s) injection, i.e., failure(s) of preceding systems feeding the system under analysis as well as normal behaviour to potentially detect components acting as sources).

4. Interpretation (conducted manually) of the analysis results for multi-concern e.g., reliability, safety and security concerns. Next, a trade-off is calculated between these properties. Base on the interpretation for multi-concern and trade off, dependability means are introduced by refactoring the system design, if the certain level of dependability is not achieved.

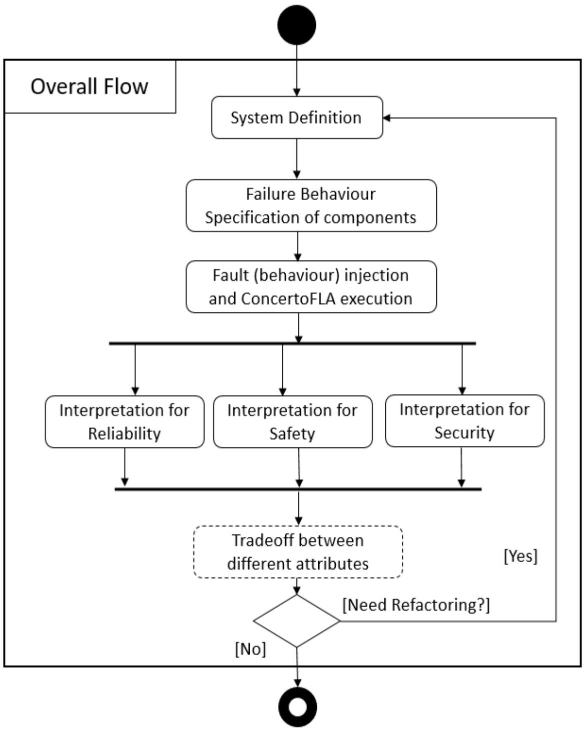
Overall Flow
System Definition
Failure Behaviour Specification of components
Fault (behaviour) injection and ConcertoFLA execution
Interpretation for Reliability
Interpretation for Safety
Interpretation for Security
Tradeoff between different attributes
[Yes]
[Need Refactoring?]
[No]

Figure 1 Multi -concern assurance approach

4 Application of Approach to Attitude Control System Engineering

In this section, first we describe the space system used for illustration purposes, then, we apply our approach to it.

4.1 Attitude Control System (ACS)

The ACS of a satellite is an on-board subsystem that controls the orientation of the satellite, relative to a reference frame, in space. For projects developed for European Space Agency (ESA), an ACS is normally developed according to ECSS standards, therefore its engineering is required to comply with the ECSS standards and a certain level of dependability, safety and security of software is assured. ACS engineering includes activities spanning performance analysis, budgets, procurement and dimensioning of sensors and actuators etc., along with the ACS development. ACS development refers to the development of ACS application software and its associated algorithms. The ACS (application) software takes sensor data containing information about the current state of the satellite and computes the control torque to be applied to the satellite body in order to achieve its target state. To do this, ACS has three functions i.e., process unit data, state estimation and computation of the control torques to minimize the difference between current and target state. ACS has different operational modes, which involves different devices and reflects the mission requirements. For example, in Sun Acquisition and Survival Mode (SASM) it is required to control the orientation of the satellite relative to the Sun to ensure sufficient solar power to the system. The SASM normally takes inputs from sun sensors and a gyroscope to compute a torque that is applied to the satellite body e.g. using propulsion thrusters.

4.1 Application

We apply our approach to the ACS in SASM mode. We limit the scope of functions of ACS to the control function, which maintains the target state in response to the estimated state. The functional requirements of control function in SASM mode are as following.

The sun acquisition control function shall compute and output a control torque based on PD controller, gyroscopic torque compensation and deadband filter in order to point the satellite (its reference direction) at the Sun.

To design the system with above-mentioned requirement, a component based model is defined using CHESS modelling environment. Figure 2 shows the assembly of the following four components implementing the SASM control function requirement.

- PDController, computes the proportional and derivative torque to orient the satellite relative to the Sun.
- SteerController, computes the proportional torque using different gains and control law.
- FeedforwController, compensates for the gyroscopic coupling.
- TorqueSelector, selects the control torque based on the current state of satellite via choosing between two control strategy to enhance the performance and fast convergence to the target orientation.

