COMPASTA

Deliverable ESR

Executive Summary Report

Written by	***Company***	***Signature***
Marco Bozzano	FBK	
Reviewed by	***Company***	***Signature***

FONDAZIONE
BRUNO KESSLER

Table of Revisions

Issue	Date	Change Record	Author
1.0	07/12/2022	First release	Marco Bozzano
1.1	10/12/2022	Edited to address comments by M.Verhoef	Marco Bozzano

Table of Contents

Glossary

AADL	Architecture Analysis and Design Language
ASN.1	Abstract Syntax Notation One
Catsy	Catalogue of System and Software Properties (ESA Study)
COMPASS	Correctness, Modeling and Performance of Aerospace Systems (ESA Study)
COMPASS3	Consolidation of COMPASS Tools (ESA Study)
CSSP	Catalogue of System and Software Properties
FDIR	Fault Detection, Isolation and Recovery
FMEA	Failure Modes and Effects Analysis
FTA	Fault Tree Analysis
HW	Hardware
Kratos	The Kratos Software Model Checker
MBD	Model Based Design
NuSMV	New Symbolic Model Verifier
nuXmv	New Extended Model Verifier
ocra	Othello Contracts Refinement Analysis
SDE	Syntax Directed Editor
SDL	Specification and Description Language
SLIM	System-Level Integrated Modeling Language
SMV	Symbolic Model Verifier (the language of the NuSMV tool)
SW	Software
TASTE	The Assert Set of Tools for Engineering
VM	Virtual Machine
xSAP	Extended Safety Analysis Platform

1. Introduction

This document contains the executive summary report of the COMPASTA project. It discusses the context of the project, the activities performed, the main achievements, and future work.

1.1 Purpose

The Executive Summary Report is mandatory deliverable, due at the end of the contract.

1.2 Objective

This document is a FBK delivery in the frame of the contract No. 4000133700/21/NL/GLC/kk with ESA “Integration of the COMPASS and TASTE Toolsets”.

2. Applicable and Reference Documents

2.1 Applicable Documents

AD1	ESA OSIP call for Ideas “Model-Based System Engineering: from documents to models”. https://ideas.esa.int
AD2	FBK Proposal to OSIP call for Ideas “Model-Based System Engineering: from documents to models”. ESA Express Procurement Plus – EXPRO+, Proposal No. AO/2-1754/20/NL/GLC.
AD3	COMPASTA Deliverable D1. D1.1: Integrated workflow. D1.2: Use cases and requirements.
AD4	COMPASTA Deliverable D3.1: Languages definition: syntax and semantics.
AD5	COMPASTA Deliverable D3.4: Extension with trace validation for testing.
AD6	COMPASTA Deliverable D4.1: Compass Software ported to python3 and GTK3.
AD7	COMPASTA Deliverable D5.2: Case studies and documentation.
AD8	COMPASTA Deliverable D5.3: Validation Report (final version).
AD9	COMPASTA Deliverable D6.1: TASTE website.
AD10	COMPASTA Deliverable D6.2: Roadmap.

2.2 Reference Documents

RD1	Final Report, ESA Contract No. 21171/07/NL/JD - “System-Software Co-Engineering: Performance and Verification”, 2007.
RD2	Final Report ESA Contract No. 4000111828 - “Catalog of System and Software Properties”, 2013.
RD3	Final Report ESA Contract No. 4000115870/15/NL/FE/as - “Consolidation of COMPASS Tools”, 2016.
RD4	COMPASS Website. www.compass-toolset.org/
RD5	TASTE Website. https://taste.tools/
RD6	OpenGEODE website. https://taste.tuxfamily.org/wiki/index.php?title=Technical_topic:_OpenGEODE,_an_SDL_editor_for_TASTE
RD7	nuXmv website. https://nuxmv.fbk.eu/
RD8	xSAP website. https://xsap.fbk.eu/
RD9	ocra website. https://ocra.fbk.eu/
RD10	Kratos website. https://es-static.fbk.eu/tools/kratos/
RD11	M. Bozzano, A. Cimatti, C. Mattarei and S. Tonetta. Formal Safety Assessment via Contract-Based Design. In Proceedings of ATVA 2014, LNCS 8837, pages 81-97. Sydney, Australia, November 3-7, 2014.
RD12	M. Bozzano, A. Cimatti, M. Gario, D. Jones, C. Mattarei. Model-based Safety Assessment of a Triple Modular Generator with xSAP. In Formal Aspects of Computing 33(2):251 295, 2021.
RD13	The TASTE development team. TASTE wiki. Available at: https://taste.tuxfamily.org/wiki/index.php?title=Main_Page
RD14	The TASTE development team. SDL Tutorial. Available at: https://taste.tuxfamily.org/wiki/index.php?title=Detailed_SDL_tutorial
RD15	International Telecommunication Union. ITU-T Z.100. Specification and Description Language – Overview of SDL-2010. Available at: https://www.itu.int/rec/T-REC-Z.100
RD16	The COMPASS Consortium. SLIM 3.0 - Syntax and Semantics. 2019.
RD17	Pierre Dissaux. TASTE IV and DV editors. Meta-model. ESA/ESTEC frame contract n°4000104809 – 29 November 2011. Call-Off Order 008 – 19 November 2018. Issue: release 0.3 – 07 March 2019.

