Multidimensional Framework for Characterizing Verification and Validation of Automated Systems

Copyright (c) 2022 IEEE. To appear in the proc. of 18th European Dependable Computing Conference (EDCC2022)

Joseba A. Agirre, Ahmet Yazici, Katia Di Blasio, Jose Luis de la Vara, Behrooz Sangchoolie,
Ugur Yayan, Raul Barbosa, Leire Etxeberria, Massimo Nazaria, Mustafa Karaca
Mondragon University, *Eskisehir Osmangazi University*, *Intecs Solutions S.p.a.*, *University of Castilla-La Mancha*,
RISE Research Institutes of Sweden, *Eskisehir Osmangazi University*, *University of Coimbra*,
Mondragon University, *Fondazione Bruno Kessler*, *IMTGD*
jaagirre@mondragon.edu, ayazici@ogu.edu.tr, katia.diblasio@intecs.it, JoseLuis.delaVara@uclm.es,
behrooz.sangchoolie@ri.se, ugur.yayan@ogu.edu.tr, rbarbosa@dei.uc.pt, letxeberria@mondragon.edu, mnazaria@fbk.eu,
mustafa.karaca@inovasyonmuhendislik.com

***Abstract*—Verification and Validation (V&V) of automated systems is becoming more costly and time-consuming because of the increasing size and complexity of these systems. Moreover, V&V of these systems can be hindered if the methods and processes are not properly described, analysed, and selected. It is essential that practitioners use suitable V&V methods and enact adequate V&V processes to confirm that these systems work as intended and in a cost-effective manner. Previous works have created different taxonomies and models considering different aspects of V&V that can be used to classify V&V methods and tools. The aim of this work is to provide a broad, comprehensive and a easy to use framework that addresses characterisation needs, rather than focusing on individual aspects of V&V methods and processes.To this end, in this paper, we present a multi-domain and multi-dimensional framework to characterize and classify V&V methods and tools in a structured way. The framework considers a comprehensive characterization of different relevant aspects of V&V. A web-based repository has been implemented on the basis of the framework, as an example of use, in order to collect information about the application of V&V methods and tools. This way, practitioners and researchers can easily learn about and identify suitable V&V processes.**

***Index Terms*—verification and validation, V&V, automated systems, framework, method, tool, workflow.**

I. Introduction

Automated systems, such as industrial robots and medical devices, have become an intrinsic part of the contemporary world. These systems are expected to autonomously and safely interact with the environment in which they operate, thus it is vital to verify and validate their behavior over their entire development life cycle. To this aim, Verification and Validation (V&V) processes are performed from the very initial system design phase to the final testing phases and beyond. V&V is becoming more costly and time-consuming due to the increasing complexity and connectivity of automated systems. This is emphasized by the fact that these systems are made of a variety of interconnected sub-systems developed by different stakeholders

The ever-growing need of improving traditional V&V workflows so as to reduce costs as well as to increase productivity has been driving the implementation of innovative methods and tools in both commercial and research areas. However, the heterogeneous and domain-specific nature of V&V processes makes the evaluation and adoption of state-of-the-art methods and workflows by practitioners challenging, which is highlighted by the fact that every industrial sector must comply with its distinct norms, practices and terminologies [1].

To overcome these obstacles, this paper presents a multidimensional framework which defines a domain-agnostic way of characterizing and classifying V&V processes in a structured way, with the aim of addressing a broad set of characterization and classification aspects around the methods, tools, environments, and concepts needed to verify and validate automated systems.

The presented framework was developed as part of the EU-funded project VALU3S [2] [3], which consists of a consortium of 41 industrial and academic partners, and is used in 13 representative pilots in six different domains of automotive, railway, aerospace, agriculture, health, and industrial robots. In addition, a web-based repository has been implemented and populated on the basis of the presented framework[1], so as to help academic and industrial practitioners learn about innovative V&V toolchains as well as real use case examples, regardless of the specific domains they were originally intended for.

II. Related Work

In a previous publication, we proposed a classification scheme for V&V methods [4] that considers their type of evaluation, e.g., testing, and if they deal with safety, cybersecurity, or privacy. It can be regarded as an antecedent of the framework presented in this paper.

Some earlier work have focused on creating taxonomies considering various V&V aspects. Utting et al. [5] proposed a taxonomy that provides the essential characteristics of model-based testing, Pesola [6] proposed a framework for early product V&V, and Firesmith [7] presents a taxonomy for

[1]The repository will become publicly available in November 2022, whereas this paper focuses only on the presentation of the framework.

testing that deals with seven abstract classes of test types. Such a taxonomy aims to help testers to create the test plans by evaluating various test types.

Metamodels for characterisation of V&V can also be found in the literature, but they are mostly focused on specific V&V areas, e.g., testing [8] and certification [9]. The metamodels define concepts, attributes of the concepts, and relationships among the concepts that characterise the corresponding V&V area. The metamodels have also been used to develop concrete solutions, e.g., the Eclipse OpenCert tool platform [10] for assurance and certification of Cyber-Physical Systems (CPS). Although such an implementation supports the specification of certain information about V&V methods used for certification purposes, the support provided is at a high abstraction level and not detailed. For example, the metamodel of OpenCert does not define the possible types of evaluation of a V&V method or of its environment. In the scope of model-based software engineering, it has been proposed to distinguish V&V needs at model, code, and hardware levels [11].