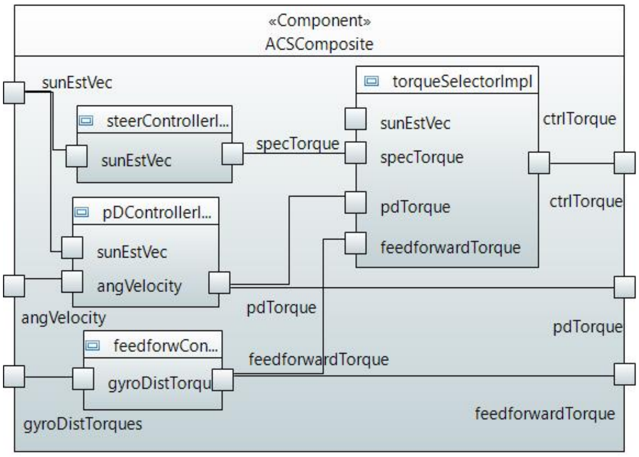

Figure 2 Component based design of ACS

The next step, after system definition, is to model dependability and perform ConcertoFLA analysis. In this regard, we modelled the failure behaviour of components as a propagator and injected the system with the failure of type “value”. It has been assumed that the injected failure, is due to the failure in state estimation unit of satellite and refers to the “state estimator unit provides inaccurate value” failure. Upon execution of ConcertoFLA analysis, the failure propagation paths are generated providing the failure behaviour at system level. To interpret the results for reliability, a fault tree can be constructed manually following the failure propagation paths. The system level failure, which refers to the “ACS computing inaccurate torques” is due to the value failure at “ctrlTorque” output port of ACS system. A partial manually constructed fault tree is depicted in Figure 3. To interpret the results for safety concern, the top event of the fault tree refers to a hazardous event, which is the combination of system level failure and the operational situation e.g., “ACS computing inaccurate torques in SASM mode” leads to a catastrophic consequences. To interpret the results for the security, the top event of fault tree refers to a security threat which is loss of one or more security properties i.e., confidentiality, integrity and availability.

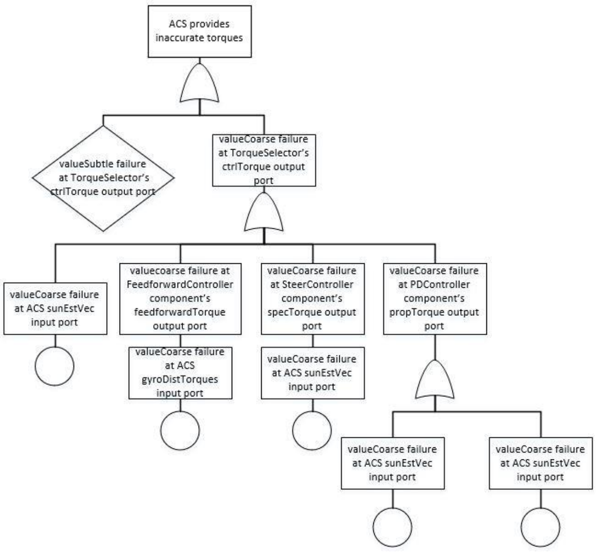
ACS provides inaccurate torques
valueSubtle failure at TorqueSelector's ctrlTorque output port
valueCoarse failure at TorqueSelector's ctrlTorque output port
valueCoarse failure at ACS sunEstVec input port
valuecoarse failure at FeedforwardController component's feedforwardTorque output port
valueCoarse failure at SteerController component's specTorque output port
valueCoarse failure at PDController component's propTorque output port
valueCoarse failure at ACS gyroDistTorques input port
valueCoarse failure at ACS sunEstVec input port
valueCoarse failure at ACS sunEstVec input port
valueCoarse failure at ACS sunEstVec input port

Figure 3 Manually constructed partial fault tree adapted from [18]

5 Conclusion and Future work

In this paper, we presented the customization of the CHESS methodology and ConcertoFLA in the context of the ECSS standards to enable architects and dependability engineers to define a system and perform dependability-centered co-analysis for assuring the required non-functional properties of the system according to ECSS requirements. Then, we applied our customization in the context of the Attitude Control Systems engineering.

