

Modelling V&V Workflows to Improve Development Processes of Software-intensive Systems

Thomas Bauer
Fraunhofer IESE
Kaiserslautern, Germany
thomas.bauer@iese.fraunhofer.de

Wolfgang Herzner
AIT
Vienna, Austria
wolfgang.herzner@ait.ac.at

Bob Hruska
LieberLieber
Vienna, Austria
bob.hruska@lieberlieber.com

Katia di Blasio
Intecs Solutions Spa
Pisa, Italy
katia.diblasio@intecs.it

Zain Shahwar
PUMACY TECHNOLOGIES AG
Berlin, Germany
zain.shahwar@pumacy.de

Abstract

The rapid increase in complexity of systems, especially due to the integration of - increasingly automated - subsystems from different domains into cyber-physical systems, presents unique challenges for efficient verification and validation (V&V) processes to meet all requirements and system properties. To address these challenges and enhance quality assurance processes, it is essential to document and analyze V&V workflows. This paper proposes VVML, a novel approach for easy modeling V&V activities in different industrial domains with varying constraints, V&V methods, and tool chains. This approach includes a dedicated modeling notation and a supporting modelling tool. It includes a dedicated modelling notation and tool support, enabling the creation of reusable workflows. The solution enables the creation of reusable workflow assets, such as V&V activities and artifacts, which can be shared across workflows. The VVML approach has been applied to 10+ industrial use cases. This paper explains the basic principles behind VVML and shows an example of its application to an industrial use case.

Keywords—*process modelling; verification and validation workflows, test processes, domain-specific languages*

I. INTRODUCTION

Analytical quality assurance activities, including verification and validation (V&V) at different levels, have become essential in ensuring high product quality for complex software-intensive systems. The efficient conduction of quality assurance activities requires their systematic description and modelling, including their sub-activities, execution steps, and work products that they process and produce.

A *process workflow* is an orchestrated and repeatable pattern of activities, enabled by the systematic organization of resources into processes that provide services or process information. It consists of ordered sets of operations that process and deliver sets of data and work products in order to achieve defined goals [1]. The effectiveness of modeling techniques in engineering processes is greatly influenced by their comprehensibility, adaptability to the specific context, and the level of adequate support by tools.

In generic modelling languages, such as UML [2], it is possible to model and solve a problem with different solutions. While the flexibility that the language offers is a positive aspect, it also brings problems in communicating ideas effectively. By creating a *domain-specific language* (DSL) with a small number of concise modelling elements and rules, everyone follows a common standardized language

This paper describes a modelling approach for V&V workflows, which has been developed as part of the European research project VALU3S, which aims at the design and implementation of V&V methods and tool chains to assure safety- and security-related requirements of complex technical software-intensive systems [3]. The approach has been applied to 30+ V&V methods and 10+ industrial used cases in VALU3S. Results of applying VVML to the use case *Human-Robot-Interaction in Semi-Automatic Assembly Processes (HRI)* from the production domain are described in the last part of this paper.

II. INDUSTRIAL USE CASE FOR EVALUATION

The HRI use case takes place on the shop floor of a manufacturing-like lab environment, comprising the execution of assembly tasks by human workers focusing on the assembly of transformer units, which consist of multiple parts. HRI systems have to manage the coordination between humans and robots according to the requirements for collaborative industrial robot systems.

The goal of the HRI use case was to perform automated validation of the fault-tolerant design concept of the production facility at the architecture level. The actual validation object was the virtual model of the distributed production facility, which contains dedicated virtual models for the production line parts, sensors, and communication networks that were created and integrated into a holistic simulation scenario.

In the HRI use case, the following V&V methods and tools have been applied and integrate: (1) *virtual validation* for enabling the co-simulation of the distributed design models and fault injection tests with the FERAL tool and (2) *failure detection diagnosis* with data analytics and machine learning components.

