Early De-Risking of Land Vehicles Open System Architecture Implementations

Daniel Ota

A thesis submitted in fulfilment of the
requirements of the University of Brighton
for the degree of Doctor of Philosophy

March 2019

Vetronics Research Centre
School of Computing, Engineering and Mathematics
The University of Brighton, UK

Declaration

I declare that the research contained in this thesis, unless otherwise formally indicated within the text, is the original work of the author. The thesis has not been previously submitted to this or any other university for a degree, and does not incorporate any material already submitted for a degree.

The Land Vehicle Verification Concept discussed in Chapter 3 was developed by the author during the draft phase of the NATO Generic Vehicle Architecture. The concept has been discussed in the NGVA Verification and Validation Group several times and has been ratified by North Atlantic Treaty Organization nations as part of STANAG 4754 NATO Generic Vehicle Architecture.

The Test Framework discussed in Chapter 4 was developed by the author. The design of the conformance test system and test suite was conducted in collaboration with Ditmir Hazizi under the research grant *NGVA Verification* supported by the German Ministry of Defence.

The Sub-System Verification Case Study in Chapter 5 has been designed by the author. The Model Maturity Testing was conducted by the author in the study *Interoperability Test Methods for Future Military Land Vehicles* at Fraunhofer FKIE under a research grant of the German Ministry of Defence. The implementation of the different test laboratory components was a group project of the Fraunhofer FKIE *Platform Capability Integration* team. The team members who worked on this programme are Reinhard Claus, Ditmir Hazizi, Manas Pradhan, and the author.

Any work that is not contributed by the author is clearly indicated within the text.

Daniel Ota, March 2019

Abstract

Military land vehicles have life cycles spanning over decades. However, equipment demand is regularly changing and seamless integration of new components is required. For facilitating sub-systems exchangeability and to standardise vehicle sub-system interfaces, Open System Architectures are under development. In the land systems domain, several European nations are defining the NATO STANAG 4754 *NATO Generic Vehicle Architecture* (NGVA). The assessment of future implementations requires new certification approaches and up-to-date verification frameworks are needed for early de-risking.

Therefore, first a generic concept for the Verification and Validation of military land vehicles is presented. It focuses on outlining a detailed verification plan, which can be tailored to nation and system specifics. For assessing the conformity of NGVA systems, sequentially-related compatibility levels have been developed, which facilitate the evaluation of the specific system requirements and form the basis for a verification process.

Second, a framework for the verification of vehicle sub-systems is discussed. It aims at providing verification mechanisms and reference implementations as early as possible to de-risk the sub-system design and certification process. The framework encourages to test the standard itself during the specification phase and to re-use resulting artefacts for systems verification in the beginning of the development cycle.

Third, an evaluation of the verification framework by means of a case study focusing on data model maturity aspects is presented. The case study was further extended for conformance and interoperability testing of NGVA-compliant system interfaces and the re-usability of test artefacts from data model testing was shown.

The results can be summarised as an approach for verifying sub-system implementations of modern military vehicles adhering to open standards. The verification measures focus on early phases of the standard specification and realisation and aim to minimise design and implementation risks from the beginning of a standards life cycle.

Contents

List of Figures

List of Tables

List of Abbreviations

AEP Allied Engineering Publication. 14, 17, 18, 26, 36, 39–41, 43, 55, 90, 100, 106, 107

AUTOSAR AUTomotive Open System ARchitecture. 8, 9, 11

C2IS Command and Control Information System. 12, 63, 75, 76, 85, 87, 88, 98, 100

CAN Controller Area Network. 11, 12, 101

COTS Commercial off-the-shelf. 85, 101

CR Compulsory Requirement. 15, 16, 18, 58

CT Conformance Testing. vii, 10, 11, 13, 23, 34, 35, 45–52, 54–56, 58, 59, 61, 75, 78, 80, 95, 103

CTSA Crew Terminal Software Architecture. 14, 15, 100

DDS Data Distribution Service. vii, ix, 16–20, 22, 23, 32, 56–58, 61, 64, 65, 68, 73, 75, 90, 102

DDSI Data Distribution Service Interoperability. 16, 58

Def Stan Defence Standard. 1

DM Data Model. vii, ix, xvi, 2–4, 14, 15, 21, 22, 32, 35, 40, 56–59, 61, 62, 64, 66, 69, 75–80, 86, 90, 93, 100, 103, 106, 107

EMC Electromagnetic Compatibility. 39

ETSI European Telecommunications Standards Institute. 11, 13, 46

EUT Equipment Under Test. 12, 48, 49, 51, 53

FACE Future Airborne Capability Environment. 9, 56

GPS Global Positioning System. 40, 77, 107

GVA Generic Vehicle Architecture. 1, 2, 7, 22, 23, 25, 28, 29, 74, 90, 91, 107

HMI Human Machine Interface. 99, 100

HUMS Health and Usage Monitoring System. 57, 58, 77, 107

ICS Implementation Conformance Statement. 51

IDL Interface Definition Language. vii, 22, 23, 58, 59, 62, 64–66, 70, 82, 89, 91, 92, 102

IEC International Electrotechnical Commission. 26, 27, 31

IEEE Institute of Electrical and Electronics Engineers. 8, 27, 31, 36, 78, 104

IFS Interoperable Features Statement. 51

IOT Interoperability Testing. vii, 10–13, 34, 35, 46–52, 55, 56, 59, 61, 64, 79, 80, 106, 107

IP Internet Protocol. 13, 56

ISO International Organisation for Standardisation. 8, 13, 26–28, 31

IT Information Technology. 1, 6

ITU International Telecommunication Union. 13

IUT Implementation Under Test. 11, 23, 47, 48, 51, 52, 58

IV&V Independent Verification and Validation. 13, 14, 36–38

LIN Local Interconnect Network. 11

MDA Model Driven Architecture. vii, 14, 22

MML Module Maturity Level. viii, ix, xvii, 59–62, 64–67, 78–81, 85–87, 90, 93–95, 99–104, 107

MoD Ministry of Defence. xv, 1, 7

MOSA Modular Open Systems Approach to Acquisition. 7, 9

NASA National Aeronautics and Space Administration. 9, 40

NATO North Atlantic Treaty Organization. ii, iii, 2, 7, 55, 106

NCL NGVA Compatibility Level. vii, 38, 39, 41, 43, 75, 106, 107

NGVA NATO Generic Vehicle Architecture. ii, iii, vii–ix, xiv, xvi, 2–4, 7, 9, 10, 14–18, 21–44, 50, 55–59, 61, 62, 64, 66, 67, 69, 73–83, 86, 90, 93, 95, 99–107

NTRS NGVA DM Test Reference System. ix, 67–69, 72, 73, 93–95, 98, 99

OASIS Organization for the Advancement of Structured Information Standards. 13

OBSVA One Box Single Vehicle Architecture. 9

OE Optional Enhancement. 15, 16, 18, 58

OMG Object Management Group. 18, 58

OSA Open System Architecture. 2, 3, 7, 105, 106

OSI Open Systems Interconnection. 39

PIM Platform Independent Model. vii, viii, 22, 23, 57–66, 70–72, 79–83, 90, 91, 99

PLEVID Platform Extended Video Standard. 16, 101

QE Qualified Equipment. 12, 48, 49, 51, 53, 55, 57, 76, 77

QoS Quality of Service. 16–19, 32, 56, 67, 76, 90, 102, 103

RI Reference Implementation. 45, 49, 55, 59, 61, 98, 99, 104, 106

SIL Safety Integrity Level. 37

SIP Session Initiation Protocol. 13

SRD System Requirements Document. 26, 40, 41, 76, 103

STANAG Standardization Agreement. ii, iii, 2, 3, 7, 14–16, 26, 28, 34, 36, 38, 40, 43, 55, 57, 105–107

SUT System Under Test. 11, 12, 14, 29, 38, 47, 48, 51, 53, 55, 67, 68, 72, 73, 75, 77, 102

UML Unified Modelling Language. 22, 23, 59, 64, 95, 107

V&V Verification and Validation. ii, iii, xvi, 2–4, 8–10, 12–15, 25, 26, 28, 29, 31, 37, 38, 43, 44, 105–107

VBS Virtual Battlespace. 85, 100

Vetronics Vehicle Electronics. 1, 3, 11, 14, 15, 17, 18, 44, 57, 58, 62, 72, 74–76

VICTORY Vehicular Integration for C4ISR/EW Interoperability. 7, 12

VPN Virtual Private Network. 101

VRC Vetronics Research Centre. 12

List of Publications and Presentations

Published Papers

In order to underpin the research impact, the author published a series of conference papers – predominantly at IEEE-listed peer-reviewed conferences.

With respect to the contributions discussed in the thesis, three paper directly reflect the findings of chapter 3-5:

Daniel Ota. 'Towards Verification of NATO Generic Vehicle Architecture-Based Systems'. In: *ICCRTS 2016: 21st International Command and Control Research and Technology Symposium*. Sept. 2016.

Daniel Ota and Ditmir Hazizi. 'Interface Conformance Testing for Future Military Land Platforms'. In: *2017 International Conference on Military Communications and Information Systems (ICMCIS)*. May 2017, pp. 1–7. DOI: 10.1109/ICMCIS.2017.7956496.

Daniel Ota, Periklis Charchalakis and Elias Stipidis. 'Towards a Verification and Validation Test Framework for Open System Architectures'. In: *2017 International Conference on Military Technologies (ICMT)*. May 2017, pp. 115–122. DOI: 10.1109/MILTECHS.2017.7988742.

In addition, further research was conducted with respect to test applications prototypes that are deployed in the interoperability test laboratory, which is discussed in chapter 5. The research concerns realised NGVA gateways to the automotive domain, to robotic systems, and to higher echelons as well as the realisation of modern crew assistance systems:

Manas Pradhan and Daniel Ota. 'Integrating Automotive Bus-based Networks in the NATO Generic Vehicle Architecture'. In: *ICCRTS 2016: 21st International Command and Control Research and Technology Symposium.* Sept. 2016.

Manas Pradhan, Alexander Tiderko and Daniel Ota. 'Approach towards achieving Interoperability between Military Land Vehicle and Robotic Systems'. In: *2017 International Conference on Military Communications and Information Systems (ICMCIS).* May 2017, pp. 1–7. DOI: 10.1109/ICMCIS.2017.7956477.

Youssef Mahmoud Youssef and Daniel Ota. 'A General Approach to Health Monitoring & Fault Diagnosis of Unmanned Ground Vehicles'. In: *2018 International Conference on Military Communications and Information Systems (ICMCIS).* May 2018, pp. 1–7. DOI: 10.1109/ICMCIS.2018.8398694.

Manas Pradhan, Alexander Tiderko and Daniel Ota. 'Approach Towards achieving an Interoperable C4ISR Infrastructure'. In: *2017 International Conference on Military Technologies (ICMT).* May 2017, pp. 375–382. DOI: 10.1109/MILTECHS.2017.7988788.

Manas Pradhan and Daniel Ota. 'An Adaptable Multimodal Crew Assistance System for NATO Generic Vehicle Architecture'. In: *2016 International Conference on Military Communications and Information Systems (ICMCIS).* May 2016, pp. 1–8. DOI: 10.1109/ICMCIS.2016.7496556.

Manas Pradhan and Daniel Ota. 'Interface Design and Assessment of Situational Awareness and Workload for an Adaptable Multimodal Crew Assistance System based on NATO Generic Vehicle Architecture'. In: *ICCRTS 2016: 21st International Command and Control Research and Technology Symposium.* Sept. 2016.

Daniel Ota and Manas Pradhan. 'Modular Verification and Validation for NATO Generic Vehicle Architecture-based Land Platforms'. In: *2018 International Conference on Military Communications and Information Systems (ICMCIS).* May 2018, pp. 1–7. DOI: 10.1109/ICMCIS.2018.8398715.

Technical Reports

The work presented in this thesis was supported by research grants from the German Ministry of Defence. The results are composed in technical reports written by the author.

The first two grants served the preparation of V&V approach presented in chapter 3 and laid the foundation for the development of the interoperability test laboratory. The latter ones supported the work on conformance testing and the conduction of Data Model maturity testing.

Daniel Ota. *Generic Vehicle Architecture – Concepts and Testbed*. Technical Report. Wachtberg, Germany: Fraunhofer FKIE, May 2016.

Daniel Ota. *NATO Generic Vehicle Architecture Collaboration Network*. Technical Report. Wachtberg, Germany: Fraunhofer FKIE, Feb. 2015.

Daniel Ota and Ditmir Hazizi. *NGVA Verification*. Technical Report. Wachtberg, Germany: Fraunhofer FKIE, May 2017.

Reinhard Claus et al. *Interoperability Test Methods for Future Military Land Vehicles*. Technical Report. Wachtberg, Germany: Fraunhofer FKIE, Mar. 2018.

Invited Talks

As chair of the NGVA V&V Working Group, the author was invited to present on up-to-date activities in the field of military land platform verification:

Daniel Ota. *NGVA Overview and Future Verification Activities*. Invited speaker. London, United Kingdom: Interoperable Open Architecture 2016, May 2016.

Daniel Ota. *NGVA Verification and Validation*. Invited speaker. Salisbury, United Kingdom: NATO Land Capability Group LE 2017 Spring Meeting, Mar. 2017.

Workshop/Presentations

During the preparation of the NGVA Verification and Validation approach discussed in chapter 3, several workshops and presentations were conducted by the author in order to receive valuable stakeholder input and feedback from governments and industry:

Daniel Ota. *Proposed NGVA Certification Process*. NGVA Working Group Meeting, 25^{th}-26^{th} Feb. 2014, Stockholm, Sweden.

Daniel Ota. *LAVOSAR System Acceptance Framework*. Dissemination Workshop on Land Vehicles with Open System Architecture (LAVOSAR), 3^{rd} June 2014, Brussels, Belgium.

Daniel Ota. *Verification and Validation Procedures for Future Military Vehicles*. NGVA Working Group Meeting, 16^{th}-17^{th} Sept. 2014, Prague, Czech Rebublic.

Daniel Ota. *NGVA Verification and Validation*. NGVA Publication Workshop, 11^{th}-13^{th} Mar. 2015, Brighton, United Kingdom.

Daniel Ota. *Final Review of NGVA Verification and Validation AEP*. MILVA Plenary Meeting, 30^{th} Sept.-2^{nd} Oct. 2015, Versailles, France.

Daniel Ota. *Approval of NGVA Verification and Validation AEP*. NGVA AEP Final Meeting, 1^{st}-4^{th} March 2016, Koblenz, Germany.

Further workshops intended to discuss and to brief the Module Maturity Level test procedures developed by the author:

Daniel Ota. *NGVA Data Model Testing Approaches*. NGVA Data Model Meeting, 18^{th}-20^{th} Jan. 2016, Gennevilliers, France.

Daniel Ota. *Model Maturity Level 4 and 5 Testing Approaches*. NGVA Data Model Meeting, 22^{nd} Nov. 2016, London, Germany.

1 Introduction

Military land vehicles at the tactical level have life cycles of several decades, but they need to keep up-to-date with latest technologies to address changing mission requirements. Due to the complexity, tight coupling, and closed nature of the systems, updates and improvements of hardware and software components are time-consuming, expensive, and only possible with deep knowledge of the sub-system interfaces and platform architecture.

Amongst the components on current military platforms, there are large numbers of sensors and effectors. These sub-systems are either not yet connected or are only linked via proprietary interfaces. If linked at all, built-in sensors and effectors can only be accessed by means of specific Command and Control information systems. Changes to Information Technology (IT)-related vehicle equipment are often feasible only by the original manufacturer. Thus, seamless integration of new components or upgrading integrated equipment is difficult and costly in terms of time and money.

In an era of asymmetric warfare, these monolithic or stove-piped systems are problematic. Due to a quickly changing nature of threats, long procurement processes may even lead to systems that are already no longer able to deal with current threats in their entirety when they are delivered. Rapid adaptability of systems is inevitable in order to have always appropriate capabilities available. Based on the needs of the next mission, a reconfiguration of the system should be possible promptly on the field in the best case. In order to achieve this, new system design processes and open system architectures have been proposed. Especially, in the Vehicle Electronics (Vetronics) domain several standardisation initiatives have started to address interoperability and exchangeability issues.

In order to standardise the interfaces of vehicle sub-systems and to enhance interoperability between them, the UK Ministry of Defence (MoD) released the first version of Defence Standard (Def Stan) 23–009 *Generic Vehicle Architecture* (GVA) in 2010 [1].

Based on this national effort, in 2011 an international initiative was started to adapt the GVA to an international North Atlantic Treaty Organization (NATO) Standardization Agreement (STANAG) [2]. The *NATO Generic Vehicle Architecture* (NGVA) provides design constraints for future land vehicle electronics concerning electronic and electrical infrastructure as well as safety aspects.

1.1 Research Challenges

Open System Architectures (OSA) potentially offer benefits related to facilitated addition of new capabilities because of easier upgradability and reduced life-cycle costs due to improved maintainability. However, the verification, validation, and certification of systems is extensive and more complex. System components and networks are no longer static and changing components may lead to altered system behaviour.