From that application it emerged that CHESSML is appropriate to design the ACS in compliance with the requirements of ECSS-E-ST-40C. More precisely, the CHESSML based design complies with Section 5.4.3 of that standard, which is focused on the software architectural design and requires the component based design. The analysis part of CHESSML i.e., ConcertoFLA supported the requirements focused on the assurance of software reliability, safety and security. Moreover, the certifiable evidences could be manually constructed to support the qualification process.

We also observed that the employment of CHESS toolset supports the end to end process, where the functional design, annotated with non-functional properties and assurance support, could shorten the feedback loop for mastering the improved design as well as reduces the complexity.

In the future, we plan to provide tool support for the manual interpretation and construction of evidences for multi concerns. In this regard, our recent work [16] automatically generates the fault tree for reliability from the ConcertoFLA results.

Acknowledgements

This work is supported by the EU and VINNOVA via the E CSEL project AMASS (No 692474) [17] [15].

References

[1] ESSB-ST-E-008 - Secure Software Engineering Standard, 2016

[2] ECSS-E-ST-40C, Space engineering - Software, 06/03/2009.

[3] ECSS-Q-ST-80C, Space product assurance - Software product assurance, 06/03/2009.

[4] ECSS-Q-ST-30C, Space product assurance - Dependability, 06/03/2009.

[5] ECSS-Q-ST-40C, Space product assurance - Safety, 06/03/2009.

[6] Gallina B., Sefer E., Refsdal A. Towards Safety Risk Assessment of Socio-Technical Systems via Failure Logic Analysis. IEEE International Symposium on Software Reliability Engineering Workshops, Naples, pp. 287-292. 2014.

[7] Wallace M. Modular architectural representation and analysis of fault propagation and transformation. Electronic Notes in Theoretical Computer Science, volume 141 n.3, pp. 53-71, December, 2005.

[8] Gallina B., Javed M.A., UL Muram F., Punnekkat S. A. Model-Driven Dependability Analysis Method for Component-Based Architectures. 38th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) Cesme, Izmir, pp. 233-240. 2012.

[9] CHESSML https://www.polarsys.org/chess/start.html

[10] ARTEMIS-JU-100022 CHESS- Composition with guarantees for High integrity Embedded Software components assembly. http://www.chess-project.org

[11] ARTEMIS-JU CONCERTO - Guaranteed Component Assembly with Round Trip Analysis for Energy Efficient High-integrity Multi-core systems. http://www.concerto-project.org

[12] A. Avizienis, J.C. Laprie, B. Randell, C. Landwehr. Basic concepts and taxonomy of dependable and secure computing. In: IEEE Trans. Dependable Sec. Comput. 1(1): 11-33, 2004.

[13] Mazzini S., Favaro J., Puri S., Baracchi L. CHESS: an open source methodology and toolset for the development of critical systems. Third Workshop on Open Source Software for Model Driven Engineering. OSS4MDE. 2016.

[14] Montecchi L. and Gallina B. SafeConcert: a Metamodel for a Concerted Safety Modeling of Socio-Technical Systems. 5th International Symposium on Model-Based Safety and Assessment (IMBSA), Trento, Italy, September, 2017.

[15] Ruiz A., Gallina B., de la Vara J.L., Mazzini S., Espinoza H. Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems. Computer Safety, Reliability, and Security. SAFECOMP. LNCS, vol 9923. Springer. 2016.

[16] Haider Z., Gallina B. and Zornoza. E. M. “FLA2FT: Automatic generation of fault tree from ConcertoFLA results” 3rd International Conference on System Reliability and Safety (ICSRS), Barcelona, 2018.

[17] AMASS, http://www.amass- ecsel.eu

[18] Gallina B., Haider Z. and Carlsson A. Towards generating ECSScompliant fault tree analysis results via ConcertoFLA, IOP Conference Series: Materials Science and Engineering, 2018.

[19] SysML v1.4 Specification Release September 2015 http://www.omgsysml.org/specifications.htm

[20] MARTE www.omg.org/spec/MARTE/About-MARTE/

[21] Gallina B., Haider Z., Carlsson A., Mazzini S., Puri S. Multi-concern Dependability-centered Assurance for Space Systems via ConcertoFLA. 23rd International Conference on Reliable Software Technologies-Industrial Presentation Track (Ada-Europe), Lisbon, Portugal, June 18-22, 2018.

[22] UML, www.omg.org/spec/UML/2.5.1/

[23] Papyrus, www.eclipse.org/papyrus/