Some engineering standards, e.g., [1], define V&V activities for different types of components or systems, also considering safety- and security-specific needs. However, they do not deal with other relevant characteristics to consider when selecting or using a V&V method. These characteristics include the evaluation environment, the tool support, the type of requirement under evaluation, and performance indicators. The NIST Framework for CPS [12] describes certain V&V aspects and facets. The aspects include concepts related to CPS such as functional, human, and timing concerns. There are three identified facets: conceptualization, realization, and assurance. The elements of assurance are claims, argumentation, confidence, and evidence which is focused on testing concepts.

Regarding the new challenges of V&V for automated system, some studies have focused on providing overviews of certain aspects for large-scale and dependable systems. The AMADEOS framework [13] describes both a high-level perspective and a viewpoint-based specialization for SoS (Systems of Systems). The high-level perspective consists of four different layers, namely mission, conceptual, logical, and implementation. The viewpoint-driven analysis includes the components of structure, dynamicity, evolution, dependability and security, time, multi-criticality, and emergence. The AMADEOS framework is more focused on SoS concepts than on general V&V characteristics. Sinha et al. [14] present results of a survey of formal methods that can be used for dependability analysis of industrial automation systems, whereas Lahami and Krichen [15] have studied runtime testing of dynamically adaptable and distributed systems, giving details of approaches, frameworks, and tools. Ota et al. [16] proposed a V&V test framework for open system architectures that is used to unify the interfaces of sub-systems and to enhance platform interoperability. Engstrom et al. [17] proposed SERP-test, a taxonomy that aims at improving communication among researchers and practitioners in the area of software testing. This can also help reduce the effort for developing V&V approaches for automated systems.

The seminal work by Avizienis et al. [18] characterizes in detail certain aspects of dependability, security, and V&V, e.g., basic concepts, types of faults, and evaluation types. However, it does not address some important aspects such as the evaluation tool type, the evaluation stage, and the purpose of the component under evaluation. Some initiatives [19] [20] propose the use of cost, benefit, effectiveness, and applicability as criteria for V&V characterisation.

In summary, the main contribution beyond the state of the art on ways to characterise V&V is the provision of a broader, more comprehensive framework that addresses a large set of characterisation needs, instead of focusing on single aspects such as testing or tool types, or on a reduced set of aspects. This way, the framework enables a more detailed characterisation, in turn allowing a user to make more informed decisions upon V&V means.

III. Framework Overview

A. Stakeholders

The potential stakeholders of the framework and the web repository are divided into two groups:

1) *VALU3S project members* who are industrial partners (e.g., V&V tool vendors or use case providers), V&V researchers, system, software, or hardware designers and developers, and Quality Assurance (QA) engineers and managers.
2) *Community members* who are all those users not involved in the VALU3S project, while working with V&V of automated systems. These stakeholders will also have a public access to the web repository.

The stakeholders are required to register themselves in order to use the repository functionalities. Population of the repository is also possible for both groups of stakeholders given that the contents to be added go through a review process.

B. User Stories

Objectives of each stakeholder in using the framework could differ with respect to their needs in different V&V activities. Therefore, the project members defined 24 user stories in order to identify the set of needs that the framework must cover. These user stories define the functionality to be implemented in the VALU3S framework and have been used in the validation process of its implementation. The 24 user stories are divided in 4 main types of user stories related with V&V activities:

1) Characterize V&V method
2) Characterize V&V tool
3) Search and compare V&V methods
4) Search and compare V&V tools

C. Structure of the framework

The structure of the V&V framework builds on two main concepts: dimensions (see §IV) and artefacts (see §V). Dimensions are the set of properties that allows a user to classify the relevant characteristics of the artefacts. Artefacts are the methods and tools used in the V&V process and applied to specific systems and domains (i.e., instantiated to specific use cases and relevant scenarios, test cases, etc.). In other words, the framework enables the classification of the V&V methods and tools into the different dimensions. Additionally, the framework enables the storage of other artefacts such as evaluation scenarios, requirements, and test cases defined in the VALU3S project [21] . In this way, when carrying out a new V&V task, users can look for V&V methods and tools that can meet their needs and optimize their workflow.

Eight dimensions have been identified for classifying the framework artefacts (see §IV). Each dimension (D) is associated with a question:

- What are we evaluating? The question is answered by providing details about the type of component (or system) under evaluation (D3, see §IV-C) and the purpose of the component (D6, see §IV-F).
- When are we evaluating? The question is answered by indicating the evaluation stage (D5, see §IV-E).
- Where are we evaluating? The question is answered by selecting of the environment type (D1, see §IV-A).
- Why are we evaluating? The question is answered by providing details about the type of requirement (D7, see §IV-G).
- How are we going to evaluate? The question is answered by indicating the type of evaluation method used (D2, see §IV-B), as well as the type of tool under focus (D4, see §IV-D).
- How can we measure the level of improvement that the evaluation offers, e.g., with respect to the time and cost of evaluation? The question is answered by identifying the type of Evaluation Performance Indicators (D8, see §IV-H).