Thus, in addition to the definition of an architecture, the issue of system Verification and Validation (V&V) has to be covered in order to guarantee conformity to the specified architecture requirements. In the domain of military land vehicles, V&V concepts have so far been realised nationally only. For example, conformity assessment was already touched lightly in the original GVA. It has now to be internationally coordinated and discussed to gain acceptance in the NGVA community. Further, the existing V&V concept needs to be detailed and matured in order to address all potential requirements regarding data exchange, power distribution, and even safety aspects. Thereby, it has to be considered that the NGVA STANAG is still subject to changes, which however should not regularly require changes to the V&V concept. Thus, the concept has to balance generality versus specificity.

With respect to the NGVA STANAG, testing is mainly needed in the area of Data Infrastructure [3] and therein especially in NGVA Data Model (DM) [4] compliance. Therefore, test frameworks, addressing DM aspects are urgently needed. Since the NGVA DM is still under development, it needs to be matured prior to the implementation in actual systems. This comprises model checking in order to guarantee a non-ambiguous description and consistency among the devised interface specification artefacts.

For early de-risking of future NGVA-based system realisations, verification tools for conformance and interoperability testing need to be provided as soon as possible. To do so, investigation for all the three aspects – data model maturation, interoperability and conformance testing – can be aligned in an overarching verification framework.

After designing the framework, its effectiveness needs to be evaluated in case studies. Therein, it should be analysed how far artefacts from data model maturity testing can be re-used in practice for subsequent interface compliance testing of actual interface implementations. It could immensely reduce the test effort, if test cases from the DM maturity testing could be adapted in order to derive tests for conformance, interoperability and final acceptance testing. Similarly, adoption of test tools and test processes might accelerate the development and early accessibility of suitable interoperability test solutions.

1.2 Research Approach

Vehicle system verification is an active research area that is gaining growing interest from academia, civilian and military industry as well as governmental organisations. The thesis analyses and combines best practices of the verification domain to provide generic V&V procedures for military land vehicles using the example of NGVA. It lays out and evaluates a methodology and a test framework tailored to the verification of military Vetronics, which allows re-using test artefacts over the standard development and implementation test cycle.

First, a new V&V concept for military vehicle architectures was developed. It focuses on how to outline a detailed verification plan that can be tailored to specific NGVA systems. Therefore, it provides details on organisational verification responsibilities; verification, review and analysis methods; as well as methods for verification independence. To assess the conformity of NGVA systems, three sequentially-related compatibility levels have been developed, which facilitate the evaluation of specific system requirements in a structured manner by ordering them for verification. These levels form the basis for a verification process consisting of five steps ranging from the verification planning to the capturing of the results. The proposed V&V concept has been discussed in the NGVA V&V Group over the entire standardisation process. It has been approved by the NGVA STANAG management group, and has finally become a part of STANAG 4754 NGVA.

Then, a verification framework for Open System Architectures (OSA) based on the NGVA STANAG was designed. It supports the testing of interface specifications in early standardisation phases as well as the verification of architecture implementations in actual systems later on. It allows to analyse, verify and improve the NGVA Data Model (DM) modules and to re-use test artefacts resulting from this process for later conformance and final interoperability and acceptance testing. The framework has been implemented in a test laboratory, supporting the entire NGVA DM life cycle – from the early specification phase until the final acceptance testing of data model implementations in actual systems.

Finally, the verification framework was validated by means of a case study supporting the NGVA standardisation. Thereby, draft NGVA DM modules under development were prototypically implemented and it was analysed if the modules are fit for purpose in order to be implemented in actual systems. Afterwards, test artefacts such as formal test cases and software prototypes were checked if they could be used as input for initial conformance as well as interoperability and acceptance testing.

1.3 Thesis Layout

The rest of this thesis is organised as follows:

Chapter 2 provides background information for all aspects discussed in this thesis. This chapter gives on overview on current standardisation initiatives in the civilian and military domain. It presents well-defined V&V approaches and standards including concepts for interoperability and conformance testing and introduces NGVA concepts and principles needed in later chapters.

Chapter 3 is the first contribution chapter of the thesis. Based on available NGVA requirements, a V&V concept is deduced by adapting internationally recognised best practices and procedures to NGVA characteristics.

Chapter 4 is the second contribution chapter. It discusses a verification framework for overarching specification, conformance, and interoperability testing of military land vehicle components.

Chapter 5 is the third contribution chapter. It provides a case study showing how the verification framework designed in chapter 4 can be applied to NGVA DM modules.

Additionally, it analyses to what extent artefacts from specification phase can be re-used later on for conformance and final acceptance testing.

Chapter 6 is the final chapter. It draws conclusions from the work presented in the thesis and discusses achievements and future work.

2 Background

2.1 Introduction

For decades, the development of military land vehicle systems followed a similar approach. System capabilities were outlined and a prime contractor to deliver the system was chosen on the basis of cost and feasibility analysis. System requirements were derived from demanded capabilities and satisfying sub-systems were identified, acquired and integrated by the prime contractor. This approach led to the current situation that military vehicles are equipped with a variety of sensors and effectors, which are not yet linked or are only connected via proprietary, vendor-specific interfaces. Hence, changes and enhancements to IT-related vehicle equipment are often only possible by the original prime contractor. A seamless integration of new and heterogeneous components is currently difficult and expensive.

In an era of asymmetric warfare, these monolithic or stove-pipe systems are problematic. Due to a quickly changing nature of threats, long procurement processes may even lead to systems, which are no longer able to deal with up-to-date threats in their entirety when they are delivered. Nowadays, rapid adaptivity of systems is inevitable. Based on the needs for next mission, a reconfiguration of the system should be Ideally possible in the field.

2.2 Recent Standardisation Initiatives

In order to facilitate faster reconfiguration, new system design processes and system architectures have been proposed in the last years.

In 2004, the US Department of Defence published the *Modular Open Systems Approach to Acquisition (MOSA)* [5] as a technical and business strategy for developing new systems and for modernising existing ones. In addition to open systems efforts for the Air Force [6, 7] and in the Navy [8], MOSA led to new initiatives for the design and integration of military land vehicle sub-systems. In the US, the *Vehicular Integration for C4ISR/EW Interoperability (VICTORY)* [9] initiative was started to develop an open combat system architecture. However due to security classification, open access to information about VICTORY is very limited.

Similar initiatives have commenced in Europe. In order to standardise the interfaces of vehicle sub-systems, the UK Ministry of Defence (MoD) released the first version of the *Generic Vehicle Architecture (GVA)* in 2010. In the current GVA issue 3 [10], specifications for power supply, data distribution and data management as well as the design of controls were defined. To enhance interoperability across *North Atlantic Treaty Organization (NATO)* nations, the GVA standard has been further developed in cooperation with European partners in order to standardise it as a NATO *Standardization Agreement (STANAG)* since 2011. The work on the initial version of the *NATO Generic Vehicle Architecture (NGVA)* was completed in March 2016. Subsequently, the STANAG was ratified by the different nations and finally it was promulgated by NATO in February 2018.

Whilst open and modular system architectures potentially offer benefits related to the facilitated addition of new capabilities because of easier upgradability and reduced lifecycle costs due to improved maintainability, verification, validation, and certification of Open System Architecture (OSA) implementations is extensive and more complex. The system components are exchangeable and no longer static, which easily leads to changed system behaviour. In order to evaluate the impact and generate a comprehensible assessment, new verification approaches need to be designed.

Related, MOSA [5] states in its *Certify Conformance* principle: "Openness of systems is verified, validated, and ensured through rigorous and well-established assessment mechanisms, well-defined interface control and management, and proactive conformance testing. The program manager, in coordination with the user, should prepare validation and verification mechanisms such as conformance certification and test plans to ensure that the system and its component modules conform to the external and internal open interface standards allowing plug-and-play of modules, net-centric information exchange, and re-configuration of mission capability in response to new threats and evolving technologies[...].” Despite this statement, open comprehensive Verification and Validation approaches for military systems have not yet been published.

2.3 Verification and Validation Approaches

For the Verification and Validation of hardware and software in general, many generic standards and guidelines have been developed over the years. This section provides an overview of approaches relevant for the thesis.

With respect to V&V processes, IEEE 1012 [11] is a process standard defining specific V&V activities and related tasks. It describes the contents of the V&V plan and includes example formats. Since there is strong coupling with life cycle processes, IEEE 1012 especially points out the relationships between V&V and life cycle processes.

The description of system life cycle processes is addressed by ISO/IEC 15288 [12]. By defining the processes and associated terminology, it introduces a common framework to describe the full life cycle of human-made systems from conception to retirement. The outlined processes are applicable at all levels in the hierarchy of a system’s structure. Referring to ISO/IEC 15288, ISO/IEC/IEEE 29148 [13] gives guidelines for the execution of requirement-related processes. It details the required processes necessary for requirements engineering and gives recommendations for the format of the documentation to be produced.

For conformity assessment and certification, ISO 17000 [14] provides general terms and principles. Additionally, it describes a functional approach to conformity assessment by specifying phases and activities to be carried out.

The standards listed above are very generic to ensure their applicability to a broad range of hardware and software systems. To conduct actual V&V, they need to be tailored to and implemented for the specific domain. Thus, various approaches to assess and certify vehicle systems according to particular architectures or standards have been developed in the military as well as in the civilian domain.

In the civilian domain, the AUTomotive Open System ARchitecture (AUTOSAR) is probably the most famous automotive standard. With the release of AUTOSAR version

4.0 in 2010, conformance testing was firstly introduced into the standard [15, 16]. In particular, organisations and processes to test standard compliance were defined. However, these requirements did not prove to be effectively realisable, and with version 4.1 AUTOSAR went back to the old principle that suppliers test their products based on their own test suites. Nevertheless, an analysis on its suitability for NGVA conformance testing is reasonable since AUTOSAR's requirements are similar to the NGVA ones: Various suppliers provide vehicle sub-systems whose interoperability should be guaranteed later in the integration process.

Another civilian standardisation effort is the British One Box Single Vehicle Architecture (OBSVA) [17], which defines requirements for the electronic architecture of police vehicles and associated equipment. Similar to AUTOSAR, OBSVA addresses compliance procedures, but puts a strong focus on administrative processes. It describes in detail, the necessary steps, which have to be completed towards a compliance listing of a component and what it implies for the process if a certain stage is not passed successfully.

Further, the avionics domain is a pathfinder and driver for V&V. Based on the integrated modular avionics concept, Rushby [18] describes a concept for the modular certification of aircraft. This concept allows pre-certifying components based on assume-guarantee reasoning to use them across many different air-planes.

In the military domain, the NASA Systems Engineering Handbook [19] gives a top-level overview of the NASA systems engineering approach addressing the entire life cyle – starting with the collection of mission requirements over systems operation to its disposal. Thus, it also covers systems verification and acceptance for the aeronautics and space domain.

Dealing also with avionics, the Future Airborne Capability Environment (FACE™) Conformance Policy [20] presents processes and policies for achieving aircraft conformance certification. Besides outlining the FACE verification and certification processes, it explicitly addresses requirements for maintaining the certification when modifying subsystems.

In addition to new certification concepts, new metrics to assess the openess of systems are under consideration. MOSA [5] proposes to measure the percentage of key interfaces defined by open standards to determine the degree of system openness. Moreover, the percentage of modules that can change without major system redesign is given as an openness measure example.

For the land vehicle domain however, there are no specific verification and certification standards released yet. The current state is that system suppliers follow their own best practices and procedures [21].

With the turn to modular open system architectures, naturally, the way of verifying and certifying platforms has to be adapted due to the increased complexity in interactions between sub-systems. To address V&V in a generic way on the military land platform sub-system level, an approach on the example of the NGVA is discussed in chapter 3.

2.4 Testing of System Interfaces

Testing of system interfaces is one key aspect of the verification process since the matching of interface realisations ultimately decides if two systems are interoperable. For sub-systems on military platforms, interfaces are typically specified to achieve compatibility with respect to their physical connectors, power and data exchange aspects.

For interface testing regarding data exchange, in general two techniques are used depending on the test goals: Conformance and Interoperability Testing [22]. While Interoperability Testing aims to test the functionality of an entire system or application at the system boundary as experienced by a user, Conformance Testing addresses the correct implementation of low-level communication aspects like used protocols and data model messages.

Conformance Testing ensures that system interfaces are actually implemented as defined in the specified standard. Thus, it verifies that the system interface implementation complies to the relevant requirements of the standard. This increases the probability that different implementations of the standard will work reliably together. If systems actually interoperate with each other is verified by Interoperability Testing. It ensures that two different implementation are able to exchange data according to the specified standard. The focus lies on proving end-to-end functionality between the systems.

With respect to system interface verification, both techniques are used since they complement each other. Therefore, first conformance of a system to the specification or standard is tested. In a second step, interoperability is proven. Both are necessary, since even without conformance two implementations can be interoperable and systems implementing the same protocols and standards are not necessarily interoperable.

2.4.1 Conformance Testing

Conformance Testing (CT) has been extensively analysed and carried out for numerous established standards in many engineering fields, e.g. software engineering, electronic and electrical engineering. Conformance Testing determines if a product or system works as the standard specifies it. Therefore, each system is tested on the basis of a test suite representing the standard. Test equipment running a test suite stimulates the System Under Test (SUT) containing an Implementation Under Test (IUT), which should produce responses as specified in the standard. The test suite consists of test cases, each one testing specific requirements or options of the standard.

In the telecommunication domain, the European Telecommunications Standards Institute (ETSI) published various guidance documents on protocol conformance testing [22]. In particular regarding vehicle communication, ETSI announced a framework for Conformance and Interoperability Testing for Intelligent Transport Systems [23]. Intelligent Transport Systems are composed of different sub-systems such as vehicles, traffic lights or road signs.

In addition to car-to-car or car-to-environment communication, in-vehicle communication also needs to conform to standards and is therefore tested. For example, research has been carried out on conformance test systems for various standards like CAN [24], Flexray [25] or LIN [26]. As mentioned, Conformance Testing procedures were part of AUTOSAR specification as well [15, 16]. Moreover, conformance tests for specific AUTOSAR components have been conducted, e.g. for car lights [27]. After withdrawal of the former conformance test specification, the AUTOSAR consortium started to develop an acceptance test specification for the latest AUTOSAR release 4.2 [28] which is organised as a set of test suites. The specification provides test cases for different communication buses such as CAN, LIN or FlexRay which can be executed via a test system.

Also in the military domain, Conformance Testing of vehicle components is considered important and plays an ever-growing role. To exchange information between Vetronics components, fielded vehicles use protocols such as Military CAN Bus (MilCAN). For the purpose of MilCAN conformance testing, a certification rig associated with test processes and test cases has been developed by the VRC.

In addition, in the area of Command and Control Information Systems (C2IS), conformance test systems have been developed to test implementations of different C2IS solutions. One of these test systems is the MIP Test Reference System [29]. Similar to the MilCAN protocol, tests are conducted to evaluate to what extent a C2IS complies to the defined protocols as well as to analyse if the agreed information exchange on the operational level conducted as defined in the MIP standard.

2.4.2 Interoperability Testing

In order to ensure that systems with different implementations of a standard function together over a specific communication medium, their interoperability is tested. Interoperability Testing (IOT) is only meaningful in single-pair combinations of systems. Thus, if the interoperability of N systems has to be tested, tests for $N * (N-1)/2$ system pair combinations have to be conducted. Within those, each combination is called a System Under Test (SUT). This means in interoperability testing, a SUT is the combination of Qualified Equipment (QE) and Equipment Under Test (EUT).

As discussed Interoperability Testing mostly follows conformance testing. Thus, most of the application domains from the last section have been investigated from the interoperability perspective as well. Especially in the vehicle-to-vehicle and vehicle-to-roadside area, several activities with respect to interoperability field tests have been published [30, 31, 32] explaining the set-up and the components, which are necessary for operational field tests.

In the military domain, a Systems Integration Lab has been established at the U.S. Army Tank-Automotive Research, Development, and Engineering Command for the VICTORY standard [33]. It allows independent V&V of VICTORY sub-systems by conducting Interoperability Testing between VICTORY implementations provided by different vendors.

2.4.3 Test Frameworks

In order to conduct Conformance Testing and Interoperability Testing, a test specification containing testing architecture, a test suite and a testing process has to be systematically devised. Research in this domain led to standardised and widely accepted methodologies. In the area of Conformance Testing, ISO/IEC 9646 [34] is the most accepted methodology, which for instance is adapted in the ITU X.290 series [35] by the International Telecommunication Union (ITU). However, ISO/IEC 9646 is considered as a generic framework allowing a high degree of freedom, but giving few practical guidance for realisation.

For this reason, organisations such as the European Telecommunications Standards Institute (ETSI) picked ISO 9646 up and developed it further for Conformance Testing of specific standards and protocols, e. g. Session Initiation Protocol [36] and Internet Protocol Version 6 [37]. Closely related, ETSI recognised that Conformance Testing alone does not guarantee end-to-end compatibility and started to specify methodologies for combined conformance and interoperability testing [38]. One recent example is the framework for Conformance and Interoperability Testing for Intelligent Transport Systems [23]. Similarly, Organization for the Advancement of Structured Information Standards (OASIS) [39] defined a test framework for CT and IOT to be used for e-business XML (ebXML) testing.