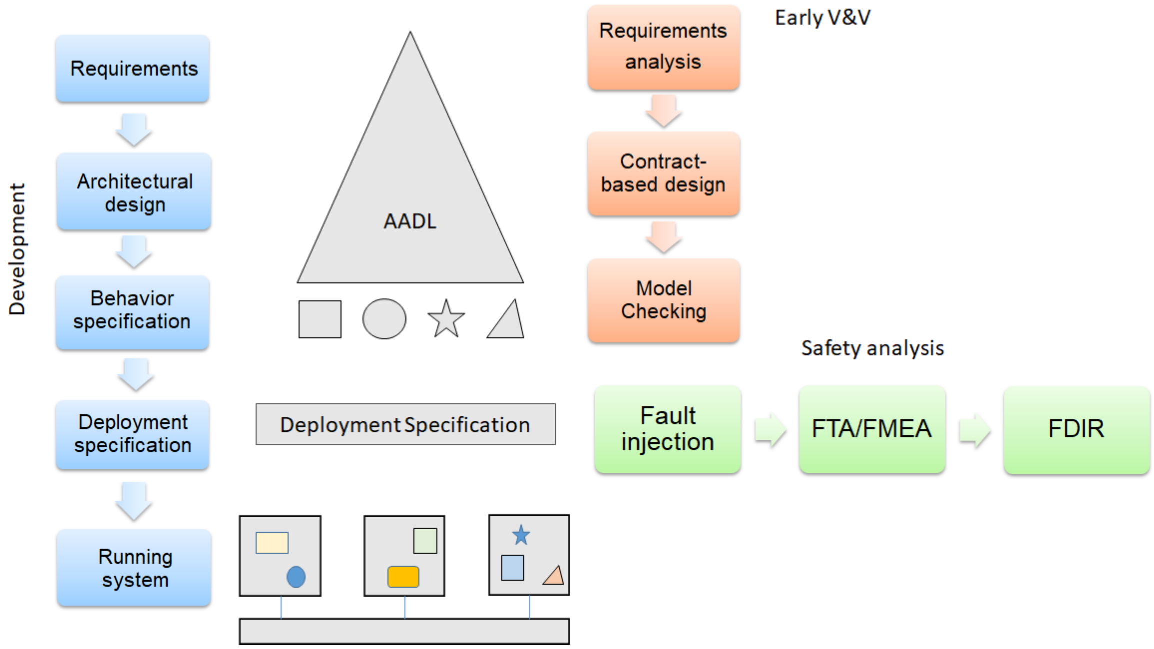
RD18	Architecture Analysis & Design Language (AADL) . SAE document AS5506D. Architecture Analysis & Design Language (AADL) AS5506D.
RD19	Cavada, R., Cimatti, A., Crema, L., Roccabruna, M., Tonetta, S. (2016). Model-Based Design of an Energy-System Embedded Controller Using Taste . In: Fitzgerald, J., Heitmeyer, C., Gnesi, S., Philippou, A. (eds) FM 2016: Formal Methods. FM 2016. Lecture Notes in Computer Science(), vol 9995. Springer, Cham.

3. Study Description

COMPASS is a tool for model-based system/SW co-engineering developed in a series of ESA studies (2008-2016). It is based on a dialect of AADL [RD18] and it offers a complete set of functionalities for formal verification, including requirements specification and analysis, contract-based design and verification, functional verification, fault specification, fault injection and RAMS analyses, including Fault Tree Analysis (FTA), Failure Modes and Effects Analysis (FMEA), Diagnosability Analysis, Fault Detection, Isolation and Recovery Analysis (FDIR). COMPASS is based on the AADL built-in concept of model extension, i.e., the possibility to automatically inject faults into a nominal model, by specifying error models and a set of fault injections.

TASTE is a development environment dedicated to embedded, real- time systems, developed since 2008 under the initiative of the European Space Agency, together with a set of partners from the space industry. It consists of various tools such as graphical editors for models, visualizers, code generators and middlewares that support the development of embedded systems within a model-based design (MBD) approach. The key technologies involved are AADL for architecture definition, ASN.1 for data modelling and SDL [RD15] for behavior specification.

The objective of COMPASTA is to integrate the COMPASS functionality into TASTE, thus providing a comprehensive, end-to-end tool chain that covers system development, early verification and validation, safety assessment and FDIR, and system deployment.