In order to show the structure and the relationships between the elements of the framework, a UML class diagram has been created (see Fig. 1 for a simplified version of such diagram). The central elements of the V&V Framework are the V&V Method and V&V Tool classes, which are categorized by the dimensions. On the User side, the focus is on Use Case Scenarios, that identify a set of Test Cases. Each Test Case is decomposed in a Workflow and a Context. The Workflow represents the sequence of the activities that characterise the test case. Each activity of the workflow is related to a corresponding V&V Method. The Context represents the set of elements over which the test case operates, i.e., the component under test, the environment, etc. Among these elements, the Testing Tools adopted to conduct the test case, are related to the corresponding V&V Tools. The Workflow and Context elements are categorized using dimension as well, this way Users can make use of the framework as a support tool during the selection of methods and tools to implement their test cases.

IV. Dimensions of the Framework

This section provides detailed information about the different dimensions of the framework. In characterization and derivation of V&V methods and tools, Dimensions are descriptive elements of V&V framework. Dimensions provide a road-map for the user who has specific requirements for V&V processes. The core of the framework structure was presented in the VALU3S project Description of Action (DoA). In the proposal, 6 dimensions and their respective layers were defined. Using these initial dimensions and their layers as a starting point a survey was carried out among the 43 partners of the VALU3S project. As a result of the survey the 8 dimensions of the VALU3S framework were designed (see Fig. 2). For each dimension, different layers are defined. A layer represents a specific way in which a dimension can be addressed. For example, the type of requirement under evaluation (dimension) of a V&V method can be safety or cybersecurity (layers).

A. Dimension 1: Evaluation Environment Type

This dimension consists of elements that support V&V execution with software, hardware, and network configurations. In system verification, different test environments can be identified and different V&V activities can be carried out. These environments can provide conditions simulating expected real-life conditions, or enable V&V under expected normal or possible abnormal conditions, configurations and situations [22]. The V&V framework defines three high-level environment layers (note that, an evaluation environment could also be a combination of the below environments.):

1) **In-the-lab evaluation environment**. The evaluations conducted in-the-lab are those that are done mainly on systems' sub-components, as well as on models of the system that are still under development. The use of simulators as well as simulation environments [23], [24] are among common approaches within this layer.
2) **Closed evaluation environment**. The closed evaluation environment is the next logical evaluation step. Upon the availability of a system prototype, the evaluation of the system can be done in an environment that is close to the actual environment where the system is designed to be used in. The advantages of a closed evaluation environment are supervision and control over the consequences of the evaluation process. An example of the environments for the automotive domain is a closed track proving grounds such as AstaZero [25], where the prototype of automated functions could be evaluated e.g., in a city area, multi lane road, high speed area.
3) **Open evaluation environment**. Evaluations conducted in an open test environment could also be referred to as in-field-testing and usually correspond to final evaluation activities where the developed system is assessed in real-world situations.

B. Dimension 2: Evaluation Type

The objective of Evaluation Type dimension is to classify V&V methods based on the mechanisms used to evaluate the system. There are related studies done to identify and extend classification on general system models [26]. In V&V method