2.4.4 Independent Verification and Validation

Military platform often contain sub-system that are safety-critical or of high-security nature. In these cases, Verification and Validation (V&V) by independent authorities is necessary. For deriving appropriate Independent Verification and Validation (IV&V) measures and to adopt them for specific programs, an increasing amount of research has been conducted in recent years. Michael et al. [40] from U.S. Naval Postgraduate School consider IV&V essential for detecting critical errors that developers often overlook. However, they argue that IV&V has not obtained full potential due to lack of appropriate tools and methodologies. To overcome this, they propose an assertion-oriented approach. They introduced a system reference model framework [41] for IV&V, which is composed of goal-oriented use cases and formal assertions specifying the desired behaviour of the SUT. The approach was demonstrated by means of a case study for the space flight software showing that it is technically and managerially effective.

Further, Akella and Rao [42] have analysed the costs and benefits of IV&V. They have shown that embedded V&V can reduce the system life-cycle costs by 15% to 20%, since implementation errors can be detected and resolved early. Further they provide ideas how to set up successful IV&V programs in organisations, which allow to re-use the same process, procedures, and methodologies all projects within the organisation.

2.5 NATO Generic Vehicle Architecture

The NGVA STANAG defines architecture concepts for future land Vehicle Electronics (Vetronics). These concepts are outlined in seven Allied Engineering Publication (AEP) volumes.

I. Architecture Approach

II. Power Infrastructure

III. Data Infrastructure

IV. Crew Terminal Software Architecture

V. Data Model

VI. Safety

VII. Verification and Validation

The Architecture Approach volume describes the NGVA concepts and provides essential military context [43]. The main focus of the STANAG concentrates on the vehicle's Power [44] and Data Infrastructure [3]. Thus, both AEP volumes are explained in detail in sections 2.5.1 and 2.5.2.

The infrastructure description effort is supported by further guidance AEPs. The Crew Terminal Software Architecture (CTSA) volume [45] defines the building blocks for NGVA-conformant Crew Terminal Software Applications. The Data Model (DM) volume [4] explains the Model Driven Architecture (MDA) approach used to specify the NGVA DM as well as the toolset required to produce and manage configuration control. Additionally, procedures to deal with safety [46] as well as Verification and Validation [47] have been outlined.

Since CTSA, DM and Safety are currently handled as guidance and not as specifications, these documents do not contain any detailed requirements. Nevertheless, the guidance documents will be updated and detailed for the next NGVA releases and may contain requirements in future revisions. Therefore, their potential contents have to be appropriately considered in the design of the V&V approach as described in chapter 3. The V&V approach as extended by a framework addressing the maturity testing of the NGVA DM and the verification of DM implementations in chapter 4. The framework is evaluated in chapter 5 on the basis of a specific DM module.

To provide a fair understanding of the structure and content of NGVA, the Power and Data Infrastructure volumes are briefly described in the next subsections. Both volumes contain requirements related to vehicle sub-systems at which two different types are distinguished: Compulsory Requirements and Optional Enhancements. A Compulsory Requirement (CR) specifies aspects that must be implemented in order to conform to the NGVA and to gain certification. An Optional Enhancement (OE) does not necessarily need to be implemented in order to conform to STANAG 4754. However, if such a capability is present, it needs to be implemented according to the stated specification in order to be compliant.

2.5.1 Power Infrastructure

The Power Infrastructure AEP volume specifies the power interfaces and requirements that form the NGVA Power Infrastructure. This includes the definition of physical interfaces and connectors for a voltage range up to nominal 28V DC and requirements for all components allowed to distribute and manage electrical power. The requirements comprise different levels of detail and abstraction. Basically, it describes how NGVA sub-systems are physically provided with power – in terms of connectors and their pin-out and which methods for power management have to be implemented by different sub-systems.

Table 2.1 provides examples for the nature of the power requirements. The requirements may relate to the whole platform (NGVA_POW_001), to the power sub-system itself (NGVA_POW_027) or to Vetronics sub-system connectors (NGVA_POW_008).

Table 2.1: NGVA Power Requirements (extracted from [44])

Unique ID	Type	Requirement Text
NGVA_POW_001	CR	All vehicle platforms and vehicle platform sub-systems shall conform to the requirements contained within MILSTD 1275D.
NGVA_POW_008	CR	The NGVA 28V DC 25 ampere low power connector shall be of type MIL-DTL-38999 series III Rev L Amdt (07/2009), D38999/XXαC98SA [...].
NGVA_POW_027	OE	The NGVA power [sub-system] shall inform the [vehicle crew] of the battery life remaining in hours and minutes at the current load.
NGVA_POW_032	OE	The NGVA Power Infrastructure shall provide controls to disable NGVA power outlets when running on battery only.

Also, the implications and therefore test procedures can range from checking the manufacturer statement of the correct connector (NGVA_POW_008) to functional checks (NGVA_POW_032).

2.5.2 Data Infrastructure

The Data Infrastructure AEP volume defines design constraints on the electronic interfaces forming the NGVA Data Infrastructure. The Data Infrastructure is used for the interconnection of mission or automotive sub-systems inside the vehicle. It consists of:

1. One or more Local Area Networks
2. Data Exchange Mechanism based on *Data Distribution Service (DDS)* [48] and *Data Distribution Service Interoperability (DDSI)* wire protocol [49] and the NGVA Data Model [4] with the appropriate *Quality of Service (QoS)* Profiles
3. Network Services (e.g. time synchronisation, network traffic management)
4. Physical interfaces and network connectors
5. Audio and video streaming data and control protocols (based on STANAG 4697 - PLEVID [50], extended by digital voice type specific control and codecs)

6. Gateways for NGVA external data communication, and for connection to legacy and automotive systems.

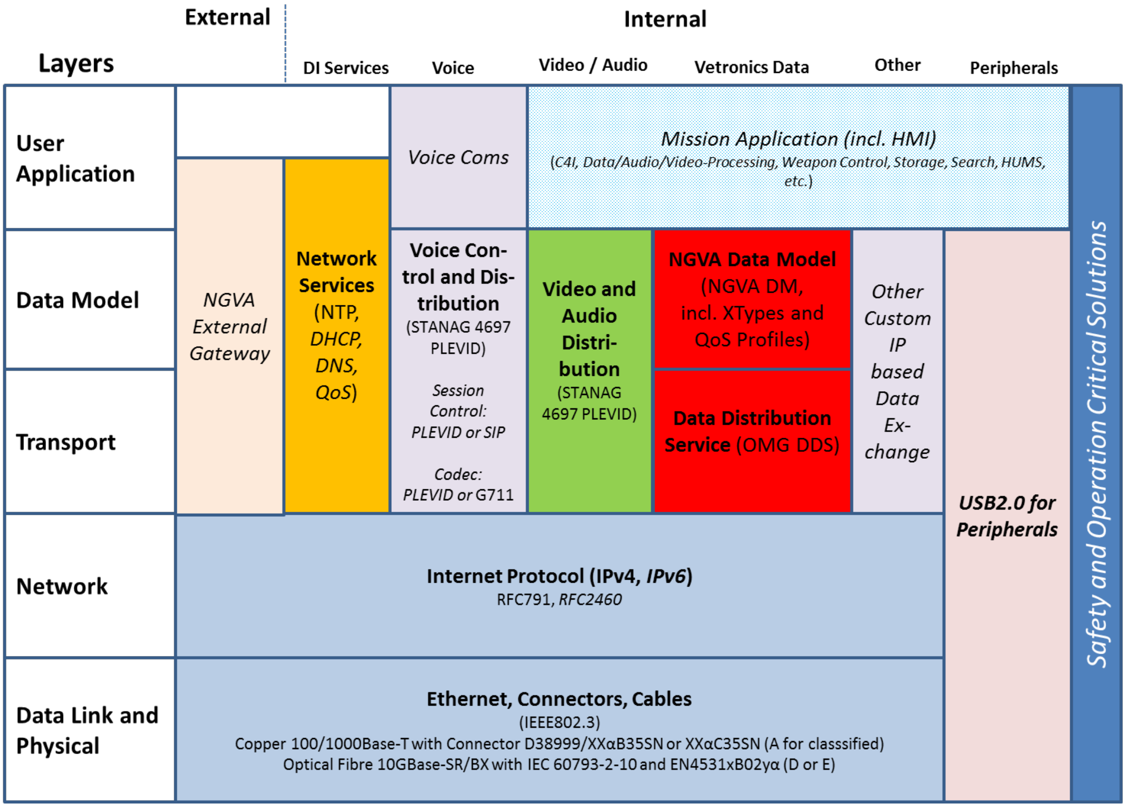


Figure 2.1: NGVA Data Infrastructure Layer

Figure 2.1 provides an overview on the electronic interfaces and protocols to be used for the information exchange among all vehicle sub-systems. The main information exchange between Vetronics sub-systems is coloured in red. It is primarily based on Data Distribution Service (DDS), which is a middleware using a publish-subscribe model to connect consumers and providers of resources or messages. The message structure is based on the NGVA Data Model [4]. From this model, standardized messages called *Topics*, are generated to be exchanged between the various Vetronics systems. These Topics define the data structures that can be published and subscribed using the primitive and user-defined data types. QoS profiles regulate the message transfer by means of specific QoS parameters that state for example that DDS communication should be reliable to ensure that all messages are delivered to subscribers of a particular Topic.

The Data Infrastructure AEP contains nearly 100 requirements specifying how vehicle sub-system data should be transmitted. As characterised for the Power Infrastruc-

Table 2.2: NGVA Data Infrastructure Requirements (extracted from [3])

Unique ID	**Type**	**Requirement Text**
NGVA_INF_002	CR	NGVA ready sub-systems shall comply with the NGVA Arbitration Protocol as defined in the NGVA Data Model.
NGVA_INF_004	CR	The NGVA network topology shall be such that the required data rates and latencies requirements can be achieved.
NGVA_INF_009	CR	Ethernet cabling and network infrastructure shall support data transfer at a minimum transmission speed of 1Gb/s.
NGVA_INF_018	OE	If DHCP is intended to be used, all switches shall be capable of DHCP Snooping.
NGVA_INF_032	CR	Vetronics Data shall be exchanged by DDS topics using the "QoS pattern" attached to it in the NGVA Data Model to assure assignment of DDS topics.

ture volume, the requirements vary in number of concerned entities, in the level of abstraction and in the verification effort needed to assure conformity. Table 2.2 gives an excerpt of five requirements. Depending on the specific requirement, it could affect nearly all Vetronics sub-systems (NGVA_INF_002) or just a particular infrastructure element (NGVA_INF_018). Verification might be simply conducted by checking of the product specification (NGVA_INF_009, NGVA_INF_018) or might imply the extensive use of software conformance test tools (NGVA_INF_002, NGVA_INF_032). In some cases the requirements are specified on a level that they are even not directly verifiable (NGVA_INF_004), but instead have to be refined by specific platform requirements depending on the actual needed platform capabilities. In the current NGVA version, requirements are not yet associated with verification methods, measures of performance or justifications. However, this is planned to change in the next versions of the AEP volumes.

2.5.3 Data Distribution Service

The NGVA information exchange is mainly based on DDS. As a standardised machine-to-machine middleware service defined by the Object Management Group (OMG) [48, 49], DDS primarily aims at systems requiring real-time information exchange. It enables scalable, real-time, robust and interoperable data exchange between nodes or applications based on a publish-subscribe model.

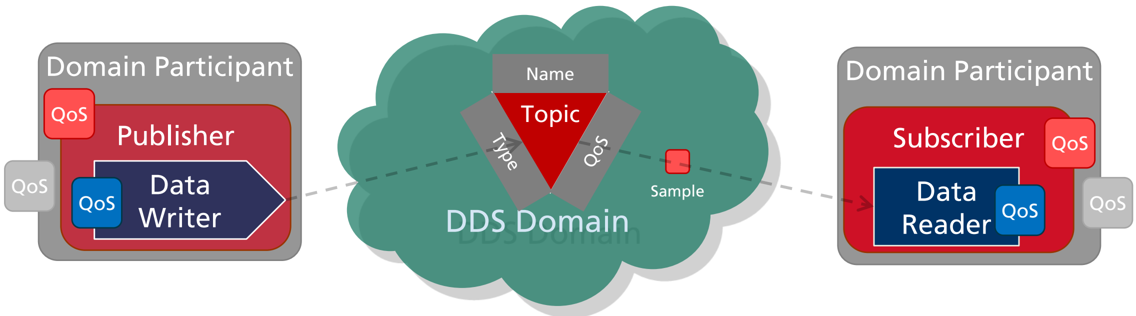
As shown in Figure 2.2a, a DDS node is identified by a unique address space, defined as a *Domain Participant*. It consists of a collection of data producers (*Publisher*) and consumers (*Subscriber*). The communication between a Publisher and a Subscriber is established if a *Subscriber* declares an interest in the data type (*Topic*) via a *Data Reader* that is offered by the *Publisher* via a *Data Writer*.

In order to initiate the information exchange, the Topic requested by the Data Reader must *match* the one offered by the Data Writer concerning *Topic Name*, *Topic Type*, and *Topic QoS*, (c.f. Figure 2.2a). In terms of name and type, matching means that both are identical: the string of the name and the Topic structure defined by the type. With respect to matching QoS, the Data Writer must offer at least the QoS that is requested by the Data Reader. If the writer offers a *reliable* communication, for example, while the reader only needs *best effort* communication, information will flow. If it is the other way around that the reader requests reliable communication, there is no match, since the QoS requirements of the reader are not fulfilled by best effort communication offered by the writer.

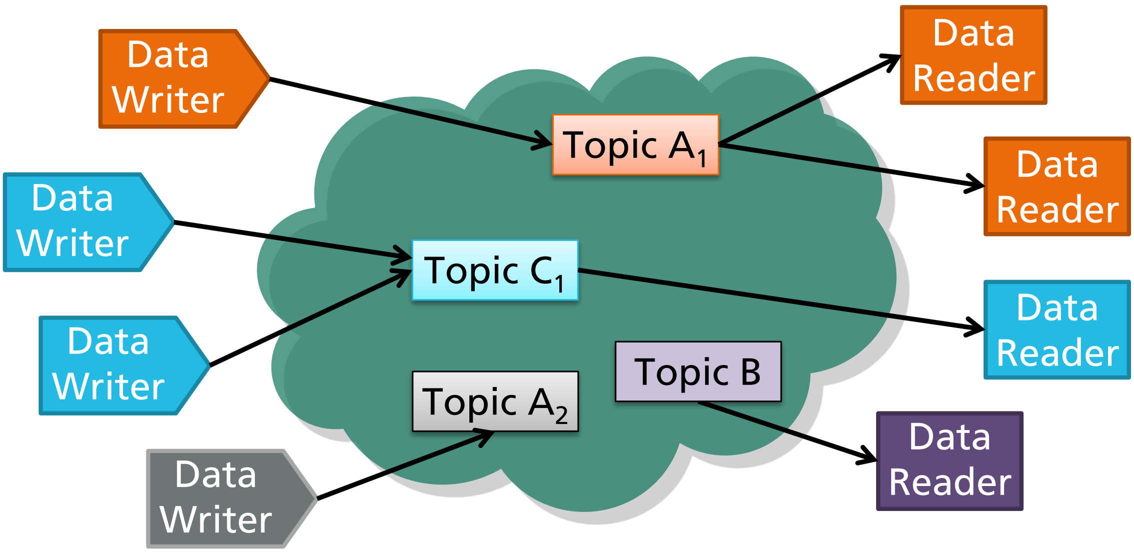
If and only if all three parameters match, *Samples* of this Topic start to flow through the DDS network on to the subscribing entities. For the sake of simplicity, a Sample is often referred to as a "message", while the Topic represents the type and structure of the message.

To represent real-world objects, the concept of an *Instance* is introduced as depicted in Figure 2.2b. Samples are updates of a particular Topic Instance. For example, military land vehicles might have two navigation systems using the same Topic to publish the vehicle's position data. To distinguish between both senders' samples, two Instances are used, each representing a message channel of a single system (cf. Topic Instances A_1 in orange and A_2 in grey in Figure 2.2b). Additionally, at the receiving side, a Sample queue is created for every Instance.

Based on their needs, applications can subscribe to either Samples of a particular Instance (depicted in orange for Topic instance A_1 in Figure 2.2b) or to those of all Instances. To create an Instance, one or more fields of the Topic are selected to form a *Key*. Thus, a Key uniquely identifies and distinguishes a Topic Instance from other Instances of the same Topic.



(a) DDS Entities and Message Exchange



(b) DDS Instance Concept

Figure 2.2: Data Distribution Service

DDS information exchange is *data-centric*. It might be the case that applications offer a Topic although there is yet no consumer for it (cf. Topic Instance A_2 in grey depicted in Figure 2.2b). Also, there can be more than one Data Writer publishing to a Topic Instance as depicted in blue. Further, it is possible to declare interest in a specific Topic even if there is no provider (cf. Topic B in purple). This allows decoupled communication of applications, since DDS takes care of existence and locations of matching entities. Once there is a match, DDS will transparently handle the message delivery without requiring intervention from the different applications.