III. V&V WORKFLOW MODELLING LANGUAGE

V&V workflows modelling requires a tailored modelling approach for activities and behavior models that consider various user requirements [4]. V&V activities are conducted in several stages of the development process. These activities consume and process dedicated input information such as system requirements, design models, fault types, and failure history and produce test-related work products such as test cases, test harnesses, test results, and reports by a sequence of dedicated steps.

For the solution, we have created a V&V-related profile for the generic, widely applied modelling language UML. The so-called *VVML (Verification and Validation Modelling Language) profile* provides a set of model constructs and deploys the UML profile as a modelling framework, enabling rapid modelling of V&V workflows.

Two levels of modelling are considered in VVML: (1) *V&V Method Definition*, which serves for defining base elements and global properties of V&V methods applied in a project, and (2) the *V&V Workflow Specification*, which contains the actual workflow implementation within a V&V method, organizes and specifies the composition of activities to reflect their sequential dependencies and the internal flow of artifacts while executing the V&V method. Sections V and VI provide more details for both modelling levels.

IV. TOOL IMPLEMENTATION

VVML is implemented as a plugin for the UML modeling framework *Enterprise Architect (EA)* [5] by Sparx Systems. In EA, new modeling languages can be created with UML-Profiles, which can be used directly afterwards or can be packaged into a *model-driven generation technology (MDG)* for a more comfortable use. MDG Technologies plug seamlessly into EA to provide additional toolboxes, diagrams, UML profiles, shape scripts, patterns, and tagged values. Such an MDG technology automatically generates a list of elements and relationships in the diagram toolbox. Therefore, EA has been extended used to implement VVML and provide a simple user-friendly interface for modeling V&V workflows with specially customized diagram types, enabling workflow modeling with V&V methods, activities, control flows, and flows of work products.

V. V&V METHOD DEFINITION

The *V&V method definition* specifies the base elements of the V&V workflows, i.e., V&V methods and the artifacts (i.e., the work product) these methods process and produce. The *Method* modelling element is a unit that represents a process workflow dedicated to a specific V&V phase. It has a defined *method type*, which is used to represent the automation level. An *Artifact* is an object that is exchanged between methods and its environment (or activities within methods, see next clause). The method has a dedicated type and represents either an information object or an active unit, i.e., program code or executable. Every method owns a set of *MethodArtifacts*, which represent the method interfaces for the artifacts that they consume or produce.

Methods can be composed by sub-methods, which also have dedicated interfaces with method artifacts. In the use case we show as an example in this paper, the overall use case workflow,

called *Combined Virtual Validation and Failure Detection Diagnosis*, references and invokes the V&V methods *Virtual Validation* and *Failure Detection Diagnosis*, which are modelled as separate V&V Method Definitions.