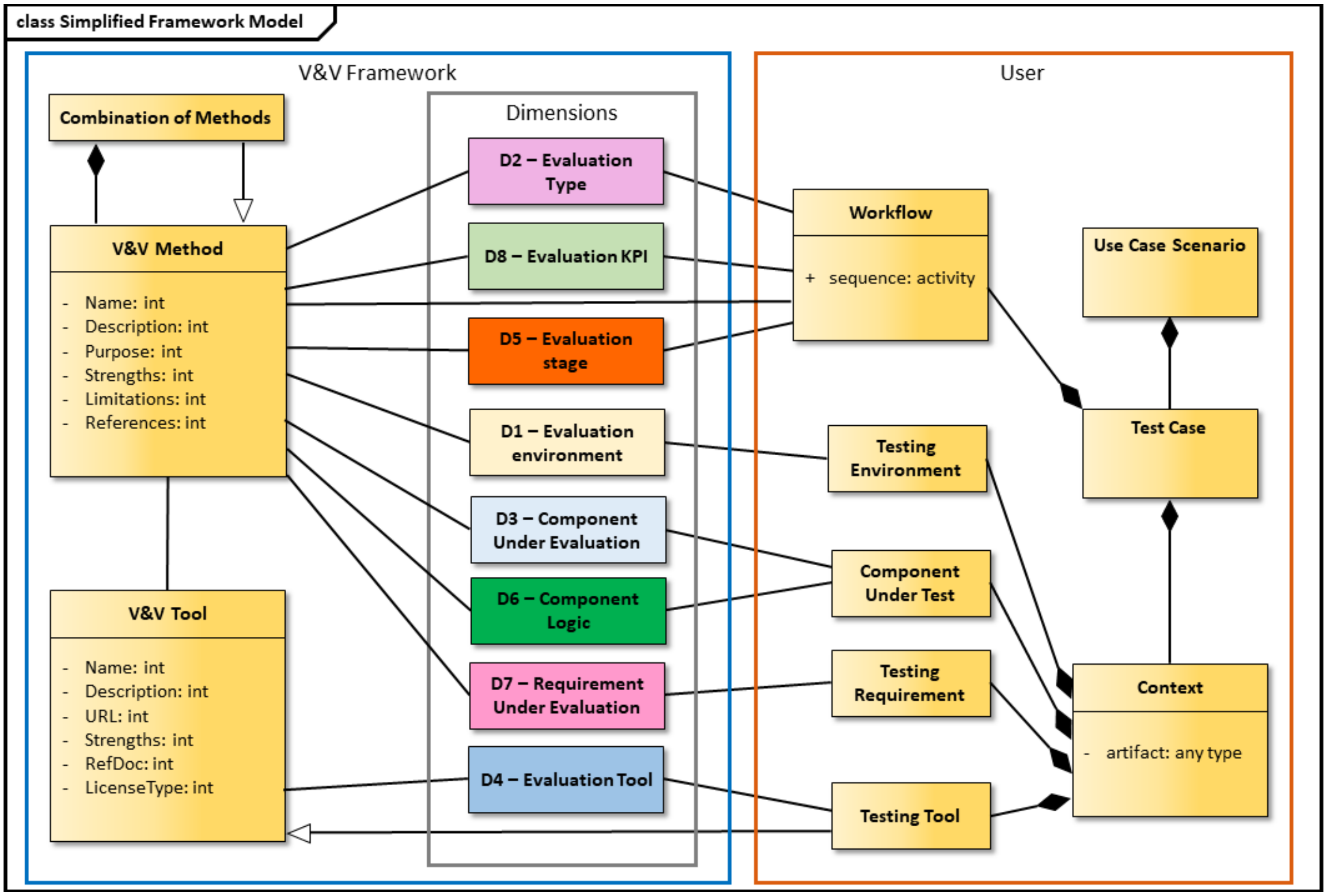

Fig. 1. UML diagram of the framework

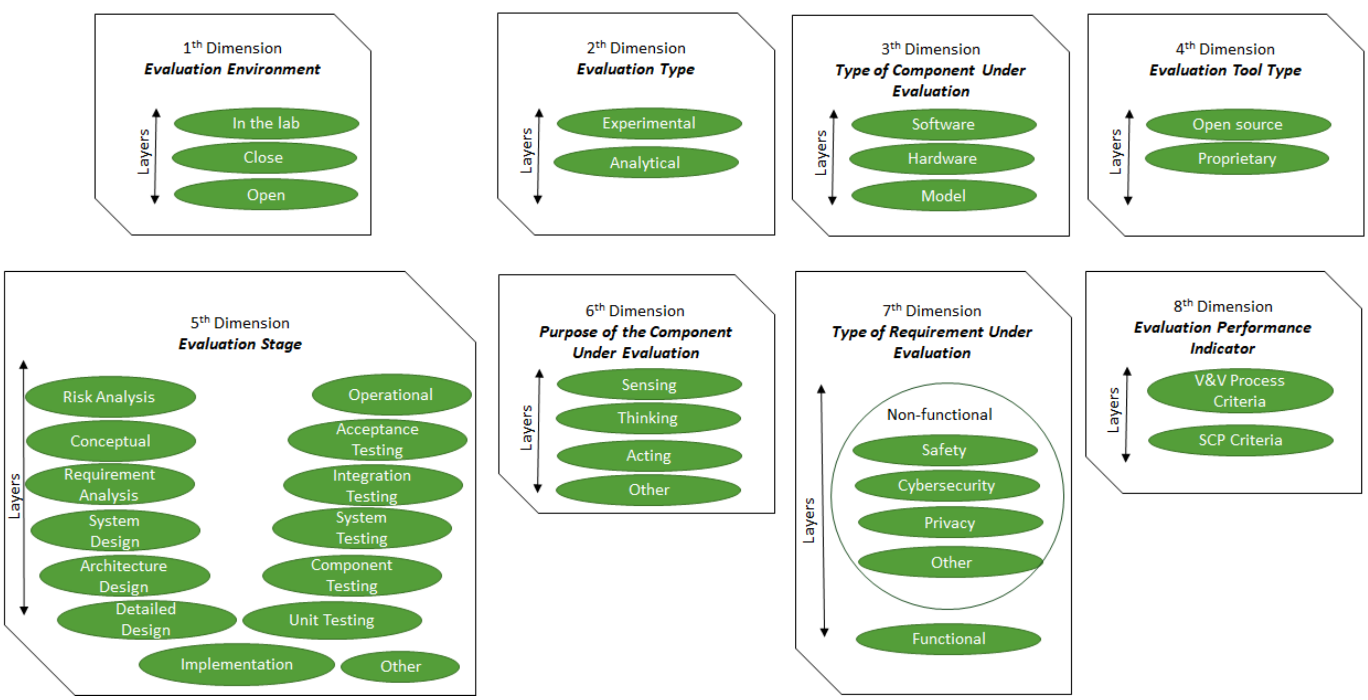

Fig. 2. Dimensions and layers of the framework

classification, these properties are categorized as experimental and analytic methods that form the layers of this dimension. Note that the V&V methods may address any of the requirements defined in §IV-G.