2.5.4 Data Model

The NGVA Data Model semantically defines the intended data exchange between the different vehicle components communicating across the NGVA electronic infrastructure. It is structured in modules, at which each is describing subject matter platform domains, such as messages of a Laser Range Finder, the Navigation unit or the Brake system of a vehicle. Figure 2.3 provides an overview of the 20 modules to be released as part of the first NGVA DM baseline. As depicted the Data Model contains modules describing sensors (Tactical Sensor, Laser Range Finder, Acoustic Gunshot Detection, etc.), effectors (e.g. Tactical Effector, Single Shot Grenade Launcher, Automatic Weapon), the interface to the operator (HMI Presentation and HMI Input Devices), and generic (automotive) functionalities such as Brakes, Routes or Power.

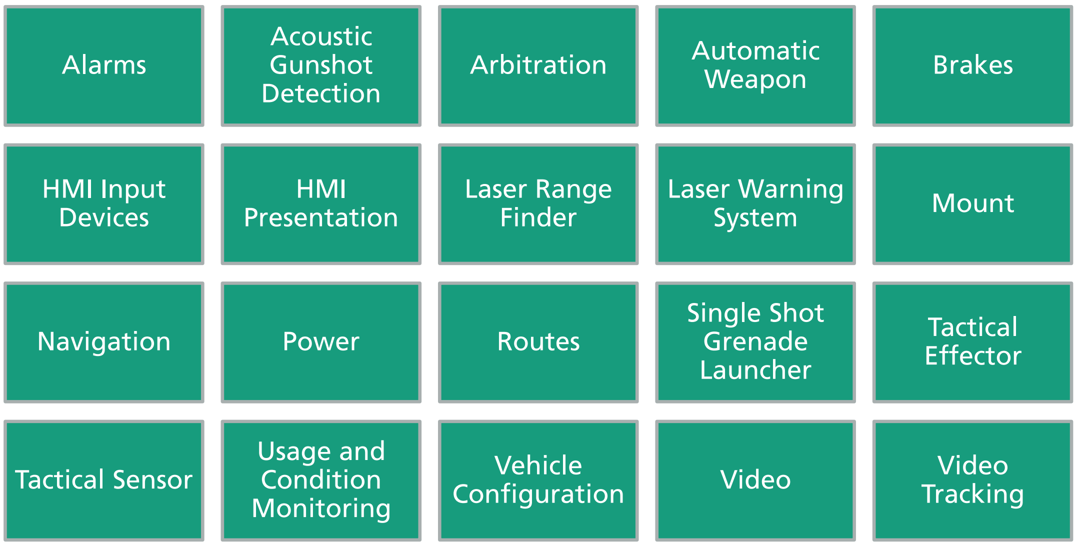


Figure 2.3: NGVA Data Model Baseline 1.0

The NGVA DM modules specify data structure definitions and the semantics for data interaction between NGVA sub-systems. The defined data structures are used by sub-systems and components in order to exchange standardised messages. Each land vehicle deployment is supposed to implement a subset of the NGVA DM modules appropriate to its requirements. Beside specifying syntax and semantics of messages, there are also artefacts in the modules, which define required behaviour such as the sequence of data exchanges or sub-system internal state changes.

Model Driven Architecture Approach

The NGVA DM expresses the system information needs in a technology independent way called a Platform Independent Model (PIM). Defined in Unified Modelling Language (UML), a PIM can be translated with a Model Driven Architecture approach to be used in actual system implementations. As one option, after transformations it can be used with DDS in order to be implemented in NGVA-based sub-systems and platforms.

Following the MDA approach as depicted in Figure 2.4, the PIM modules are transformed by means of defined rules into an interface language describing the specific messages to be exchanged. Since all NGVA-based sub-systems and platforms use DDS as a middleware, Interface Definition Language (IDL) was chosen. Each NGVA module can be separately transformed into IDL files describing the messages to be exchanged among DDS nodes. This transformation is proven and fully automated using the GVA PIM2PSM and PSM2IDL translators. In case, translations for further exchange standards are needed, for example web services, similar translations from UML to XML can be derived.

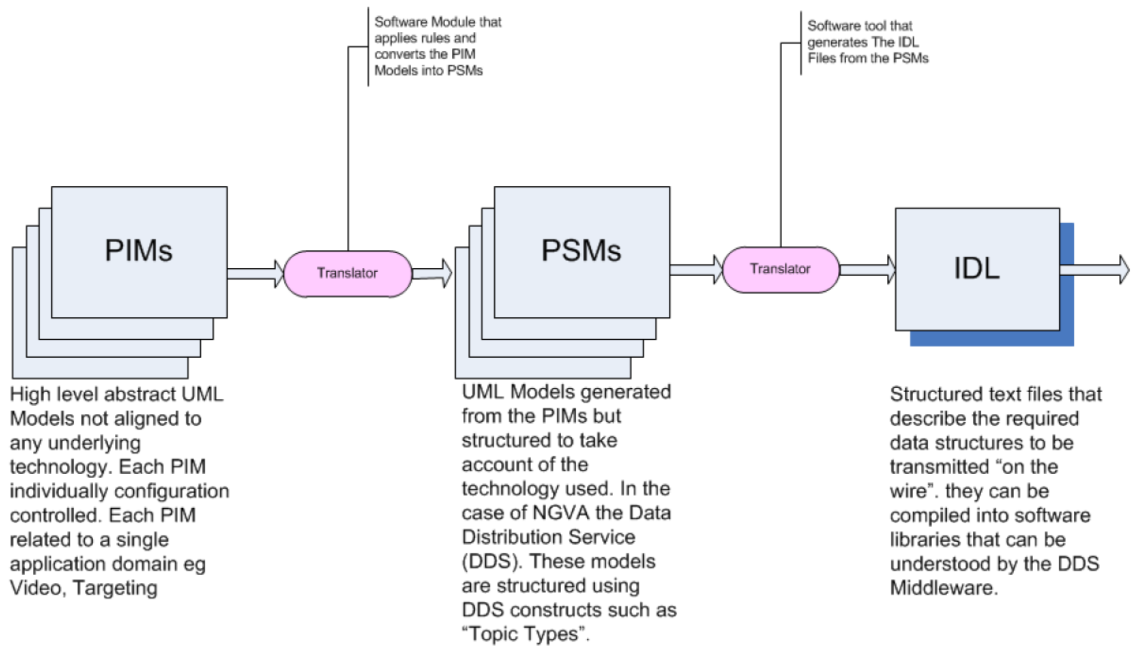


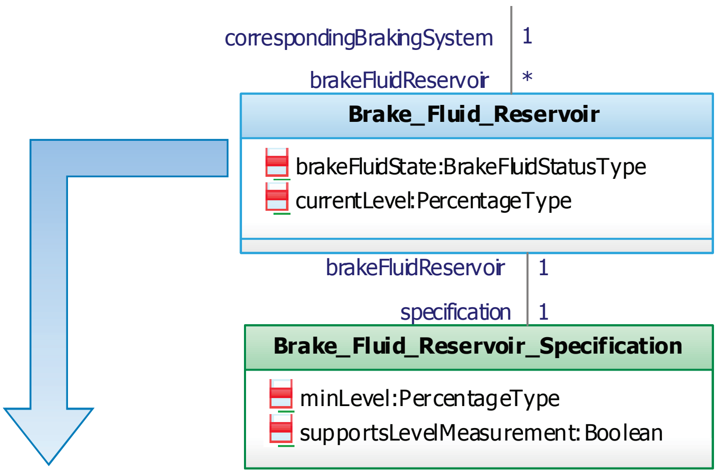
Figure 2.4: MDA Approach for NGVA DM Modules according to [4]

All NGVA DM PIM modules have to contain use cases and class diagrams, but can be optionally enhanced by state charts and sequence diagrams. The use cases specify user requirements, which are thereafter realised by classes and operations in the PIM.

After translation, classes are represented mostly by state and specification structures in the IDL while operations result in command topics. The translation process takes into account state charts and class diagrams in order to generate the IDL files. Further information such as use cases or sequence diagrams are neglected by the current GVA translator version.

Figure 2.5 illustrates the translation process on the example of two classes from the Brakes Module, which are translated to IDL code expressing DDS topics. As an example, it shows the topic *C_Brake_Fluid_Reservoir*, which is a translation from the *Brake_Fluid_Reservoir* UML class (both depicted blue). The *C_Brake_Fluid_Reservoir* topic has a member *A_currentLevel* resulting from an UML attribute and member *A_Specification_sourceID* resulting from a class association.

Thus, the PIM modules are already indirectly specifying the interface that compliant NGVA systems have to implement. The IDL files resulting from the PIM translation form an input for IUT and CT, since they contain the DDS topic definitions with their attributes as well as links to other topics.




```
struct C_Brake_Fluid_Reservoir {
    P_LDM_Common::T_IdentifierType       A_sourceID; //@key
    P_LDM_Common::T_DateTimeType         A_timeOfDataGeneration;
    T_BrakeFluidStatusType               A_brakeFluidState;
    P_LDM_Common::T_PercentageType       A_currentLevel;
    P_LDM_Common::T_IdentifierType       A_correspondingBrakingSystem_sourceID;
    P_LDM_Common::T_IdentifierType       A_specification_sourceID;
};
```

Figure 2.5: Example translation of a PIM class to IDL code

NGVA Topic Types

Basically, NGVA mainly makes use of three different types of topics to be used for the information exchange: Specification Topics, State Topics, and Command Topics.

Specification Topics describe the specific configuration of an NGVA sub-system. The class depicted green in Figure 2.5 provides a specification for a Brake Fluid Reservoir, for example, and is used to define if the reservoir supports a level measurement. If so, the Topic additionally specifies the minimum fluid level inside the reservoir before an insufficient fuel level is indicated by the Brake Fluid Reservoir State Topic.

State Topics contain information related to the current condition of a physical or logical NGVA sub-system. For example, the class depicted blue in Figure 2.5 provides information about the current brake fluid level inside a reservoir if measurements are supported by the reservoir. Further, it indicates whether the hydraulic fluid level in the reservoir is sufficient.

Command Topics are used to change the state of an NGVA sub-system. For example, for the brake system, there exists a command to apply or respectively release the parking brake. If the Boolean parameter *apply* is set to *TRUE*, the parking brake should be applied and in case of *FALSE* the parking brake should be released.

3 A Verification Concept for Land Vehicle Sub-Systems

This chapter introduces a new Verification and Validation (V&V) approach for future military land vehicles and its sub-systems. As introduced in section 2.2, many new standardisation initiatives emerged in the last years to address the issue of proprietary sub-system interfaces and missing interoperability in military land vehicles. A very promising standard in the land domain is the NATO Generic Vehicle Architecture (NGVA), which defines especially architecture concepts concerning data and power infrastructure of future land vehicle electronics (cf. section 2.5.1 and 2.5.2).

On the example of the NGVA, this chapter discusses a V&V concept allowing to verify that systems meet the requirements defined in the standard. The concept is based on an early version of the UK GVA verification approach. The chapter focuses on how to outline a detailed verification plan tailored to the specific NGVA system to define a verification process. Therefore, it provides details on organisational verification responsibilities; verification, review and analysis methods; as well as methods for verification independence. To assess the conformity of NGVA systems, three sequentially-related compatibility levels are presented, which facilitate the evaluation of the specific system requirements in a structured manner by arranging the order of their verification. These levels form the basis for a verification process consisting of five steps ranging from the verification planning to the capturing of the results.

The rest of the chapter is organised as follows: First, section 3.1 introduces a common terminology, which has been derived. Then, a refined and more detailed verification plan based on an early version of the UK GVA verification approach is presented in section 3.2, followed by the suggestion in section 3.3 to use Compatibility Levels to structure the requirements verification procedure. Section 3.4 provides a verification process consisting of five steps ranging from the verification planning to the capturing of the results, before closing in section 3.5 with a conclusion.

3.1 Terminology

The field of V&V for electronic systems has been widely explored in research over the last decades. It is strongly associated with quality management and conformity assessment. Therefore, numerous guidelines and widely recognized standards have been published over the years (cf. section 2.4). However, there is no single standard which is directly applicable to the V&V of NGVA-based (sub-)systems. As indicated in section 2.5, the sub-systems and the related requirements differ highly in complexity and abstraction. Thus, several ISO, IEC, and military standards as well as best practices were analysed and combined to form a basis especially for the NGVA Verification and Validation AEP volume [47]. This section provides the findings of the literature review related to the terminology proposed for the V&V volume.

3.1.1 Verification

With respect to NGVA, verification confirms that the requirements defined in the AEP volumes have been followed and met. This means that the characteristics and behaviour of the equipment or sub-system comply with the requirements specified in STANAG 4754, which might be refined in an additional System Requirements Document (SRD) or equivalent.

Verification is an assessment of the results of both the design/development processes and verification process carried out by a supplier, system integrator, designer or an independent assessment body. Verification is not simply testing, as testing alone cannot always show the absence of errors. It is a combination of reviews, analysis and tests based on a structured verification plan. Verification is usually performed at sub-system as well as platform level.

For the NGVA V&V volume, the standards ISO 9000 [51] and ISO/IEC 15288 [12] were consulted for the definition of Verification.

Definition (Verification). *Confirmation, through the provision of objective evidence, that specified requirements have been fulfilled. [ISO 9000:2005]. NOTE: Verification is a set of activities that compares a system or system element against the required characteristics. This may include, but is not limited to, specified requirements, design description and the system itself. [ISO/IEC/IEEE 15288]*

3.1.2 Validation

Especially with respect to military platforms, validation generates objective evidence that the capabilities enabled by the equipment or system satisfy the needs defined in the user requirements document or equivalent. Therefore, validation is an assessment to confirm that the requirements defining the intended use or application of the system have been met.

The overall intention is to build a vehicle fit for purpose that operates correctly for all the defined scenarios in the system concept of use, noting that the concept of use may change through life. Validation must also address the ability of the system to cope with various faults and failure modes.

Validation evaluates the correct operation of the complete system on specific use cases. Therefore, an operational context is needed, which varies with the particular purpose of the system. However, specifics concerning operational requirements are not part of the NGVA in the first version. Nevertheless, the compliance with overarching NGVA concepts such as openness, modularity, scalability, and availability should be validated.

In NGVA, again ISO 9000 and ISO/IEC 15288 were accessed for the definition of Validation.

Definition (Validation). *Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled. [ISO 9000:2005]. NOTE: Validation is the set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals and objectives (i.e., meet stakeholder requirements) in the intended operational environment. [ISO/IEC/IEEE 15288]*

3.1.3 Conformity Assessment and Accreditation

Verification and Validation encompasses the processes and activities of conformity assessment concerning requirements. The certification of conformity after conducting the actual V&V is very important. In order to define the nessessary terminology ISO 17000 [14] was consulted since it provides a recognized nomenclature for a accreditation chain.

Accreditation refers to the appointment of assessment bodies. Assessment bodies, for example independent institutes or military test sites, are authorized to conduct conformity assessment of NGVA (sub-)systems. Thus, accreditation is by definition different from the issue of an NGVA conformity statement.

In case of NGVA, therefore governments ratifying the STANAG have to appoint national accreditation bodies – usually governmental organisations – which have the authority to perform accreditation of NGVA conformity assessment bodies. The national accreditation bodies agree on procedures and aligned conditions to appoint conformity assessment bodies. Thereby, the formation of a network of international conformity assessment bodies is enabled where a conformity assessment body can specialise in particular verification contents (e.g. power) and is accepted by accreditation bodies from several other nations. The appointed conformity assessment bodies perform the assessment services, which include demonstration, test, analysis, inspection as well as certification.

3.2 Verification Plan

The first release of the UK GVA [1] contained a section outlining a potential *Verification Plan* for GVA-based systems. Therein, the GVA Office states that a Verification Plan shall include:

- Organisational responsibilities within the verification process
- Verification methods to be used including review and analysis methods
- Methods for verification independence, where necessary
- Description of verification tools and hardware test equipment
- Re-verification guidelines in case of system/design modifications

- Guidelines for previously developed or off-the-shelf equipment.

Due to missing level of detail, the V&V section was completely removed in the next release of the UK GVA Defence Standard.

However to support the verification process for NGVA systems, this verification plan can be considered as a sensible starting point. For this reason, the demand is picked up in this section by improving and extending the originally proposed structure. The following verification plan fractions are written in a generic way and are therefore applicable to single sub-systems or to a composition of sub-systems. Of course, the following subsections have to be adapted to the specific System Under Test.

3.2.1 Organisational Verification Responsibilities

For the development of a verification plan of an NGVA system, the different stakeholders should be defined and their responsibilities should be determined. Figure 3.1 gives an overview of potential stakeholders and their commitments for the verification of NGVA (sub-)systems:

1. The System Designer and Supplier; possibly represented by the same stakeholder. The System Supplier is responsible for the Electronic Infrastructure of the NGVA system by outlining and providing means for power distribution and data exchange between the sub-systems forming the NGVA system.

2. The Sub-System Designer and Supplier; potentially subcontractors of the System Designer. The Sub-System Suppliers are responsible for the provision of the individual sub-systems.

3. The System Integrator may be the same player as the System Supplier initially, but may change during the maintenance phase. The System Integrator delivers the complete system.

4. The Customer, e.g. the Procurement Office, typically handles the acceptance of the verification plan to ensure that it meets the initial (or refined) stakeholder requirements.

5. The Conformity Assessment Authority is often a governmental institution or independent authority providing V&V of the system.