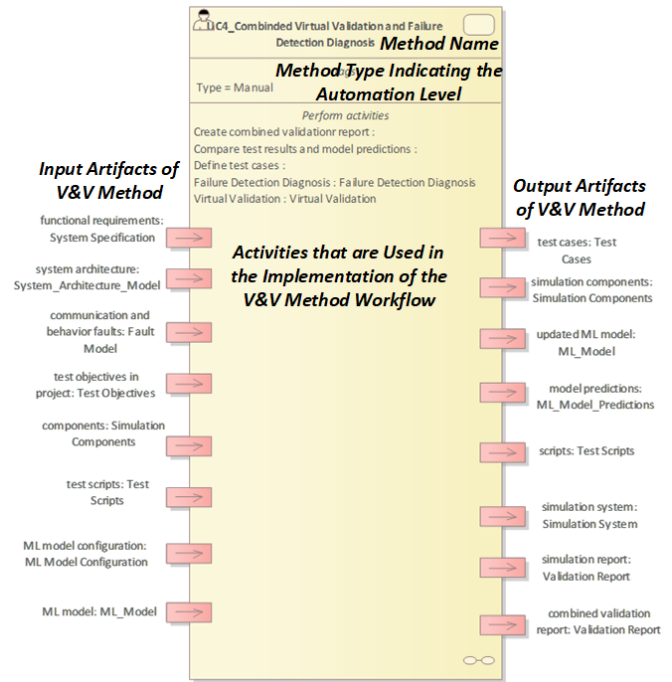


Fig. 1. Method Definition Diagram of the example V&V Workflow

Fig. 1 shows the method definition of the overall V&V workflow of the HRI use case, represented as a rounded rectangle. In the diagram, the required input artifacts are located on the left side of the method (such as the *Fault Model*); the produced output artifacts are located on right side (such as the *Validation Report*). Additionally, the containing activities of the V&V method are listed as method properties. Activities can be atomic, i.e. not refined such as *define test cases*, or decomposed, i.e. referencing a sub-workflow such as *Virtual Validation*.




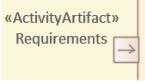

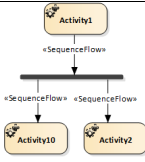
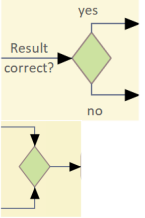
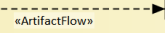
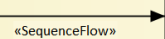
VI. V&V WORKFLOW SPECIFICATION

The *V&V workflow specification* defines Control Flows and Artifact Flows of a V&V method. A *Control Flow* is defined by sequences of V&V activities that are executed in a defined order. Branches in the Control Flow are supported by *Gateways*. Quasi parallel execution is realized by *Fork* and *Join* Elements. *Start* and *End Nodes* indicate beginning and ending of a workflow. Activities can exchange Artifacts through their interfaces, which define the *Artifact Flow* of the V&V workflow. The internal Artifact Flow is defined between activities, whereas the external Artifact Flow is defined from the method interface to the activities for method inputs or from the activities to the method interface for method outputs. A complete list of elements for V&V workflow specifications is provided in TABLE I.

The overall workflow of the HRI use case was designed and developed to detect failures in the early simulation runs. Virtual validation with co-simulating different virtual components and simulation tools is applied to check robustness and fault tolerance of the system architecture design. Furthermore, a dedicated machine learning component is used as enhancement

and improvement of the validation results. The specifications of the integrated V&V workflow of the HRI use case and the referenced workflow of the V&V method Virtual Validation are shown in Fig. 2.

TABLE I. ELEMENTS FOR V&V WORKFLOW SPECIFICATIONS

Element	Notation	Description
Activity		Basic (atomic) step within a Workflow (WF). Owned by a «Method» element. Can be atomic or decomposed (reference to other WF)
StartWorkflow		Initiates a WF
StopWorkflow		Indicates the termination of a WF Upon reaching, all execution in the WF diagram is aborted.
ActivityArtifact	 	Structural element representing an exposed «Artifact» on an «Activity» element. Direction flow is indicated (input-on the left or output-on the right)
Fork/Join		Split incoming flow into multiple concurrent sequence flows Merge concurrent seq. flows. Synchronize concurrent seq. flows at a certain point in the WF
Gateway		Creating (on the left) or joining (on the right) alternative control flows. Carries condition label for alternative flows Sequence flow continues along the outgoing branch with the corresponding guard.
Artifact Flow		Connects two VVML Activities using the «ActivityArtifact» elements (from output to input). Indicating a specific Artifact passing through it.
Sequence Flow		Modeling an active transition between activities. Bridges the flow between activities by directing the flow to the target diagram element once the source node action is completed.

Like V&V methods, activities have input and output interfaces, which indicate the artifacts that they process or produce. Activities consume and process artifacts from the environment (external) or from other activities (internal). Furthermore, activities produce and provide artifacts as result of their execution to other activities for further processing

(internal) or to the environment (external). The sequence of activities is shown by the sequence flow (solid arrows). The flow of artifacts between environment and activities is shown as dashed arrows.