1) **Experimental methods**. Experimental methods are those relating to, or based on, experience or experiments. The layer is further decomposed in three sub-layers:
   - *Testing*: Testing involves the execution of a system or component (software, hardware or both) to evaluate one or more properties of interest.
   - *Monitoring*: Runtime monitoring is a lightweight and dynamic verification technique that involves observing the internal operations of a system and/or its interactions with other external entities, with the aim of determining whether the system satisfies or violates a correctness specification.
   - *Simulation*: A simulation is an approximate imitation of the operation of a process or system over time.
2) **Analytical methods**. Analytical methods are those using analysis or logical reasoning. This layer is further decomposed in the following three sub-layers:
   - *Formal*: Formal methods are based on formal mathematical proofs or correctness and are the most thorough means of V&V.
   - *Semi-formal*: Semi-Formal methods are types that follow formal structures. However, due to incomplete semantics, these methods require human reasoning to complete interpretation for result extraction.
   - *Informal*: Informal V&V methods do not follow predefined rules or mathematical bases in analysis.

C. Dimension 3: Type of Component Under Evaluation

This dimension specifies the elements on which V&V activities are or can be carried out in three layers (note that, a combination of elements could also be considered representing a system or part of a system.):

1) **Software**: Software includes the software components that interact with the physical parts of a system.
2) **Hardware**: Hardware includes the physical parts of a computing system.
3) **Model**. A model is an informative representation of a system.

Here, software components refer to any functionality that is written in the software code, which needs to be verified and validated. This also includes any software-in-the-loop system [27] as well as integrations among different software components. With respect to the hardware components, the V&V process needs to be conducted e.g., on cameras, connectors, sensors, hardware-in-the-loop, processing chips, and hardware integrations. With respect to models, we refer to conducting a V&V process on any component that is implemented as a model, such as Simulink models.

D. Dimension 4: Evaluation Tool

This dimension characterizes the kind of tool used to evaluate a system or component based on the requirements defined in §IV-G. In testing, many tools have been developed and rolled out by commercial vendors and adopted by market. There are numerous ways a tool may be characterized, e.g., according to the execution resources required, its level of automation, whether it is qualified according to standards, its level of availability. Some tools may be available as open source and documentation may have been made freely available to the public. Other tools are proprietary and may have costs associated with accessing and extending them. However, part of this cost may be due to qualification according to various SCP (safety, cybersecurity, privacy) standards. The layers of the evaluation tools are:

1) **Open Source**. Open source is a source code that is made available for possible modification and redistribution.
2) **Proprietary**. Proprietary software or tool is known as non-free software, or closed-source software.

E. Dimension 5: Evaluation Stage

V&V activities are typically performed at different stages of the development process. Earlier stages focus more on verification activities to check that the system is being developed correctly, while during the final stages, validation activities are often mandatory to check that the system provides its intended services. Development processes according to the commonly used V-model [28] are usual. Apart from verification activities such as reviews, analyses and tests performed on the results produced in each stage on the left-hand side of the V-model, verification, typically in the form of testing, is the focus of the activities on the right-hand side (see Fig. 2). Validation tests are, on the other hand, performed to check that the system performs its intended functions correctly as well as is compliant with the considered regulations or safety, cybersecurity and privacy standards. To summarize, the objective of this dimension is to support the determination of the evaluation activities executed in the different phases of development. For this purpose, the following layers have been defined:

1) **Concept**. In this stage, product feasibility is evaluated.
2) **Requirement Analysis/engineering**. The process for determining user expectations for a new/modified product.
3) **System Design**. The process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.
4) **Architecture Design**. The process of defining a collection of hardware and software components and their interfaces for the development of a system.
5) **Detailed Design**. The process of refining and expanding the preliminary design of a system or component so that the design is sufficiently complete to be implemented.
6) **Implementation**. Realization of a technical specification or algorithm as a software/hardware component or other computer system through computer programming.
7) **Unit testing**. Checking if components are fulfilling functionalities or not.
8) **Integration testing**. Checking the flow from one module to other modules.

9) **System testing**. Evaluating both functional and non-functional needs for the testing of a system as a whole.
10) **Acceptance testing**. Checking if the requirements of a specification or contract are met as per its delivery.
11) **Operation**. The stage after the delivery of the system, when the system is running.
12) **Risk analysis**. The process for comprehending the nature of hazards and determining the level of risk.
13) **Other**. Any stages not covered in the previous layers.

The layers have been inspired by the V-model [28] stages, but the usage of the framework is not tied to the V-model. In fact, the "Other" layer has been added to support the consideration of other activities and the "Risk Analysis" layer has been added as it is a key activity in certain domains.

F. Dimension 6: Purpose of the Component Under Evaluation

This dimension represents the component from the point of view of its overall aim, referring to the following layers [29]:

1) **Sensing**. Sensing the environment.
2) **Thinking**. Making a decision based on a predefined task and the information sensed from the environment.
3) **Acting**. Performing a predefined task by adapting to the environment.
4) **Other**. Any other purpose that cannot be classified in the previous layers.