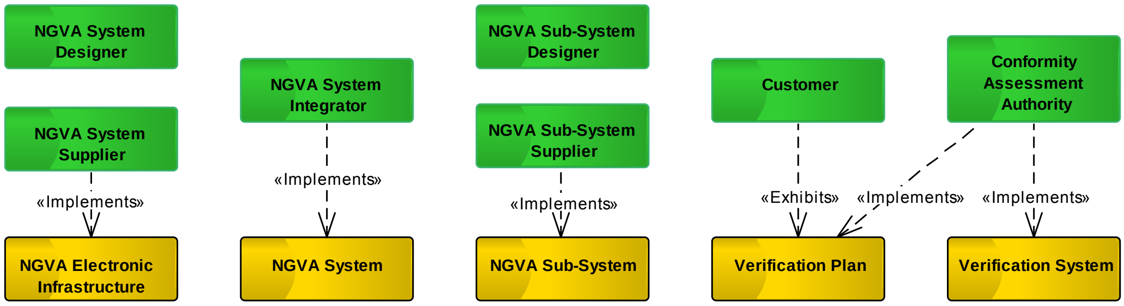


Figure 3.1: NGVA Verification Stakeholders

The stakeholder roles may change during the systems development and procurement process. Depending on the level of the verification activity, the same stakeholder or NGVA (sub-) system may have different roles. This can be illustrated by an example of a camera C which should be integrated in a surveillance unit U that in turn is mounted on a scout vehicle.

On the lowest level, the system to be verified, is the camera C itself. In this case, C is the NGVA (sub-) system and provides internally "its infrastructure"; while the camera manufacturer CM is *Designer* and *Integrator*. Thus, C and its manufacturer CM adopt the three left roles of Figure 3.1. The surveillance unit manufacturer UM is the *Customer* and may even conduct conformity assessment according to UMs requirements.

Assuming that the surveillance unit U is directly procured by the government to be deployed in several vehicles and U should be verified, U is the NGVA system that provides the infrastructure for the NGVA sub-system C. This results in CM being the *Sub-System Supplier* and UM taking the roles of *System Integrator* and *System Supplier*. Hence, the government procurement office has the role of the *Customer* and can be supported by an independent *Conformity Assessment Authority* in the verification of U.

The highest abstraction level is the integration of the components in an actual scout vehicle. Therefore, a Platform Manufacturer acts as the *System Supplier* providing the foundation with the Power and Data Distribution Infrastructure. UM from the last paragraph would provide U, being one of the sub-systems to-be integrated, as a *Sub-System Supplier*. The integration of all sub-systems on the platform is conducted by the *System Integrator*, which can be Platform Manufacturer again or a different prime contractor. The *System Integrator* is responsible to deliver the entire NGVA System to be verified. The System is checked for acceptance by the *Customer*, e.g.

the procurement office, according to the Verification Plan possibly with the help of an independent *Conformity Assessment Authority*.

3.2.2 Verification Methods

With respect to potential verification methods, the literature analysis yield ISO/IEC/IEEE 29148 [13] to be a candidate providing appropriate methods. The detailed description of verification methods in ISO/IEC/IEEE 29148 complements in a ideal way with the method definitions provided by MIL-STD-498, which were adapted for the NGVA V&V terminology.

This section gives detailed overview of the four standard verification methods in that standard: inspection, analysis or simulation, demonstration, and test. Additionally, the section indicates how they benefit the verification to obtain objective evidence that NGVA requirements have been fulfilled.

Inspection

According to [13], *Inspection* proves the item against applicable documentation to verify properties best determined by examination and observation (e.g., paint colour, weight). Inspection is generally non-destructive and typically includes the use of sight, hearing, smell, touch, and taste; simple physical manipulation; mechanical and electrical gauging; and measurement.

Regarding NGVA, Inspection is appropriate to verify requirements related to the fulfilment of other standards (cf. NGVA_POW_001 in Table 2.1), to the use appropriate connectors (cf. NGVA_POW_008, Table 2.1), or to check if the equipment supports relevant features (NGVA_INF_018 in Table 2.2).

Analysis (including modelling and simulation)

Analysis uses analytical data or simulations under defined conditions to show theoretical compliance where testing based on realistic conditions cannot be achieved or is not cost-effective. Analysis may be based on "similarity" by reviewing a similar item's prior verification and confirming that its verification status can be legitimately transferred to the present system element [13].

Many requirements related to the correct implementation of the NGVA Data Model should be proved using Analysis. Especially with respect to the verification of sub-system implementations, an analysis in a test lab with simulated counterpart equipment allows to detect issues with the correct usage of DDS Topics and QoS settings (e.g. NGVA_INF_002, NGVA_INF_032 in Table 2.2) before integration in the actual operational environment.

Demonstration

Demonstration is a qualitative exhibition of functional performance, usually accomplished with no or minimal instrumentation or test equipment [13]. Demonstration uses a set of test activities with system stimuli selected by the supplier to show that system or system element response to stimuli is suitable or to show that operators can perform their allocated functions when using the system.

Demonstration is useful to prove functional NGVA requirements. This concerns especially requirements related to informing the vehicle crew or to letting the crew members physically control a sub-system. NGVA_POW_027 and NGVA_POW_032 in Table 2.1 are two examples for these requirements.

Test

A *Test* quantitatively verifies the operability, supportability, or performance capability of an item when subjected to controlled conditions that are real or simulated. These verifications often use special test equipment or instrumentation to obtain very accurate quantitative data for analysis [13].

Tests are especially needed to ensure that vehicle infrastructure provides the required latencies and data rates in the interaction of all vehicle systems. For example, referring to NGVA_INF_004 in Table 2.2, evaluating the network load under operational conditions with network tools allows to detect possible bottleneck in the vehicle architecture.

3.2.3 Review Methods

Throughout the verification process, formal system reviews and audits should be performed at different phases of the verification, e.g. Test Readiness Reviews [52, Section 3.6]. The verification plan should include necessary reviews as well as corresponding review methods.

E.g., these reviews should ensure that all relevant NGVA requirements for specific system are captured by the verification plan, appropriate verification methods are used, and verification is conducted properly. Therefore, check-lists or other aids should be used.

3.2.4 Analysis Methods

The verification plan should include means to assure traceability and coverage analysis of requirements. All requirements must be traceable to an implementation/realisation in the system. This allows comprehensive proof that all relevant requirements are fulfilled.

Additionally, provisions to link requirements and verification activities or test cases should be described in the verification plan. Therefore, a requirements traceability matrix, also known as requirements coverage matrix, can be used.

3.2.5 Verification Tools and Techniques

Usually, hardware and software tools are used to assist and automate verification processes. For example, software tools supporting test coverage analysis and regression testing. The verification plan should include guidelines for these tools and any hardware test equipment. This includes detailed description of needed tools, explanations of each tool's performance, required inputs and outputs generated.

Additionally, the verification plan should address test facilities and integration and system test laboratories supporting the verification effort, e.g. specific conformance test systems or interoperability test labs.

Conformance and Interoperability Tests

The main objective of NGVA is the assurance of interoperability between NGVA (sub-) systems. To evaluate system conformity to standards in this vein, typically Conformance and Interoperability Testing are used. Both techniques are complementary; often, Conformance Testing addresses protocols and lower-layer communication aspects while Interoperability Testing is used for entire systems and applications.

According to [22, Section 4.2], Conformance Testing is conducted by a *Test System* which stimulates a *System under Test*. This *System under Test* often contains an *Implementation under Test*, which is subject to Conformance Testing (cf. Figure 3.2). Conformance Testing is a formal process, deterministic and repeatable. It ensures that a system meets a defined set of requirements, for example, a correctly implemented protocol stack. This is especially crucial for data exchange specified in the NGVA Data Infrastructure and Data Model.

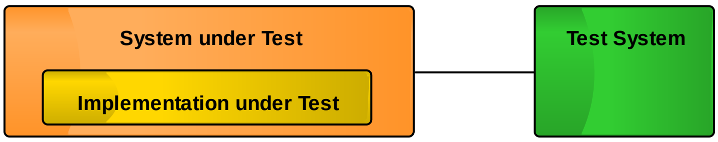


Figure 3.2: Conformance Testing (adapted from [22, Section 4.2])

In contrast to CT, Interoperability Testing [22, Section 4.1] is performed at system interfaces, which offer only normal user control and observation. Therefore, it is based on functionality as experienced by a user and does not necessarily check how the information exchange is realised at the protocol level. The purpose of Interoperability Testing is to prove that end-to-end functionality between at least two NGVA (sub-) systems, i.e. the *Equipment under Test* and the *Qualified Equipment* (cf. Figure 3.3), is realised as defined as in the NGVA STANAG.

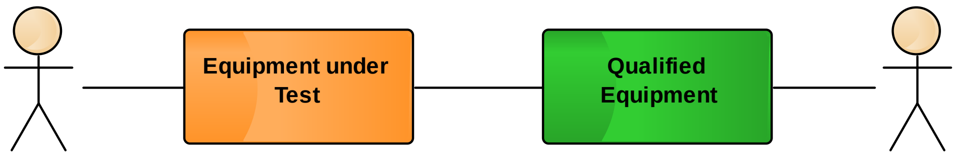


Figure 3.3: Interoperability Testing (adapted from [22, Section 4.1])

Interoperability Testing as well as Conformance Testing should be tackled in the NGVA verification process. Whilst CT guarantees adherence to the NGVA specification by testing against a reference system, IOT guarantees end-to-end interoperability between two different system implementations. Therefore, the verification plan should describe, which of the following tests should be conducted.

NGVA Data Model Conformance Testing

One verification key aspect should be NGVA Data Model Conformance Testing. This evaluation may be conducted by independent conformity assessment bodies providing appropriate test systems potentially over Virtual Private Networks (VPN) or other secured internet connections.

These Conformance Test Systems have to verify the NGVA DM conformity of NGVA systems. Therefore, NGVA sub-systems as *Systems under Test* are considered as black boxes and system specific test suites to evaluate the system response for valid, inopportune and invalid input are run.

These formalized test suites should support an automatic execution of test cases as well as an automatic and unbiased assignment of test verdicts. Centrally maintained NGVA DM Conformance Test Systems have the advantage that all (sub-) system vendors can always access the latest release of the test suite. However, restricted by spatial distribution, they cannot reflect a real vehicle bus and are not suitable for real-time testing.

Test Laboratories and Test Beds

For overarching conformance and interoperability tests, vendors as well as vendor-independent authorities should maintain test beds to conduct tests prior to the initial release of products or upgrades. These test beds allow a collocated testing verifying that all real-time, safety, and security requirements are met.

In particular, the test beds should provide the infrastructure to which NGVA systems, the *Equipment under Test*, have to be interoperable in order to be verified. Therefore, the test beds may consist of components that can control and request data from the *Equipment under Test* or else gateway components might be necessary.

Demonstrators and Experiments

Especially for the confirmation of functional and operational requirements, demonstrators and experiments should be used. They can be used for verification as well as validation to prove the intended use of the system. Thereby, the defined concept of use of the system is validated in predefined operational scenarios.

3.2.6 Verification Independence

Verification by independent authorities may be necessary for requirements that are safety-critical or of high-security nature. This is currently not highly relevant for NGVA since the Safety AEP volume does not contain requirements but this may change in the next release of the NGVA STANAG. Additionally, a separate Security volume is under development. Therefore, the verification plan should include provisions to take an appropriate amount of independence into account.

According to IEEE Std 1012-2012 [11], Independent Verification and Validation (IV&V) is defined by three parameters:

1. Technical Independence: Requires effort to use personnel who are not involved in the development of the system or its elements.
2. Managerial Independence: Requires that the responsibility for the IV&V effort be vested in an organisation separate from the development and program management organisations.
3. Financial Independence: Requires that control of the IV&V budget be vested in an independent organisation.

Depending on the complexity of the NGVA system to be verified, different forms of independence have to be adopted for a verification organisation. The five most prevalent are *Classical*, *Modified*, *Integrated*, *Internal*, and *Embedded* IV&V [11]. The verification plan should state the appropriate form for the addressed NGVA system.

Classical IV&V

Classical IV&V embodies all three independence parameters. It is generally required for Safety Integrity Level (SIL) 4 (i.e., loss of life, loss of mission, significant social loss, or financial loss) through regulations and standards imposed on the system development.

Modified IV&V (No managerial independence)

Modified IV&V is used in many large programs where the system prime integrator is selected to manage the entire system development including the IV&V. Because the prime integrator performs all or some of the development, the managerial independence is compromised by having the IV&V effort report to the prime integrator. Modified IV&V effort is appropriate for systems with SIL 3.

Integrated IV&V (No technical independence)

This type is focused on providing rapid feedback of V&V results into the development process and is performed by an organisation that is financially and managerially independent of the development organisation to minimise compromises with respect to independence. The rapid feedback of V&V results into the development process is facilitated by the integrated IV&V organisation but has a potential impact on technical independence.

Internal IV&V

Internal IV&V exists when the developer conducts the IV&V with personnel from within its own organisation, although preferably not the same personnel involved directly in the development effort. Technical, managerial, and financial independence are compromised. This form of IV&V is used when the degree of independence is not explicitly stated and the benefits of pre-existing staff knowledge outweigh the benefits of objectivity. This approach is preferable as long as the NGVA (sub-) system under development has no safety obligations.

Embedded V&V

This type is similar to internal IV&V. The embedded V&V organisation however works side by side with the development organisation and attends the same inspections, walk-throughs, and reviews as the development staff. Embedded V&V is not specifically tasked to assess independently the original solution or conduct independent tests. The focus is on ensuring conformance to the development procedures and processes. This results in minimal technical, management as well as financial independence.

3.2.7 Re-Verification Guidelines

After modifications of design or implementation, NGVA equipment has to be reverified. Depending on the level of change, the complete system may need to be reverified. Thus, the verification plan should describe re-verification guidelines depending on the type and level of (sub-) system changes. If there are no guidelines given, the complete system needs to perform the full verification process.

3.2.8 Legacy Equipment Guidelines

For any previously developed or off-the-shelf equipment, a description of the methods to satisfy the objectives of the NGVA STANAG should be given. These methods can incorporate the development of software and hardware adapters as well as descriptions for dealing with safety and power issues. In addition, a roadmap for a long-term NGVA adoption should to be outlined. If there are no descriptions given, all legacy and off-the-shelf equipments are treated as a genuine NGVA system.

3.3 NGVA Compatibility Level

As described in section 2.5, NGVA requirements vary a lot in the level of abstraction and in their impact on the (sub-) System Under Test and the related vehicle infrastructure. Since NGVA is a fairly new specification, not all future vehicle sub-systems may address all NGVA requirements from the beginning. Thus, this section proposes so-called NGVA Compatibility Levels that allow to verify the different system requirements in a structured manner by arranging the order of verification. In addition, the Comparability Levels permit to certify NGVA conformity up to a certain level.

Therefore, an incremental certification is proposed based on three NGVA Compatibility Levels (NCL): Connectivity Compatibility, Communication Compatibility, and Functional Compatibility. As indicated in Figure 3.4, these levels are sequentially-related: Communication Compatibility includes Connectivity Compatibility and Functional Compatibility is based on and includes the previous levels.

Figure 3.4: NGVA Compatibility Levels

3.3.1 Connectivity Compatibility

The first level, Connectivity Compatibility ensures that the (sub-) system can be physically integrated into the NGVA architecture without any negative impacts to existing NGVA components. Physical power and network interfaces comply with the requirements of Power and Data Infrastructures AEP volumes [44, 3].

Thus, this level applies to requirements that concern the electrical and physical specifications of the connectors as well as low level means to transfer data between NGVA (sub-) systems, e.g. OSI Layer 1-4 protocols. Additionally, the first level contains requirements that may compromise other services; for example, requirements that are related to Electromagnetic Compatibility (EMC), safety and power supply. These first level requirements are mainly verified by physical inspection or testing. In some cases like EMC, even the inspection of conformity statements from vendors may be sufficient.

3.3.2 Communication Compatibility

If applicable to a (sub-) system, Communication Compatibility refers to the correct implementation of the NGVA Data Model and video streaming standards. On the basis of achieved Connectivity Compatibility, data interfaces (e.g. Data Distribution Service, Video/Audio Protocols) and associated NGVA DM implementation (e.g. Topic Types, Quality of Service) need to comply with the NGVA Data Model AEP volume [4].

Based on requirements stating the services provided by the system and the NGVA-related sub-systems are integrated into the system, relevant parts of the NGVA Data Model covered by the equipment are derived and tested. These tests cover the systems data exchange specified in the NGVA Data Model, e.g. correct publishing of specification topics and correct response to mode changes.

3.3.3 Functional Compatibility

Underpinned by Communication Compatibility, Functional Compatibility evaluation ensures that data flows conform to data exchange, performance and specific functional requirements. Concerning data exchange, NGVA DM tests covering the system or component response for valid, inopportune and invalid inputs are conducted. This includes tests for the publishing of correct information and data format (e.g. for the current GPS position) and the proper behaviour for commands (for example mount movements).

Additionally, this level should be evaluated if real-time and bandwidth requirements are met and specific functional requirements e.g. regarding security are fulfilled. If further operational requirements are provided, they should be tested here as well.