Virtual validation, as a referenced sub-method in Fig. 2, uses a fully virtual setup, i.e., virtual components and a virtual environment, to check specific properties of the system under test through simulation. A virtual simulation environment and dedicated test cases are constructed and used to run and evaluate validation scenarios. The simulation scenarios comprise a set of simulation components, which are connected by dedicated adapters. Missing or incomplete components and adapters are created or extended in separate steps. Simulation components and adapters are deployed to defined execution nodes and connected to the FERAL framework [6]. FERAL executes the simulation scenarios and controls the simulation components and the data flow between them. Log data from simulation runs is collected and provided. A validation report is created after the execution and evaluation of all simulation scenarios. The outputs of the method virtual validation are processed and further refined by the invoking workflow

VII. RULES

Modelling approaches require rules and guidelines to facilitate their usage and prevent incorrect or inappropriate modelling. For VVML, a set of rules have been defined, which are checked either manually by the modelling team or automatically by the EA plugin. A document with guidelines and rules is being prepared. For example, in VVML a workflow is assessed as correct if:

- it never blocks before reaching the stopping node
- it never reaches the stopping node while some activity is still running
- it can always reach the stopping node;
- it never re-enters a running activity; and
- is able to start all of its activities.

Further guidelines address graphical layout and naming, to foster consistent and easy to read appearance of VVML workflow diagrams.

VIII. CONCLUSION AND OUTLOOK

In this paper, VVML – *the Verification and Validation Modelling Language* was introduced, a DSL for compact and efficient modelling of V&V methods and workflows. Due to its small set of modelling elements, it has a low learning curve and is therefore easy to use also for V&V-experts and stakeholders not familiar with rich modelling languages such as UML. This makes it especially appropriate for systematic modelling, describing, and exchanging V&V processes to assure safety- and security-related requirements of complex technical software-intensive, increasingly integrated, and automated systems. By allowing referencing of (already defined) methods as activities in new workflows, VVML supports re-use by decomposition.

VVML was developed with a focus on modelling V&V processes for complex, software-intensive systems in the domains addressed by the VALU3S project, i.e. automotive, rail, industrial automation, and medical equipment. It is, however,

applicable to other domains and areas. Although the basic motivation for VVML is to keep it as simple as possible, future extensions are possible when they turn out as valuable. VVML was implemented as UML-profile for the modelling tool Enterprise Architect.

ACKNOWLEDGMENTS

We would like to thank all VALU3S partners for their contributions and feedback. The research leading to this paper received funding from the ECSEL Joint Undertaking (JU) under grant agreement no. 876852. The JU receives support from the European Union's Horizon 2020 research and innovation program and from the governments of Austria, the Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, and Turkey. The views expressed in this document are the sole responsibility of the authors and do not necessarily reflect the views or position of the European Commission.

References

- [1] Georgakopoulos, D., Hornick, and M., Sheth, A., "An overview of workflow management: From process modeling to workflow automation infrastructure", *Distributed and parallel Databases*, 3(2), 119-153, 1995
- [2] OMG Unified Modeling Language (OMG UML) v.2.5.1, <https://www.omg.org/spec/UML/2.5.1/PDF> [Accessed 6 Apr 2023]
- [3] Website of VALU3S project, <https://valu3s.eu/> [Accessed 6 Apr 2023]
- [4] Bauer, T. et al., "Cross-domain modelling of verification and validation workflows in the large-scale European research project VALU3S", *21st Int. Conf. Embedded Computer Systems: Architectures, Modeling, and Sim. (SAMOS 2021)*, pp. 366-382
- [5] Website of Enterprise Architect by Sparx Systems, <https://www.sparxsystems.eu/newedition> [Accessed 6 Apr 2023]
- [6] T. Kuhn, P. O. Antonino and A. Bachorek, "A Simulator Coupling Architecture for the Creation of Digital Twins". *14th European Conference on Software Architecture Companion, (ECSA 2020)*, pp. 326-339