G. Dimension 7: Type of Requirement Under Evaluation

The development cycle of automated systems starts with setting up system requirements, which are created according to user needs. This dimension is targeted to identify both non-functional and functional requirements:

1) **Non-functional Requirements**. A non-functional requirement is a requirement that specifies criteria that can be used to judge the quality of the operation of a system, rather than specific behaviors.
    - *Safety*: Safety is freedom from unacceptable risk, where risk could be defined as a combination of the probability of occurrence of harm and the severity of that harm. Safety can also refer to the control of recognized hazards in order to achieve an acceptable level of risk.
    - *Cybersecurity*: Security of cyberspace, where cyberspace refers to the set of relationships between objects that are accessible through a generalised telecommunications network. It also considers the set of objects themselves, with their interfaces allowing remote control, remote access to data, or participation in control actions within that cyberspace.
    - *Privacy*: Degree to which unauthorized parties are prevented from obtaining "personal" sensitive information. Here "personal" explicitly addresses the concern connected to the personal information. In legal definition, privacy is concern of protecting an individual's rights according to the law. Privacy sets requirement for not to illegally publish, interfere or intrude personal information, activities, and spaces. [30]. In similar means, privacy includes subfactors of anonymity and confidentiality. [31]
    - *Other*: Other non-functional requirements such as availability, reliability, maintainability, etc.
2) **Functional Requirements**. Functional requirements define specific behavior or functions.

H. Dimension 8: Type of Evaluation Performance Indicators

This dimension deals with identification of two types of evaluation criteria that reflect and measure the improvement obtained with respect to V&V:

1) **V&V process criteria**. Evaluation performance indicators related to the improvement of the V&V process.
2) **SCP Criteria**. Evaluation performance indicators related to SCP criteria of the systems.

The V&V process criteria are those related with the improvement of the V&V process, for example, those criteria related with time, cost, effort, and coverage. SCP Criteria, on the other hand, are those related with SCP, such as the number of safety/security requirement violations or the number of prevented accidents (safety).

V. Artefacts of the Framework

As mentioned earlier, a web-based repository has been implemented to manage V&V information according to the framework. The repository is implemented using Plone CMS [32] and collects V&V methods and tools, as well as artefacts related to the application of such methods in a workflow for concrete use cases. Artefacts are organized according to the eight dimensions characterized in §IV and depicted in Fig. 2. The overall goal of the framework is to facilitate the substantiation of future decisions based upon experience from previous projects; specifically, understanding what has worked in the past, knowing what techniques are commonly applied in relevant scenarios, searching for cost-effective ways to address V&V challenges in new scenarios, and to support continuous improvement of V&V processes. Here, we describe the different kinds of artefacts collected in the repository.

1) **Method**: A method corresponds to concrete procedures that can be applied to V&V activities.
2) **Tool**: A tool corresponds to a software tool used as support for a method. Tool support is fundamental for achieving cost-effective application of V&V methods.
3) **Use Case**: Use cases consist in a real engineering effort with respect to SCP requirements. Such use cases allow one to document V&V efforts and establish relationships to methods and tools that have been successfully applied in the past. Every use case has an owner and contributors.
4) **Organization**: This artefact characterizes the companies or partners contributing to a use case.
5) **Domain**: Every use case is connected to one or more domain(s). We are currently considering the following domains: Automotive, Agriculture, Railway, Healthcare, Aerospace or Industrial Robotics/Automation. New domains may be added if needed.

6) **Evaluation Scenario of a use case**: An evaluation scenario establishes a particular focus or specific V&V objective with which to evaluate a use Case.
7) **Requirement (related to dimension 7)**: Specific requirements evaluated during V&V activities are stored, which can be useful for the user to search for common activities.
8) **Test Case or V&V Activity**: This artefact specifies the evaluation activities performed in a use case (i.e., to verify one or more requirements) and is linked to a method. In practice, the test case is performed using a method defined in the framework and consists of a set of preconditions, input conditions, and expected results.
9) **Component Under Evaluation** (related to the dimension 3 and 6): The specific component, set of components, or system that is being evaluated by a particular test case (e.g., an operating system kernel).
10) **Evaluation Criteria** (related to dimension 8): The criteria that are measured in the evaluation activity, which could reflect SCP or V&V process criteria (e.g., failure rate resulting from a fault injection campaign).
11) **Context and Environment** (related to the dimension 1): The specific environment elements used in the evaluation activity.
12) **Standard**: The standard that is being followed for a certain V&V activity. A use case provider may e.g., be following IEC 61508, which provides functional safety guidelines for the lifecycle of electronic systems.
13) **Workflow**: The concrete process and tooling workflow for V&V of an automated system. This also facilitates effective combination of distinct methods, e.g, Model Checking and Human Interaction Safety analysis.

VI. EVALUATION

In order to evaluate the efficiency of the proposed framework, we first present a comparison between different frameworks for categorising V&V processes; we then present the way the framework and its web based implementation can be used to search for V&V methods. To compare the different V&V process categorisation frameworks, the user stories defined by the 43 industrial partners of the VALU3S European project were used. Due to space limitations, in the comparison, only 5 user stories have been used out of the total of 24. Table I shows the compared frameworks and their impact in the selected 5 user stories taking into account the number of dimensions and layers of the framework. When assessing the impact on user stories, three levels are considered: *low*, *medium* and *fully*. The table shows that although the dimensions of the VALU3S framework are general, they allow to address at the same time very important aspects of the V&V process. It can be concluded that the dimensions of the framework can cover crucial aspects of the V&V process.