3.4 Verification Process

The actual evidence that NGVA (sub-) systems fulfil the requirements specified in the NGVA STANAG is established in the *Verification Process*. Thus, the following subsections propose an NGVA Verification Process consisting of five steps based on the NASA Systems Engineering Handbook [19, Section 5.3]. Typically, the Verification Process is performed by the developer that realises the NGVA end-system with participation of the end users and independent conformity assessment bodies. The Verification Plan of section 3.2 and the System Requirements Document (SRD) of the system under test are the key inputs for the Verification Process.

3.4.1 Verification Planning

Planning of the verification process is a first key step. Based on the SRD and the requirements of the NGVA AEP volumes, the specific requirements are collected and verification types (e.g., analysis, demonstration, inspection or test) for them are established. Additionally, the verification plan should be reviewed for any specific procedures, constraints or further measures that have to be considered prior to the actual verification.

3.4.2 Verification Preparation

In preparation for verification, the system requirements are reviewed, confirmed, and allocated to the different NCLs. The NGVA system to be verified is acquired, as well as any enabling products and support resources that are necessary for verification. The verification preparation includes the verification environment. For Connectivity Compatibility this may cover tools or measuring devices for a particular pin-out. In case of Communication Compatibility, an account for the NGVA Data Model Conformance Test System has to be requested or further measures to connect to the system have to be considered. To test Functional Compatibility simulations may have to be prepared. The particular measures depend on the specific system requirements.

3.4.3 Verification Performance

In this step, the verification of NGVA systems is conducted and conformity to each relevant verification requirement is tested. Therefore, the responsible stakeholder should ensure that the procedures are followed and performed as specified in the verification plan and the data is collected and recorded for verification analysis. In this phase, the tests for the three NCLs are conducted in sequential order from Connectivity over Communication to Functional Compatibility. The different test procedures and outcomes are linked to the requirements by appropriate means, e.g. a requirements traceability matrix.

3.4.4 Verification Outcomes Analysis

Once the verification activities have been completed, the collected results are analysed, in particular for quality and correctness. Based on this analysis and possible defects, it could be necessary to re-realise the system or to re-engineer the sub-systems assembled and integrated into the system verified and to re-perform the NGVA verification process. Additionally, verification test outcomes can be unsatisfactory for other reasons, including poor conduct of the verification process (e.g., procedures not properly followed, use of un-/mis-calibrated equipment, etc.). This would cause re-performing of the affected verification steps, as well.

3.4.5 Capturing of Verification Results

In the last step, verification results shall be produced from the verification process activities. The verification results should:

- Identify the verified system including its configuration or version number
- State verifier and verification date
- Specify the used tools including their configuration and version numbers
- Indicate each procedure that passed or failed during the activities
- Contain any corrective action taken and the lessons learned (including feedback to improve the NGVA specification)
- Include a traceability analysis
- Capture the final pass/fail results for each requirement
- Document proof that the realised system did (not) satisfy the requirements
- Include conclusions and recommendations for further verification activities
- Mention consequences for the validation of the system.

3.5 Conclusion

This chapter provided guidance on the verification of systems designed according to the NGVA STANAG. The approach is intentionally kept generic in order to deal with any type of NGVA (sub-)system and take future AEP volume changes into account.

Based on a common terminology, a detailed verification plan was introduced. To facilitate the conformity assessment of NGVA systems, three sequentially-related NCLs have been developed, which allow the evaluation of the requirements in a structured manner. Additionally, a verification process consisting of five steps from planning, over preparing, conducting, and analysing to the capturing of the verification results has been proposed.

The presented verification concept has been discussed with the NGVA community over the entire development cycle and was accepted by all international stakeholders. During this time, the author was the lead of the NGVA V&V Working Group and was therefore responsible for drafting the AEP document. The final AEP version is now officially part of the NGVA STANAG as the NGVA Verification and Validation Volume [47].

4 A Test Framework for Vehicle Sub-Systems

In the civilian domain, the number of electronic components in vehicles is rapidely increasing [53]. The same progress is already foreseeable for military vehicles in the future and due to their modularity especially for NGVA-based ones. More and more Vetronics components are networked and the complexity of interactions between them is rapidly increasing.

Obviously, interoperable communication is needed in order to realise the cooperation of components and the V&V concept as discussed in chapter 3 provides the basis to ensure interoperability. However, the protocols and standards defining the data exchange among Vetronics components are typically specified in natural language. This may lead to a different understanding and divergent implementations of the same specification since in many cases those specifications are ambiguous and not precise. As a consequence, Vetronics components may not be able to exchange data properly although they are implementing the same specification.

This issue has already been identified and an approach to tackle it for NGVA was presented in the previous chapter, but so far all considerations are at the conceptual level. At the moment, no tools for NGVA interface verification are designed or implemented though.

The following sections provide an overview of established test approaches and describe the procedure to derive specific test suites and test architectures for vehicle sub-system testing. First, a motivation for a test framework supporting early de-risking in the Vetronics development cycle is given by analysing the benefits and costs of early testing. Based on well-established methods and procedures, the actual test framework addressing specification, conformance and interoperability testing of future military land vehicles is provided in section 4.5, afterwards.

4.1 Benefits and Costs of Testing

The main objective of testing is to gain a higher degree of confidence that a system works properly for specified use cases. Testing of complex distributed systems like modern vehicle implementations can never be exhaustive. Since it can only be tested during a restricted period of time, testing cannot ensure complete correctness of an implementation. Testing only shows the presence of errors, not the absence of errors.

Testing is traditionally done near the end of development phase of the platforms [54]. Vendors assume that they have implemented the specification correctly and apply for certification. However, being under pressure to get the product into the market or, even worse, being typically near the development budget limit is not a good precondition if testing fails due to a wrongly interpreted specification. At this stage, changes are orders of magnitude more expensive in terms of time and costs.

Ideally, testing is carried out through-out the entire product development process. Once additional requirements of the specification have been realised in a development step, their correct implementation should be tested. This raises the probability that the final product passes conformance testing entirely. For this purpose, the vendor needs early access to conformance test tools to incorporate them into the product development process. Open and unlimited access to test suites for the product developer already in the implementation phase reduces the risks posed by traditional conformance testing of the final product. Additionally, it reduces the costs that vendors have to incur in developing their own testing tools since it offers the possibility to test the product against a carefully designed and widely accepted set of test cases. By doing so, early confidence for future customers is generated and the procurement process might be simplified on both sides.

Further, early development of test tools facilitates the standardisation and the acceptance of the standards itself. If Conformance Testing is linked with the standardisation process at an early stage, it allows fast improvements of the specification and can provide valuable feedback to the standard authors. Especially in an early phase of the standard life-cycle, confidence in the standard is given to vendors at an early phase of the product life-cycle. Later on, the matured test tools can serve as widely accepted Reference Implementations.

Besides the benefits, some costs of course should be considered. First, test suites have to be designed and developed, which contain test cases describing objectives, needed inputs and desired outputs of tests. The younger and more immature a specification is, the higher is the development effort, since there is neither much experience with implementing the specification nor there are already existing results, which be be can be re-used. Second, tools to conduct the tests, evaluate the test runs and generate reports have to be developed. Third, the actual effort of executing the tests during and after product implementation needs to be considered as well.

If Conformance Testing is started already during the development phase, the testing activities in those early phases tends to be more frequent but less comprehensive than in the end of the development cycle since more and more requirements of the specification should be properly implemented.

4.2 Test Approaches

In order to analyse the interface compatibility of electronics systems, two forms of testing have been established over the last decades: Interoperability Testing (IOT) and Conformance Testing (CT). For both approaches exist different definitions, which however have a common meaning. Conformance Testing determines if an implementation complies with the requirements of a specification or standard while Interoperability Testing (IOT) analyses if two implementations are actually compatible and work reliably together. For the purpose of this chapter the following definitions by ETSI [55] have been chosen.

Definition (Conformance Testing)**.** *Process of verifying that a single implementation conforms to the individual requirements of one or more standards or specifications or profiles.*

Definition (Interoperability Testing)**.** *Process for verifying that several implementations can interoperate while conforming to one or more standards or specifications or profiles.*

Since Conformance Testing verifies that a single implementation conforms to the defined standard, it is often seen as first step towards interoperability with other systems compliant to the specification. Testing the conformance of two implementations to a specification leads to higher degree of confidence that both implementations will interoperate. Further it allows different vendors to implement the standard independently of each other with a high probability that the implementations are actually interoperable.

Since conformance does not ensure comprehensive interoperability, typically, Interoperability Testing is conducted afterwards to test if two implementations are actually able to interact based on defined information exchange scenarios. CT improves the probability of interoperability while IOT checks if interoperability has been achieved at the user level. IOT is more costly due to expenditure of time and involvement of personnel. However, experience shows that only a combination of IOT and CT can guarantee a flawless information exchange among implementations. Both techniques are complementary, neither IOT can substitute CT nor vice versa since implementations conforming to a standard may not be interoperable and interoperable implementations may not conform to the standard.

4.2.1 Conformance Testing

As briefly discussed in section 3.2.5, Conformance testing is conducted by means of a Test System, which stimulates a System Under Test (SUT) with valid, inopportune and invalid inputs and evaluates the response of the SUT. The SUT often contains an Implementation Under Test (IUT) (cf. Figure 4.1) implementing the actual specification.

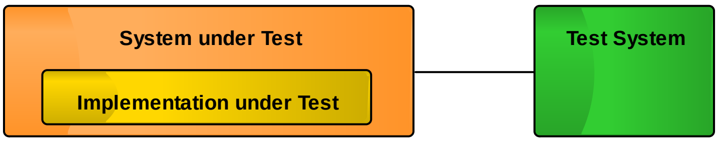


Figure 4.1: Conformance Testing (adapted from [22, Section 4.2])

CT assesses to what extent the SUT/IUT fulfils the requirements specified in the standard. It is a formal process, deterministic and repeatable, and ensures that a system meets a defined set of requirements. This includes for example, testing if the protocol stack is correctly implemented, verifying the protocol message structure and content, as well as the defined sequences of the message exchange. Thus, it aims to gain confidence in the correct operation of the implementation and increases the probability that the implementation will communicate successfully with other implementations but it cannot guarantee it.

4.2.2 Interoperability Testing

In contrast to Conformance Testing, Interoperability Testing is performed at system interfaces, which offer only normal user control and observation (cf. Figure 4.2). For this purpose, it is based on the functionality as experienced by a user and not necessarily as specified at the protocol level.

The purpose of interoperability testing is to prove that end-to-end functionality between at least two (sub-) systems, the Equipment Under Test (EUT) and the Qualified Equipment (QE), is accomplished as defined in the underlying specification or standard. It determines the ability of the Equipment Under Test (EUT) to provide the defined functionality between itself and another QE to which it is connected to.

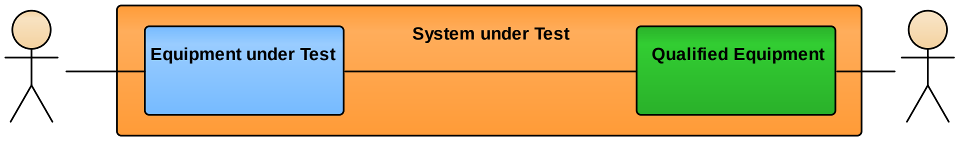


Figure 4.2: Interoperability Testing (adapted from [22, Section 4.1])

4.2.3 Combining Conformance Testing and Interoperability Testing

As stated, both CT and IOT cannot substitute each other. Rather both approaches are complementary. This section highlights the advantages and disadvantages of CT and IOT and draws conclusions how they can be combined.

By definition, Conformance Testing foremost aims at testing if a system or an implementation adheres to the requirements of a specification. CT has to test all specified requirements, but is also limited to test requirements from the specification only. Interoperability Testing in contrast is in general less strict and formal in testing *all* the requirements of the specification. Since IOT is not limited to the standard and generally focuses on the correct end-to-end-functionality of both systems, the tested requirements are more user-centric. Thereby, issues not addressed by the standard, which however prevent interoperability can be detected.

Conformance Testing has the advantage that problems in the implementation are easier to detect because there is only one SUT with one IUT involved and the conformance test system is generally considered as a Reference Implementation. Issues identified during IOT can be caused by the first, second or even both implementations.

The same reasoning influences the test effort, which is linear in case of CT for multiple systems. If n systems should be tested for conformance to the specification, all n have to pass all test cases once. In case of IOT, $n*(n-1)/2$ combinations have to be verified, since every system has to be tested with all other $n-1$ systems resulting in a quadratic testing effort.

In order to reduce the disadvantages and to increase the benefits of CT and IOT, combining both testing approaches becomes implicit. For this purpose, the configuration of Figure 4.2 is extended by Protocol Analysis Equipment. The resulting set-up was proposed in [22, Section 4.3]) and is depicted in Figure 4.3.

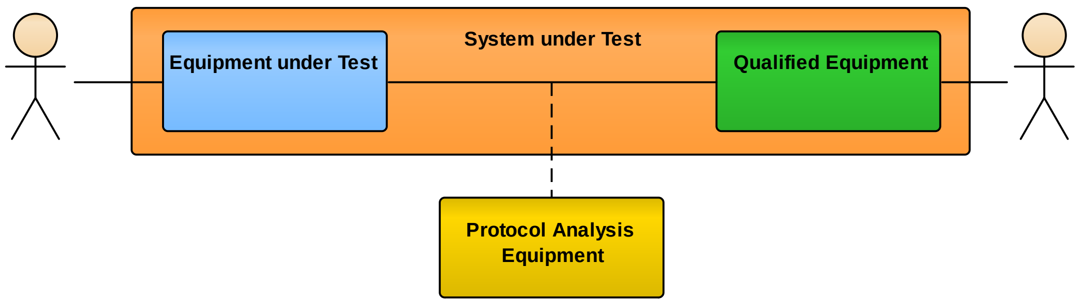


Figure 4.3: Interoperability Testing with Conformance Analysis (adapted from [22, Section 4.3])

Of course, the combined testing describes a compromise, which does not provide the depth and breadth of separated CT and IOT, but it allows deriving combined results. The configuration can be seen best as extensive IOT enriched with limited protocol conformance monitoring. The examination of the protocol analysis output allows to judge and to capture if the message exchange between EUT and QE conforms to the specification throughout IOT. If the Protocol Analysis Equipment is not only capable of monitoring, but even supports the injection of messages, it can be shown that each interoperability test case executed between the QE and EUT can be properly divided into two different conformance test cases [56].

4.3 Test Framework Development

In order to facilitate early testing and to reduce the test effort during the development of the specification as well as of actual systems, an integrated approach considering their entire development cycles is needed. Furthermore, such an approach promotes the acceptance and applicability of new standards and open architectures.

Different forms of specification-related testing are usually conducted within a standard life-cycle. In the draft phase, the specification itself is subject to testing since it needs to be assured that it is comprehensive, consistent and unambiguous. Later, systems are developed following the specification. In this phase, these systems are subject to testing, since proof has to be provided that the specified requirements are properly realised. For this purpose, conformance tests are carried out for verifying the system behaviour for valid, inopportune and erroneous input. Additionally for final acceptance, the system is analysed in combination with already established systems in order to prove its capabilities and interoperability in a real environment.

Therefore, well-defined sets of test cases are developed under consideration of the verbal specification. To handle ambiguous, incorrect or even contradictory requirements, test cases are developed in consensus with competent partners and the standardisation body or the corresponding organisation responsible for the standard. In order to allow an automatic execution to reduce time and costs and to facilitate the repeatability of the tests independently of time and place, networked and distributed testing approaches are used to carry out the test cases.

4.3.1 Test Specification

Conformance Testing as well as Interoperability Testing is carried out based on a test specification containing a comprehensive and structured suite of tests.

For deriving an integrated test framework for vehicle sub-systems based on NGVA, first, general procedures for the development of a test specification have been analysed. It was noticed that the basic approach for CT and IOT is similar, however there are differences in the specific results concerning the execution of the actual testing process. Figure 4.4 shows the typical development process of a test specification, which is explained in the following.

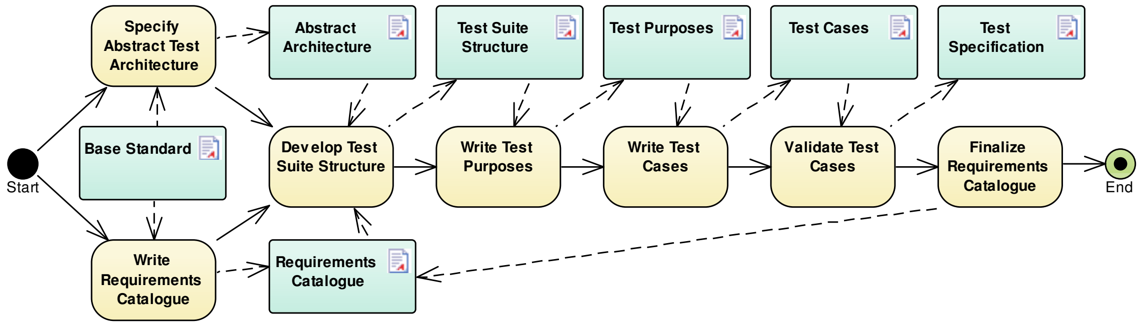


Figure 4.4: Development of a Test Specification (adapted from [22, Section 8])

On the basis of the standards and specifications that the system should be verified against (the so-called base standards), first a design of the abstract test architecture is developed and the requirements to be verified are captured in a requirements catalogue.

