(ITEA 3 – 17003)

PANORAMA

Boosting Design Efficiency for Heterogeneous^3 Systems

Deliverable: D6.7

Integrated Safety Analysis Techniques

Work Package: 6

Design Flow and Traceability

Task: T6.5

Integrated Safety Analysis

Document Type:	Software	**Classification:**	Public
Document Version:	Final	**Contract Start Date:**	2019-04-01
Document Preparation Date:	2022-03-31	**Duration:**	2022-03-31

ITEA 3

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

EUREKA

Contents

List of Figures

List of Tables

1 Introduction

This document describes the application of the model checking tool RTANA2_{sim} for performing timing analyses in the context of the Workpackage 6 "Design Flow and Traceability". The tool is applied in two design contexts: (1) For *Virtual Integration Testing* in order to verify correct decomposition of timing requirements along the system decomposition at the system design level, and (2) for *Satisfaction Checking* in order to verify that the components at the HW/SW design level satisfy their timing contracts. The document informs about the input artefacts required to perform the analysis as well as the produced output artefacts.

2 Timing Analyses with $\text{RT}\textsc{ana}_{2sim}$

As reported previously in Deliverable D3.1 [PAN19], OFFIS provides timing analyses for AMALTHEA models that internally use the model checker $\text{RT}\textsc{ana}_{2sim}$ also developed by OFFIS. The analysis methods support a contract-based design methodology where timing specifications are expressed in a textual yet formal language [KKS+22; BKS21].

2.1 Timing Analysis Methods

OFFIS provides two types of timing analysis for AMALTHEA models:

1. A *virtual integration test* checks whether some component is correctly decomposed into subcomponents with respect to timing. The analysis model simulates the guaranteed timing behavior of the sub-components as well as the behavior assumed from the environment, and monitors the timing behavior of the composition.

2. The *timing satisfaction check* is used to verify that the atomic components at the bottom of a hierarchical design satisfy their timing contracts. The analysis input is an AMALTHEA model of the software tasks that implement the component and their allocation to CPU cores. It is translated into an $\text{RT}\textsc{ana}_{2sim}$ model that simulates the software and monitors if the component's guarantees are satisfied. Furthermore, the AMALTHEA model can be annotated with information on implemented safety mechanisms. This allows to specify and verify timing of fault mitigation under consideration of a failure model.

The $\text{RT}\textsc{ana}_{2sim}$ model checker tries to perform a complete state space exploration, which allows to formally verify timing and safety properties [SRGB13]. Also, the model checker can apply some heuristics that often allow to detect contract violations even if the model is too complex for analyzing the complete state space. More details on the methodology can be found in the PANORAMA WP3 deliverables. [PAN19; PAN22].

2.2 Inputs and Outputs

The analyses consume information from AMALTHEA, ODE [PSR19] and UML/Papyrus[1] models.

- AMALTHEA is used for representing HW/SW architectures, i.e. software elements (tasks, runnables, ...) and the mapping to hardware

[1] `https://www.eclipse.org/papyrus/`

Table 2.1: Traceability Information

Tracelink	Source		Target	
	Type	Model	Type	Model
SpecifiedByContract	Component	UML/Papyrus	DependabilityContract	ODE
ImplementsComponent	Task, Runnable	AMALTHEA	Component	UML/Papyrus

- ODE is used for storing timing contracts
- An UML/Papyrus component model is used for the functional architecture (components, ports, safety mechanisms, etc)

The ODE, Papyrus, and AMALTHEA models are linked with each other using Eclipse CAPRA[2] [MS16]. The CAPRA trace link types have been aligned with the traceability information model (TIM) from the MobSTr data set [SKB+21] wherein the analyses have been used. The used traceability information is listed in Table 2.1.

SpecifiedByContract This trace link maps UML/Papyrus Components to their specification in form of contracts in an ODE model.