In order to show the usability of the framework, here we explain an example of its use in the VALU3S project. One of the methods stored in the repository is the "Model Based Safety Analysis with Failure Logical Analysis" method. The method is associated with the Eclipse CHESS tool [33], and can be characterized according to the following:

- **Evaluation environment**: In the lab.
- **Evaluation type**: Analytical-Semi Formal.
- **Type of component under evaluation**: Software, Hardware, Model.
- **Evaluation stage**: Unit Testing, Integration Testing, System Testing.
- **Purpose of the component under evaluation**: Sensing, Thinking, Acting.
- **Type of requirements under evaluation**: Functional, Non-functional Safety, Non-functional Cybersecurity.
- **Evaluation performance indicator**: Safety Criteria.

The CHESS tool, on the other hand, is characterized as: **Evaluation Tool**: Open source.

The characterization criteria used to store information in the repository can help practitioners when they need to search and compare methods and tools. For example, a practitioner looking for a method used for the semi-formal analysis of a system can take a look at the above mentioned method and compare it with the others stored in the repository, having the same evaluation type. In addition to have a catalogue of methods and tools, the repository enables to stores information about the concrete application of them. These information can be also useful for practitioners. For example, for the above mentioned method and tool, the repository stores a concrete application of them in different use cases, such as: **Use case**: Neuromuscular Transmission for Muscle Relaxation Measurements. The method can be used for Software, Hardware and Model, but, in this concrete application, it has been applied to the system's model.

Six use case scenarios have been defined for this use case. The described method has been applied in one of them: **Use case scenario**: Safety analysis and certification, which aims to improve the specific model-based safety analysis. This scenario has several test cases defined and stored. Each of them has a context with: environment description, requirements to be tested (related to safety in this case), used tool (CHESS) and components under test.

VII. CONCLUSIONS

This paper presented a multi-domain framework to characterize and classify V&V processes, concluding that the artefacts and dimensions of the VALU3S framework enables the classification of different methods and tools in a comprehensive way. In addition, a web-based repository has been implemented to let users obtain information about methods and tools, and also as an example of how to use the framework. Such repository was implemented on top of the Plone CMS [32] and the content types created have been defined using the VALU3S framework as a basis. The web repository has been populated with 55 V&V methods, 29 tools and 12 use cases with 52 evaluation scenarios, 193 test cases related to 155 requirements and 23 standards, of 43 involved organizations of VALU3S project. Using the web repository practitioners and researchers can easily learn about and identify suitable V&V

TABLE I
COMPARISON OF V&V CHARACTERISATION FRAMEWORKS.

Description of User Story (VALU3S Dimensions used)	IEEE Std (1012)	NIST	Eclipse OpenCert	Amadeos	Seminal Work by Avizienis	VALU3S
Characterise the V&V methods (All Dimensions)	medium	medium	medium	low	medium	fully
Characterise the V&V tool (D4)	medium	low	medium	low	low	fully
Derive a catalogue of suitable V&V tools/methods for my test object by describing e.g. intended domain, standard, type of component, test type (All Dimensions)	low	medium	medium	low	low	fully
Identify V&V methods/tool applicable on system/software/hardware/model level (D3, D6)	medium	low	medium	medium	medium	fully
List available V&V methods/tools rated according to a certain KPI (D8)	low	low	low	low	low	fully

processes. The repository will become publicly available in November 2022 and prior to its release, we plan on making use of independent users to evaluate the usefulness and user-friendliness of the repository.

ACKNOWLEDGEMENT(S)

This work has been developed in the VALU3S project, which has 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 programme and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain (MCIN/AEI ref. PCI2020-112001; NextGen.EU/PRTR), Sweden, and Turkey (TUBITAK, under contract no:119N356). We would like to thank the VALU3S partners who contributed to the design of the framework. Moreover, the research leading to this paper has received funding from the ETHEREAL (MICINN/AEI ref. PID2020-115220RB-C21; ERDF) and Treasure (JCCM ref. SBPLY/19/ 180501/000270; ERDF) projects, and the Ramon y Cajal Program (MCIN/AEI ref. RYC-2017-22836; ESF).

REFERENCES

[1] "IEEE standard for system, software, and hardware verification and validation," *IEEE Std 1012-2016 (Revision of IEEE Std 1012-2012/ Incorporates IEEE Std 1012-2016/Cor1-2017)*, pp. 1–260, 2017.

[2] "ECSEL JU VALU3S project." [Online]. Available: https://valu3s.eu/

[3] J.A. Agirre *et al.*, "The VALU3S ECSEL project: Verification and validation of automated systems safety and security," *Microprocessors and Microsystems*, vol. 87, p. 104349, 2021.

[4] Jose Luis de la Vara *et al.*, "A proposal for the classification of methods for verification and validation of safety, cybersecurity, and privacy of automated systems," in *14th Inte. Conf. on the Quality of Information and Communications Technology*, 2021.

[5] M. Utting, A. Pretschner, and B. Legeard, "A taxonomy of model-based testing approaches," *Software Testing Verification & Reliability*, vol. 22, no. 5, SI, pp. 297–312, AUG 2012.