The abstract test architecture describes the environment, in which the tests that are specified in the test suite are performed. Thus, defining the test architecture at an early stage facilitates the specification of the actual test cases. Therefore the architecture is directly derived from base standards and specifications. For IOT the architecture characterises the logical entities involved in the test and their set-up: the QE(s), the EUT, the interfaces and communication links between both as well as any further (analysis or monitoring) equipment involved. Similarly for CT, the interfaces between SUT/IUT and test system are defined. Therefore, assumptions are defined for the lower protocol layers, which are not necessary explicitly tested in the actual CT due to complexity reasons. Additionally, upper layer interfaces and testing methods are described, which enable sending stimuli or receiving responses from the protocol layers above the IUT.

The requirements catalogue identifies mandatory, optional and conditional (depending on other functions) functionality and behaviour defined by the base standards under consideration for testing. For CT, these requirements are captured in an Implementation Conformance Statement (ICS), for Interoperability Testing it is called Interoperable Features Statement (IFS). Additionally, the ICS/IFS can serve as proforma for the SUT to state, which functionality is supported and therefore to be tested. In case of CT, it is called Protocol Implementation Conformance Statement (PICS). As depicted in Figure 4.4, at the beginning of the specification development a draft requirement catalogue is developed, which is corrected during definition process of the test suite.

Based on requirements catalogue and abstract test architecture, the Test Suite Structure is derived. Therefore, groups of tests with a logical coherence are identified. In many cases either the architecture or the functional requirements provide already a rough structure.

Afterwards, the objectives of each test case are defined by a Test Purpose to clarify the intent in enough detail that a test writer is able to specify the test case. Therefore, a Test Purpose refers to the requirements identified in the requirements catalogue and defines what is tested rather than how the testing is performed. A Test Purpose consists of header including ID, external references, and a behaviour description that is briefly explaining initial conditions, expected behaviour and final conditions.

The Test Case consists of preconditions, test steps and test verdicts. Preconditions cover for example necessary configuration instructions for setting up the CT/IUT to a specific state before running the actual test. The test steps describe at an appropriate level of detail the sequence of actions and observations, which are conducted in order to execute the test. Depending on the observation, test verdicts are assigned, which typically differentiate between Pass, Fail and Inconclusive. The latter is used for aborted tests or for test outcomes, which are not fully specified in the base standards.

Following, the test cases are validated in order to ensure that each test case fully addresses the Test Purpose. Furthermore, it has to be guaranteed that sufficient but no unnecessary preconditions have been defined and the abstract architecture can be realised such that it supports the execution of the specified test cases. Then, it is checked that the test steps provide an unambiguous and complete sequence of actions and the correct test verdicts are assigned.

At the end of the test specification development, the requirements catalogue is adjusted again. This allows the identification of inconsistencies or gaps in the draft catalogue.

4.3.2 Testing Process

The different steps of the testing processes are similar for CT and IOT, too. However, how the testing is conducted differs in the way that CT is predominantly carried out in (semi-) automatic manner while IOT mainly relies on manual execution by human testers. Thus, the procedure as depicted in Figure 4.5 is applicable for both and is therefore further explained in the following.

Typically, the testing process involves three stages:

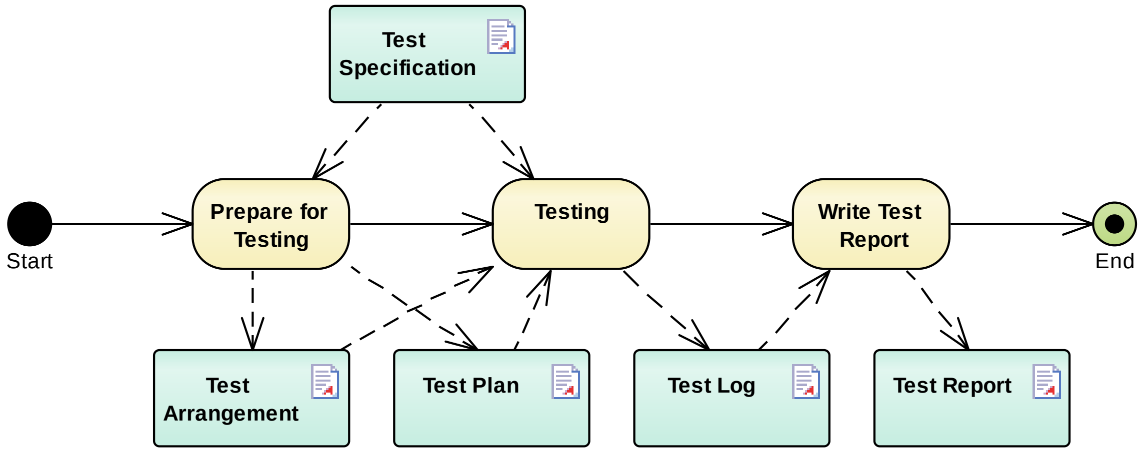


Figure 4.5: Execution of Testing (adapted from [22, Section 9])

- Test Preparation
- Test Execution
- Result Capturing

The preparation comprises deriving a test arrangement, which maps the abstract test architecture of the test specification to a specific test configuration. This results in the Test Arrangement including information on the tested system like name, serial, software and hardware version, configuration as well as on the connection between QE and EUT or SUT and conformance test system respectively. Additionally, a Test Plan is developed specifying the relevant Test Cases and their ordering for the test execution. Moreover, necessary (human) resources for the specific test cases are captured in the Test Plan.

During the testing phase, the actual tests cases are applied in the specified sequence. The test outcomes are compared with the expected outcomes and based on the comparison results, verdicts are formulated about the correctness of the implementation and are recorded in a test log.

Finally, a test report summarising the results is composed. This includes organisational information (e.g. tester name, place of testing, date and time), information on the equipment involved in the test (e.g. hard- and software ID, version, used configuration) and the test information (e.g. testing authority, summary of conducted tests and test results).

4.3.3 Re-Using Artefacts

Since maturing the specification as well as Interoperability and Conformance Testing rely on the same base standards, it turns out that many test artefacts could be shared and re-used among them. This involves formal descriptions and procedures as well as developed tools:

- Test suites containing the test cases describing objectives, required inputs and desired outputs of tests can be re-used.
- Tools developed to conduct the tests, evaluate test runs and generate reports might be universally exploited.
- Procedures for the actual execution of testing, during and after product implementation might be reapplied as well.

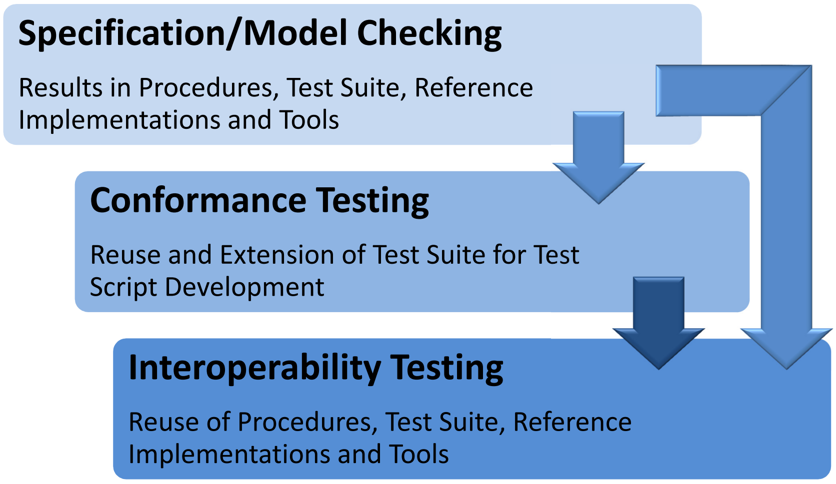


Figure 4.6: Re-Using Artefacts among Different Testing Activities

Figure 4.6 gives an overview of artefacts re-used among the different test activities. The following subsections discuss in detail, which artefacts are produced in the three phases and how they can be re-used.

Specification and Model Checking

The first priority of specification testing and related data model checking is to prove that both are consistent. For this purpose, usually test beds are developed based on the underlying standards. Therein Reference Implementation (RI) are used to identify missing documentation and contradicting or ambiguous requirements. Based on requirements of the specification and modelled use cases, test purposes and test cases can be derived, which allow to understand and analyse the functionality and behaviour of system implementations. This is complemented by test procedures and analysis tools in order to trace errors and document findings.

Conformance Testing

In order to perform CT, the derived test suite from the specification testing phase can be re-used to develop detailed test cases and test scripts, which are executed by the conformance test system to stimulate the SUT. Additionally, available Reference Implementations can be tested against the CT system to provide further trust in both, the conformance test system and the RI.

Interoperability and Acceptance Testing

IOT is usually performed for final acceptance testing to check if the developed system is able to interoperate with existing systems. For this reason, test suites developed in the previous stages provide a basis for IOT, which need to be complemented by system specific requirements in order to exhaustively validate the systems functionality. Besides, test bed architectures, procedures, and RI from specification checking might be re-used as QE since they have been matured during previous stages.

4.4 NGVA Testing Needs

The findings of the previous sections should be more deeply related to and applied to NGVA, now. As introduced in section 2.5, NGVA is a new emerging NATO STANAG, which defines standardised interfaces of future military land platform sub-systems. The different STANAG AEP volumes refer to a number of base standards – e.g. see Figure 2.1 for the information exchange specification especially.

Many of these base standards are widely adopted commercial standards, in particular with respect to the lower layer level protocols. Testing for established protocols as

Internet Protocol (IP) Version 4 is already realised for decades [57]. Even for newer protocols used in NGVA such as IP Version 6 and higher level protocols as TCP and UDP testing has been heavily studied [58, 59].

However, the core of the NGVA interoperability standardisation – specified by an NGVA DM compliant data exchange based on Data Distribution Service (DDS) – has not been comprehensively researched. So far, several interoperability test events have been conducted for DDS [60, 61] on evaluating the interoperability among overall nine vendors. Therefore, a basic test suite has been designed including functionality tests for the different implementations. The test scenarios cover basic connectivity using IP, publisher and subscriber discovery, Quality of Service (QoS) compatibility and the exchange of multiple topics in instances including content filtering. Thereby, basic interoperability of the DDS core has been proven, however an extensive IOT of the entire specification has not been conducted. Further, no conformance test tools or test suites have been developed and employed so far.

Since the first baseline of the NGVA DM is still under development, of course, neither IOT nor CT for NGVA DM messages has been conducted until now. Although, test approaches for similar military standards have been developed. For example, a conformance test suite and tools for the Future Airborne Capability Environment (FACE) has been developed [20, 62]. FACE data exchange is also based on DDS and FACE specific data model. By means of the FACE test suite, FACE Conformance Testing only aims to ensure the architecture and interfaces of the software meet the FACE requirements. Thus, no software product is executed and no performance or behaviour requirements are tested.

In addition, research has been conducted to the analysis of DDS QoS parameters by means of a framework for automated DDS performance testing [63]. Since it is challenging to predict the system performance with respect to latencies or throughput for different combinations of QoS configurations, a tool has been developed at Vanderbilt University to test various QoS configuration options while keeping the DDS application business logic remaining unchanged. Extending this approach to multiple vendors allows to conduct multi-vendor DDS IOT. Furthermore, the choice of proper DDS QoS profiles is important for NGVA as well. So far, a basic set of QoS parameter for different message types has been identified, however applicability needs to be proven in realistic scenarios during IOT sessions.

4.4.1 Test Foundations

As motivated in the previous section, NGVA-related testing is mainly needed in the area of Data Infrastructure and therein especially in NGVA Data Model compliance. Concerning the verification concept provided in chapter 3, these works especially promote the Communication Compatibility and Functional Compatibility of future NGVA sub-systems (cf. section 3.3).

In order to develop test specifications, the requirements of the NGVA STANAG need to be analysed first to draft a requirements catalogue. ConcerningDDS and NGVA DM compliance, the Data Infrastructure volume defines the relevant requirements (cf. Table 4.1 containing all Data Infrastructure requirements filtered with respect the terms *DDS* and *Data Model*).

Besides requirements for supported middleware features and the configuration of the DDS middleware, the Data Infrastructure volume and thereby the entire NGVA standard contains only few requirements for the DDS-based data exchange among vehicle sub-systems. In essence, requirements NGVA_INF_051, NGVA_INF_052 and NGVA_INF_053 state that the message exchange should be based on the NGVA DM. With respect to supported sub-system types and functionality, requirement NGVA_INF_002/063 demand support for arbitration and Health and Usage Monitoring System (HUMS) data while NGVA_INF_064 addresses the forwarding of automotive data into the Vetronics domain.

Thereby, NGVA does not specify any constrains for the implementation of specific topics or states functional and behavioural restrictions on the message exchange unless it is explicitly defined in the NGVA DM. Thus, all test cases for interoperability and conformance tests have to be derived from the data model, while some configuration parameters of the test reference system and Qualified Equipment can be obtained from the remaining requirements of Table 4.1.

4.4.2 Data Model Maturity

As introduced in section 2.5.4, the NGVA Data Model is composed of different modules, each representing a specific land vehicle sub-system. Those PIM modules are translated

Table 4.1: Data Infrastructure Requirements related to DDS and NGVA DM

Unique ID	**Type**	**Requirement Text**
NGVA_INF_002	CR	NGVA ready sub-systems shall comply with the NGVA Arbitration Protocol as defined in the NGVA Data Model.
NGVA_INF_019	OE	All device communication via DDS shall be compliant with the OMG DDS Security specification.
NGVA_INF_046	CR	DDS shall be configured to use Multicast UDP for discovery
NGVA_INF_047	OE	DDS Data distribution in the vehicle shall use Multicast UDP for DDS User Data, if supported by the DDS implementation
NGVA_INF_048	CR	Publishers targeting Multicast UDP for DDS User Data shall also support unicast subscription for potential Subscribers that cannot take advantage of Multicast UDP user data delivery
NGVA_INF_049	CR	Vetronics Data shall be exchanged on the NGVA Data Infrastructure using DDS middleware [48] conforming to the DDSI (RTPS) wire protocol [49]
NGVA_INF_050	CR	The DDS middleware shall use the default configuration listed in Section 9.6.1 of [49]
NGVA_INF_051	CR	Vetronics Data shall be exchanged by DDS topics typed compliant to the NGVA Data Model to assure assignment of DDS topics.
NGVA_INF_052	CR	Vetronics Data shall be exchanged by DDS topics using the "QoS pattern" attached to it in the NGVA Data Model to assure assignment of DDS topics.
NGVA_INF_053	CR	Vetronics Data shall be exchanged by DDS topics named exactly like the topic type to assure assignment of DDS topics.
NGVA_INF_054	CR	The DDS middleware shall be DDS XType compliant.
NGVA_INF_063	OE	NGVA Data Infrastructure and the NGVA sub-systems may implement the concept of HUMS using the NGVA Data Model
NGVA_INF_064	OE	Automotive HUMS data shall be published using the gateway between the Automotive Bus based network and the NGVA Data Infrastructure using the NGVA Data Model.

to Interface Definition Language (IDL) code, which can be used in interface implementations. Thus, the PIM modules are indirectly specifying the interface to be implemented by systems for NGVA compliance. Further, the IDL files form a direct input for IUT and CT, since they contain the message structures by means of DDS topic descriptions.

Since the NGVA DM is still under development, the different PIM modules have different levels of maturity. In order to express the module status, Module Maturity Level (MML) are used [64]. Table 4.2 gives a description of all nine MMLs. At MML 3 the modules have been fully conceptualized by subject matter experts by means of UML. Use Cases have been defined, which have thereafter turned into various class diagrams supported by behavioural UML diagrams. Finally, the content of the PIM modules has been agreed by the NGVA DM review group and the error-free translation into IDL has been proven.

As discussed in section 4.1, an early development of test tools facilitates the standardisation and the increases the acceptance of the standards itself. Testing should be linked with the standardisation as early as possible in order to provide valuable improvements of the specification and feedback to the standard authors. Hence, confidence in the standard is given to vendors in an early phase of the standard life-cycle as well as a foundation for later test tools and Reference Implementations is provided.

According to Table 4.2, MML 5 is a good candidate to combine early feedback into standardisation as well as to provide initial input for later CT and IOT efforts. MML 5 approval requires the test of "the complete set of classes for that PIM on a development rig which includes the simulation of operating applications that use the interface data structures" [64]. Therefore, an implementation of the entire DM module is needed, which – if properly realised in a forward-looking way – can be directly used for IOT afterwards. Thus, the next section proposes an approach to raising DM modules from MML 3 to MML 5 and describes the process of adapting the gained artefacts for CT and IOT afterwards. This dramatically improves the DM module maturity. Afterwards, the modules are accepted as fit-for-implementation in a prototype demonstrating actual vehicle sub-systems.

4.5 Early De-Risking Future Vetronics Implementiations

This section describes a framework for integrated Specification, Conformance and Interoperability Testing of future military land vehicle sub-systems on the example of NGVA. As discussed in section 4.2, preparing the different test activities needs the definition and development of test specifications containing a test architecture and the test cases to be executed by means of an realisation of the architecture. The following subsections explain how such a framework can be realised for all three testing aspects.