ImplementsComponent This trace link forms the main connection between AMALTHEA and UML/Papyrus component models: It connects tasks and runnables to the components they implement

The UML/Papyrus component models are used as an entry point for the analysis, i.e the user selects the component to be analyzed, and RTANA$_{2sim}$ automatically traverses incident traceability links to find the timing contracts and the HW/SW model. The prototype has been designed as a plugin for the APP4MC IDE with a simple integration into the user interface.

As an output, the analysis produces a simple HTML report, as well as the intermediate RTANA$_{2sim}$ model, and a BTF trace. The RTANA$_{2sim}$ model is processed by the model checker backend to derive the verification result and can be used to debug the results in depth with the native user interface of RTANA$_{2sim}$. The BTF trace represents an example run of the modeled system supporting the anaylsis result. More information about this can be found in [PAN22]. Additionally, the tool allows to export the analysis configuration and results in a structured data format that allows to replay the analysis and, e.g., can be linked to a safety case for documenting the verification plan.

2.3 Software Release and Accessibility Information

Both the analysis plugins and the RTANA$_{2sim}$ model checker are research prototypes that are not intended for production use. The software is not publicly available. Access to the

[2] `https://projects.eclipse.org/projects/modeling.capra`

software (including documentation) will be negotiated on a bilateral basis upon request. For more information contact Jan Steffen Becker (`jan.becker@dlr.de`).

Bibliography

[BKS21] J. S. Becker, B. Koopmann, and I. Stierand, "Safety relevant time intervals for MobSTr," Tech. Rep., 2021, https://github.com/panorama-research/mobstr-dataset/blob/ab7cbaf05935810880d39fdb9d93d6e7312cdbf1/org.panorama-research.mobstr.requirements/timing.pdf.

[KKS+22] J. Kröger, B. Koopmann, I. Stierand, N. Tabassam, and M. Fränzle, "Handling of operating modes in contract-based timing specifications," in *15th International Conference on Verification and Evaluation of Computer and Communication Systems (VECoS)*, to appear. Preprint avaliable from https://www.researchgate.net/publication/355118482_Handling_of_Operating_Modes_in_Contract-based_Timing_Specifications/, 2022.

[MS16] S. Maro and J.-P. Steghöfer, "Capra: A configurable and extendable traceability management tool," in *2016 IEEE 24th International Requirements Engineering Conference (RE)*, IEEE, 2016, pp. 407–408.

[PAN19] PANORAMA Consortium, *PANORAMA Deliverable D3.1: Description of state of the art of analysis methods*, 2019.

[PAN22] PANORAMA Consortium, *PANORAMA Deliverable D3.4 & D3.5: Requirements and implementation for dynamic analysis methods of modelling formalisms of wp1*, 2022.

[PSR19] Y. Papadopoulos, I. Sorokos, and J. Reich, "ODE Profile V2," DEIS project, Whitepaper, 2019, http://www.deis-project.eu/fileadmin/user_upload/DEIS_Open_Dependability_Exchange_Profile_V2_Whitepaper.pdf, last accessed May 25, 2020.

[SKB+21] J.-P. Steghöfer, B. Koopmann, J. S. Becker, I. Stierand, M. Zeller, M. Bonner, D. Schmelter, and S. Maro, "The mobstr dataset - an exemplar for traceability and model-based safety assessment," in *29th IEEE International Requirements Engineering Conference, RE 2021, Notre Dame, IN, USA, September 20-24, 2021*, IEEE, 2021, pp. 444–445. DOI: 10.1109/RE51729.2021.00062. [Online]. Available: https://doi.org/10.1109/RE51729.2021.00062.

[SRGB13] I. Stierand, P. Reinkemeier, T. Gezgin, and P. Bhaduri, "Real-time scheduling interfaces and contracts for the design of distributed embedded systems," in *2013 8th IEEE International Symposium on Industrial Embedded Systems (SIES)*, IEEE, 2013, pp. 130–139.