[6] J.-P. Pesola, "Building framework for early product verification and validation," *VTT Publications*, no. 736, 2010. [Online]. Available: https://cris.vtt.fi/en/publications/building-framework-for-early-product-verification-and-validation

[7] D. Firesmith, "A taxonomy of testing," Carnegie Mellon University's Software Engineering Institute Blog, Aug. 31, 2015. [Online]. [Online]. Available: http://insights.sei.cmu.edu/blog/a-taxonomy-of-testing/

[8] Object Management Group, "UML testing profile 2, version 2.1," 2019. [Online]. Available: https://www.omg.org/spec/UTP2/

[9] J. L. de la Vara, A. Ruiz, K. Attwood, H. Espinoza, R. K. Panesar-Walawege, Á. López, I. del Río, and T. Kelly, "Model-based specification of safety compliance needs for critical systems: A holistic generic metamodel," *Inf. Softw. Technol.*, vol. 72, pp. 16–30, 2016.

[10] J. L. de la Vara, A. Ruiz, and G. Blondelle, "Assurance and certification of cyber–physical systems: The AMASS open source ecosystem," *Journal of systems and software*, vol. 171, p. 110812, 2021.

[11] J. Schumann and K. Goseva-Popstojanova, "Verification and validation approaches for model-based software engineering," in *2019 ACM/IEEE 22nd Int. Conf. on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019, pp. 514–518.

[12] Cyber-Physical Systems Public Working Group, "Framework for cyber-physical systems: Volume 1, overview, version 1.0, NIST special publication 1500-201," 2017.

[13] A. Babu, S. Iacob, P. Lollini, and M. Mori, "Amadeos framework and supporting tools," in *Cyber-Physical Systems of Systems: Foundations – A Conceptual Model and Some Derivations: The AMADEOS Legacy*. Springer International Publishing, 2016.

[14] R. Sinha, S. Patil, L. Gomes, and V. Vyatkin, "A survey of static formal methods for building dependable industrial automation systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, JUL 2019.

[15] M. Lahami and M. Krichen, "A survey on runtime testing of dynamically adaptable and distributed systems," *Software Quality Journal*, 2021.

[16] D. Ota, P. Charchalakis, and E. Stipidis, "Towards a verification and validation test framework for open system architectures," in *2017 Int. Conf. on Military Technologies (ICMT)*, pp. 115–122.

[17] E. Engstrom, K. Petersen, N. bin Ali, and E. Bjarnason, "Serp-test: a taxonomy for supporting industry-academia communication," *Software Quality Journal*, vol. 25, no. 4, SI, pp. 1269–1305, DEC 2017.

[18] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.

[19] L. Miller, J. Hayes, and S. Mirsky, *Guidelines for the verification and validation of expert system software and conventional software*. US Nuclear Regulatory Commission, 1995.

[20] M. Roza, "Verification, validation and uncertainty quantification methods and techniques," NATO, 2014.

[21] Aleš Smrčka *et al.*, "VALU3S vocabulary," Tech. Rep., 2022. [Online]. Available: https://valu3s.eu/wp-content/uploads/2022/06/VALU3S-Vocabulary-v2.6-Final.pdf

[22] Federal Election Commission *et al.*, "Performance and test standards for punchcard, marksense, and direct recording electronic voting systems," 1990.

[23] M. Maleki and B. Sangchoolie, "SUFI: A Simulation-based Fault Injection Tool for Safety Evaluation of Advanced Driver Assistance Systems Modelled in SUMO," in *2021 17th European Dependable Computing Conference (EDCC)*, 2021, pp. 45–52.

[24] M. Malik, M. Maleki, P. Folkesson, B. Sangchoolie, and J. Karlsson, "ComFASE: A tool for evaluating the effects of V2V communication faults and attacks on automated vehicles," in *52nd Annual IEEE/IFIP Int. Conf. on Dependable Systems and Networks (DSN)*, 2022.

[25] "AstaZero test track." [Online]. Available: https://www.astazero.com/the-test-site/about/

[26] G. I. Efremov and J. A. Geller, "Experimental-analytical method of technological processes modeling in education," in *SHS Web of Conferences*, vol. 29. EDP Sciences, 2016, p. 02013.

[27] M. F. S. Ahamed, G. Tewolde, and J. Kwon, "Software-in-the-loop modeling and simulation framework for autonomous vehicles," in *2018 IEEE Int. Conf. on Electro/Information Technology (EIT)*. IEEE, 2018.

[28] K. Forsberg and H. Mooz, "The relationship of system engineering to the project cycle," in *INCOSE Int. symposium*, vol. 1, no. 1. Wiley Online Library, 1991, pp. 57–65.

[29] M. Siegel, "The sense-think-act paradigm revisited," in *1st Int. Workshop on Robotic Sensing, 2003. ROSE' 03.*, 2003, pp. pp. 5–9.

[30] F. Li, H. Li, B. Niu, and J. Chen, "Privacy computing: Concept, computing framework and future development trends," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 1145, 2018.

[31] D. Firesmith, “Common concepts underlying safety, security, and survivability engineering,” Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU/SEI-2003-TN-033, 2003.
[32] Plone Foundation, “Plone CMS: Open source content management,” https://plone.org/, August 2021.
[33] ECLIPSE Foundation, “Eclipse CHESS tool,” 2022. [Online]. Available: https://projects.eclipse.org/projects/polarsys.chess