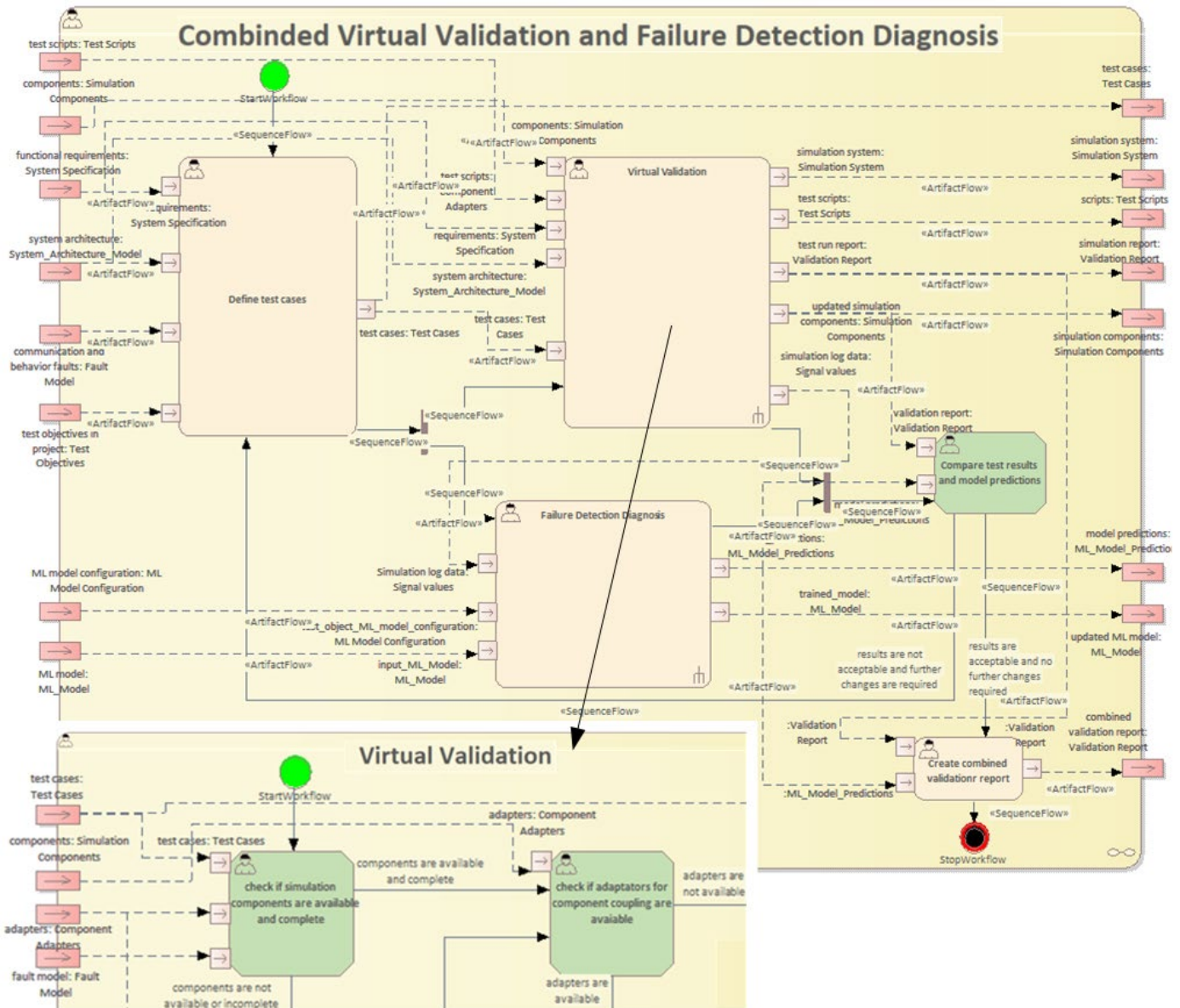


Fig. 2. V&V Workflow Specifications of the Combined Method and the referenced sub-method Virtual Validation (excerpt)