4. Main Achievements

COMPASTA integrates the COMPASS functionality into TASTE, thus making a first step towards an integrated and coherent tool chain, filling the gap between the architectural level design and the system implementation and deployment, harmonizing the functionalities for system design and system implementation.

Technically, the main contributions of COMPASTA are:

- A lightweight integration of the COMPASS functionalities into the TASTE GUI, and the implementation of a Command Line Interface (CLI) that implements a set of scripts to run all the functionalities of the toolset from the command line.
- Harmonization of the models and input languages provided by COMPASS (SLIM, a dialect of AADL) with the ones available in TASTE (in particular, AADL and SDL) for the specification of system architecture, component behavior and interaction and system implementation.
- Definition of syntax and semantics for modeling languages. The semantics of the AADL extension has been defined as an adaptation of the semantics of COMPASS. The semantics of communication has been adapted to match the TASTE semantics.
- Encodings and translators from the AADL and SDL input languages into the language supported by the verification engines (SMV), based on the semantics.
- Implementation of the fault injection and model-extension functionality from COMPASS.
- Porting of the verification and validation functionalities from COMPASS.

- Implementation of a prototypical trace validation functionality, which enables the re-execution and validation of a trace generated by TASTE testing in the formal model.
- Demonstration of the approach on a sample case study.

5. The COMPASTA Workflow

In summary, the COMPASTA workflow can be described as follows:

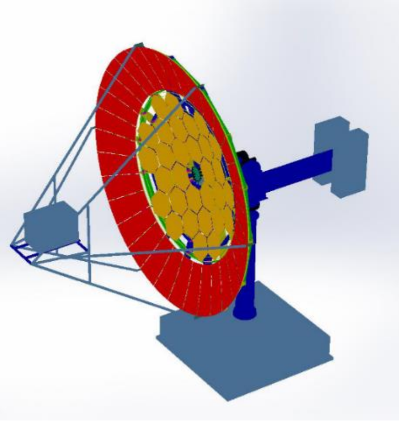
- COMPASS is used to build and validate a formal model of the system (including both HW and SW) architecture, to specify the nominal behavior of the HW components and their faults.
- TASTE is used to model the behavior of the SW components, for deployment and code generation, and to test the final implementation.

Specifications in COMPASTA are based on the AADL and SDL as specification languages AADL is used for the interface view and for the specification of the behavior of the HW components. SDL is used for the specification of the behavior of the SW components.

The COMPASTA workflow consists of the following steps:

- Modeling the System Architecture using the TASTE graphical editor.
- Modeling the behavior of HW components in the SLIM language.
- Modeling the behavior of the SW components in SDL.
- Modeling faults and fault injections.
- Specification of properties and contracts.
- Formal verification, e.g., running functional verification and RAMS analyses.
- Modeling iterations, to adapt the model, depending on the outcome of the analyses.
- Compilation-ready model transformation, to turn HW components into the corresponding SW-HW interface components.
- Deployment and code generation, using TASTE.
- Testing and simulation, using TASTE.

6. Evaluation



The COMPASTA toolset has been evaluated on the CONTEST case study [RD19], a case study from the energy domain. The case study concerns the co-generation of electric and thermal energy, using a solar concentration made up by a dish, plus a stirling engine. The system can be subject to faults (e.g., engine errors, dish over-heating) and must satisfy several functional and safety requirements (e.g., automatic shutdown in case of unrecoverable errors).

Modeling and analysis have been focused on the dish and the dish controller components.

The dish exposes a high-level interface, namely the TASTE system communicates to the dish controller with a specific protocol composed by high level commands. Examples of analyses include the adherence to communication protocols, the response to protocol violations, and response to error conditions from the dish.

The outcome of the analyses revealed that the model checker is unable to deal with the most complex models, and that further work is needed, in order to make the COMPASTA approach practicable for realistic models. The main reasons for the limited scalability of the verification engines are likely related with the large number of blocks, data and connections, that the model checker needs to deal with, and the underlying semantical model based on interleaving semantics. Possible solutions to these shortcomings, to be address these issues as future work, are re-thinking the semantical model, enhance the model checker by embedding knowledge about the semantical model into it, and use compositional reasoning for verification and validation.

7. Future Work

The future developments and extensions of COMPASTA include improving the usability of the toolset, its technology readiness level, its functionality, and in particular improving its performance and scalability. Further work is needed to improve the integration with TASTE and the overall workflow, demonstrate the applicability of COMPASTA for designing (industrial) systems of realistic size, improve the visibility of the toolset, market penetration, commercial exploitation, and industrial usage, and integrate COMPASTA with other tools and design environments, such as Cameo, Chess, and Capella.

END OF DOCUMENT