Table 4.2: Module Maturity Level Definitions [64]

Level	MML Description	Equivalent TRL Description
MML 9	Data interface model generated from module PIM has been embedded in multiple deployed designs and proven to operate.	Actual technology system qualified through successful mission operations.
MML 8	Data interface model generated from module PIM has been embedded within an actual electronic architecture design which has passed all test and validation and is proven in-Service.	Actual technology system completed and qualified through test and demonstration.
MML 7	Data interface model generated from module PIM has been embedded within an actual electronic architecture design, and is ready for final test and demonstration.	Technology system prototype demonstration in an operational environment.
MML 6	Data interface model generated from module PIM has been embedded and implemented in a whole system context either on a systems integration rig or on an actual system using a majority of real components.	Technology system/ sub-system model or prototype demonstration in a relevant environment.
MML 5	Data interface model generated from module PIM has undergone testing of the complete set of classes for that PIM on a development rig which includes the simulation of operating applications that use the Interface data structures.	Technology component and/or basic technology subsystem validation in relevant environment.
MML 4	Data interface model generated from module PIM has undergone initial lab tests by ensuring that all classes have been exercised by at least one write operation and at least one read operation, thereby demonstrating correct Data transport.	Technology component and/or basic technology subsystem validation in laboratory.
MML 3	Module PIM has been subject to several reviews, agreed by an approved review body or working group and has been translated and compiled without errors.	Analytical and experimental critical function and/or characteristic proof-of concept.
MML 2	Module PIM has undergone a single review against relevant Use Cases at a stakeholder workshop session, module elements are fully documented and the module is compliant with LDM Methodology.	Technology concept and/or application formulated.
MML 1	Initial Use Cases and PIM created for Module.	Basic principles observed and reported.

For early de-risking, the framework supports the analysis and evaluation of the specification at an early stage of the specification life-cycle, but also provides artefacts to be re-used in later CT and IOT phases. With regard to NGVA, therefore a approach is presented which takes into account the maturing of the specified NGVA DM modules by means of MML 5 first. Afterwards, the results find input into Conformance Testing and the development of a distributed IOT test laboratory.

The initial MML 5 test set-up is extended to a test architecture for IOT. The prototypical implementations serve as a basis for later Reference Implementations deployed in the IOT test lab, while the developed test cases form the basis for CT and IOT test suites. This approach is presented and evaluated in a detailed case study of the NGVA Brake PIM in chapter 5.

4.5.1 Module Maturity Level 5 Testing

Module Maturity Level 5 Testing is a valuable effort in maturing the DM but also in order to prepare later CT and IOT. First of all, MML 5 testing should provide improvements for the specification and feedback to the standard authors.

Although not explicitly stated in its definition (cf. Table 4.2), MML 5 aims at proving that the DDS topics resulting from a particular PIM interact logically and form a complete representation of the system communication represented in the module. Therefore, MML 5 testing checks if the PIM module serves the purpose and intention by testing all topics of a PIM in a coherent way. In doing so, it allows to detect missing topics needed to fulfil its purpose, to detect missing or redundant relationships and to detect topics not covered by any use case.

However, a MML 5 test of a PIM module is by definition always independent of other PIM modules since no links across PIM modules are tested. Thus missing or redundant inter-PIM-relationships are not discovered. This aspect has to be covered by later IOT.

Additionally, the tests may not consider all information exchange needs of real equipment. For example, if additional topics are needed in a functional manner, they cannot be determined by MML 5 testing since only the existing PIM is tested. To address this, the information exchange needs of existing legacy hardware should be analysed, e.g. by checking available Interface Control Documents or conducting vendor surveys.

In order to mitigate this uncertainty, all topics resulting from an NGVA DM PIM should be tested using components and applications as closely as possible representing real systems. This ideally involves real components (sensors, effectors, applications, etc.) but should at least incorporate simulations or prototypes.

In contrast, major difficulties of performing MML 5 testing with real components are:

- Real vehicle sub-systems barely implement the complete set of messages resulting from a PIM. Typically, only messages that are either mandatory or operationally needed are implemented.
- Testing by means of real sub-systems is often expensive due to hardware costs and hardware-specific software development restrictions.

For these reasons, a MML 5 testing approach has been designed that targets towards the use of simulation, software and hardware prototypes (proof-of-concept) that are as similar as possible to real equipment. The approach is aligned with the test framework development presented in section 4.3.

Abstract MML 5 Test Architecture

From the Vetronics equipment's point of view, the IDL file resulting from a PIM consists of two types of topics: topics which are consumed by and topics which are published by the equipment implementing the IDL file. Since those topics need to be tested in both directions, a MML 5 test architecture needs to consider both communications ends. Therefore, a first test component representing the actual equipment and a second component providing stimulus to and receiving output from the actual equipment is needed.

Moreover, NGVA PIM modules can be categorized according to whether they do or they do not depend on inputs and outputs from other PIMs. This shall be distinguished by the terms *processing* and *terminal* components, respectively.

Terminal components include equipment such as sensor devices, weapons, etc., which are either a source or a sink of sub-system data. *Processing components* represent intermediate nodes of the processing chain. This can be data fusion and threat assignment applications, for example.

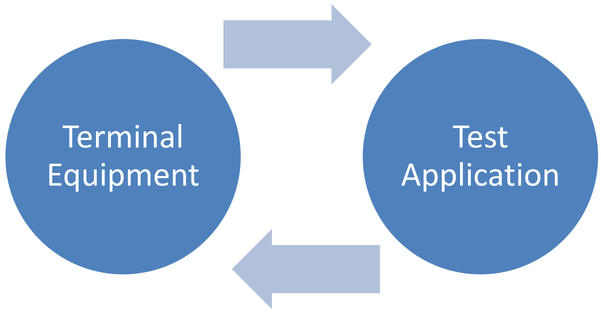


Figure 4.7: Terminal Component Test Set-Up

For *terminal component* PIMs, testing might be conducted using (virtual) simulators or hardware prototypes representing the equipment side (cf. Figure 4.7 left) and an additional test application that creates stimuli and displays the output (cf. Figure 4.7 right). E.g. a laser range finder or a weapon could be simulated, while a video sensor could easier be tested with prototypic, low cost lab hardware based on the Arduino or Raspberry Pi platform. For creating the stimuli or displaying (video) output, a prototypic software implementation or an adapted Command and Control Information System (C2IS) is sufficient.

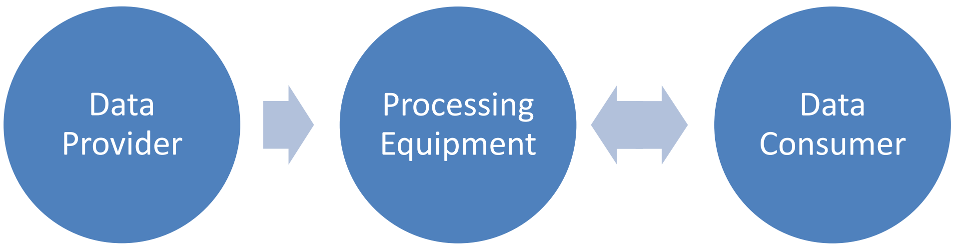


Figure 4.8: Processing Component Test Set-Up

In cases of *processing component* PIMs, the testing effort depends on the capabilities of the *processing components*, but is expected to be bigger than for *terminal components*. First some (terminal) equipment providing the data to be processed needs to be implemented, e.g. by simulations or hardware prototypes again. Of course, in this case only information relevant for processing needs to be simulated. Then, the actual processing capabilities need to be mocked or fully realised. Finally, the processed information needs to be presented by a consumer, which could be a prototypic software or again an adapted C2IS (cf. Figure 4.8). Additionally, the data consumer might configure the processor accordingly to its needs.

In summary, the set-up should at least consists of a test component representing the actual (terminal or processing) equipment and further components providing stimulus to and receiving output from the actual equipment. In cases of *terminal components* a single additional component can be sufficient, while for *processing components* several components might be necessary to provide data to the component and to receive the output. This is important when it comes to the actual deployment in the development rig, since the parts would be deployed to different network nodes.

For MML 5 testing, it is assumed that the lower level NGVA protocols (Physical, Data Link, and Network Layer) as defined in Figure 2.1 function according to specification. The focus explicitly lies on testing the DDS data exchange of topics specified in the module.

The architecture developed for MML 5 testing and the resulting implementations can be re-used and adapted with minimal effort as a set-up for IOT, which is discussed in section 4.5.3.

MML 5 Test Suite Development

MML 5 testing is not accomplished by implementing the IDL file only, since it should serve a "subsystem validation in relevant environment" (cf. Table 4.2). In order to check that all topics work reasonably together, an approach to derive the test cases based on the NGVA DM is presented in this section, which can be used with a realisation of the proposed architecture.

Therefore, test cases are derived based on the use cases and (if available) further behavioural UML diagrams, which are outlined in the PIM under test. The PIM use cases should serve as the foundation since they already outline a sequence of steps that describe the desired PIM module interactions.

E.g. for *terminal equipment*, the use cases describe what the actor wants to perform with and which output he expects to receive from the terminal equipment. Thus, test cases derived from use cases are particularly helpful for finding defects in the real use of the equipment.

Thus for MML 5 testing, use cases described in the PIM could be adapted as test cases as followed:

- The specified preconditions of the use case must be adapted for and must be met prior to the start of the actual test case.
- Each use case has at least a basic flow and sometimes additional alternative flows (covering, for example, special cases or exceptional conditions). All branches (basic/alternative flows) should be run through, which may result in multiple test cases in the test suite for a single use case.
- For each test case, post conditions adapted from the use cases (which are observable results and a description of the final state of the system) needs to be defined stating conditions required to pass the test case.

MML 5 Test Process

The test process starts with the prototypical implementation of the IDL file in two or more components as outlined in the previous sections. As stated in the MML 5 definition, the implementation has to cover all topics of the IDL file as DDS data writers and DDS data readers. Implementing all topics in a cooperative manner allows evaluation, if all topics fit logically together and can be properly linked. Thus, missing or redundant relationships not modelled in the PIM module are discovered.

Afterwards, the actual the testing comprising the run of the developed test cases is conducted and the results are captured. During the test case runs, the exchanged topics should be logged.

This procedure allows to evaluate the topic coverage and to detect topics not utilised by any use case. This might happen due to various reasons: First, there are topics resulting from the PIM, e.g. specification topics, for the purpose of defining the capabilities of the equipment. Those are mandatory to implement, but their usage is generally not explicitly covered by use cases. Second, there could be additional functional topics, for which an explicit use case is missing, but which provide beneficial functionality for the equipment. If reasonable, additional use cases to cover both issues could be added to the next PIM module release. Third, if there are further missed topics, additional test cases need to be developed to cover those topics as well and ensure that all topics are tested. It is recommended to request for a justification for the topics, e.g. by means of use cases, in the outcome capturing.

At the end of the testing process, a review must be conducted to check if all topics have been tested and the model fits the use cases. The outcome should be captured in a MML 5 test report. For each DM module, the following should be documented:

- Organisation and personnel involved
- Date of testing
- Version of the PIM that has been tested (e.g. by stating the subversion revision)
- Version of the PIM translator and parameters/properties used to generate the IDL file
- Used DDS vendor and version
- QoS parameters used for the topics
- Detected missing topics or missing links between topics
- Additionally developed use cases to complete topic coverage
- Recommendations for PIM improvements to pass MML 5 testing

Additionally, for each PIM use case, the report should give information about

- Mapping of Test Cases to use cases
- Mapping of topics to Test Cases
- Conducted test case runs along with relevant test evidence, e.g. output logs, screenshots
- If the test cases have been passed/failed with existing topics

4.5.2 NGVA Test Reference System

This section describes the realisation of a conformance test system for the data exchange of NGVA-based vehicle sub-systems. The test system allows verifying NGVA DM compliance of sub-systems with respect to interface implementation and system behaviour by means of stimulation with valid and inopportune messages. For this purpose, the test system architecture and test categories are derived from the NGVA specification.

Conformance Test System Architecture

In order to provide the possibility of checking system implementations for NGVA compliance, a NGVA DM Test Reference System (NTRS) was developed based on the successful *MIP Test Reference System* [29]. As for MML 5 testing, it is assumed that the lower level NGVA protocols work properly and that the testing needs only to focus on the topic implementation and the used QoS parameters.

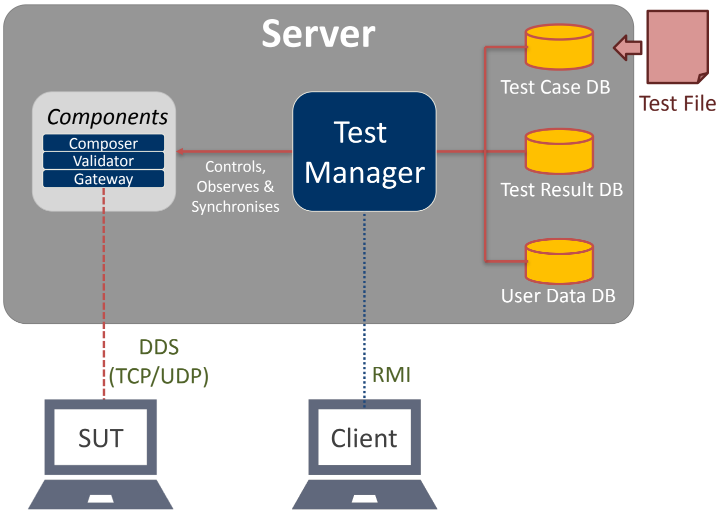


Figure 4.9: NTRS Architecture

As depicted in Figure 4.9, the NTRS uses a client-server design consisting of a Test Server, Test Client and SUT. The Test Server consists of a Test Manager, Components and several databases. The Test Manager handles the execution and evaluation of Test Cases. For this purpose, it sets up, controls, observes, and synchronises Test Components to accomplish specific tasks. Additionally, the Test Suite, the test results as well as user data is stored in server databases. A Test Operator is able to run Test Cases and to analyse the data processing via a Test Client. The next section gives a brief overview about the message exchange between the Test Server components.

Test Server

Three Components have been conceptualised and implemented for the NTRS Test Server: composer, validator and gateway. Figure 4.10 depicts a component-wise representation of the NTRS information flow.

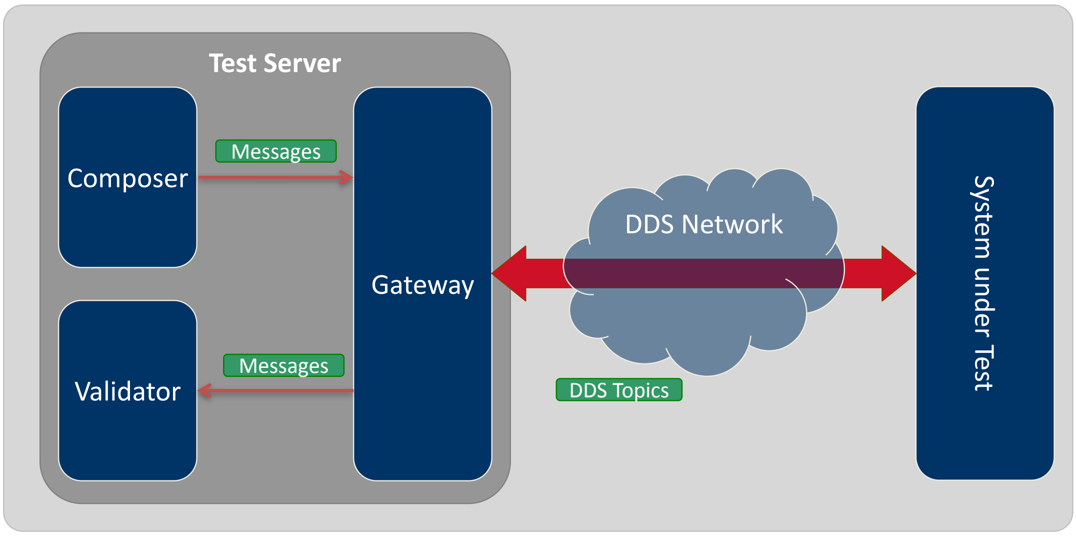


Figure 4.10: NTRS Information Flow

A Message represents the information, which is exchanged between different components of the NTRS. In this context, an NTRS Message basically wraps a Topic. It does not introduce new functionalities but rather ensures the Topic itself is handed over to other components by means of NTRS Messages.

Serving as a connection point to the DDS network, the gateway component handles the wrapping and unwrapping of every DDS Topic passing through. The composer generates randomised data for Topic attributes if appropriate, whereas the validator checks data after reception. Additional components can be integrated with little effort in case there is a need.

Test Client

The NTRS Test Client represents the user interface for executing and navigating through Test Cases as well as for the logging of test runs. As depicted in Figure 4.11, the client is composed of three main areas; Test Suite Browser, Properties Panel and Test Run Panel.

The Test Suite Browser (on the left) depicts the entire Test Suite including all Test Groups and Test Cases in a hierarchical tree structure. Additionally, for every Test Case all Test Runs can be viewed. The Properties Panel (in the upper-right corner) displays meta-information for every tree node selected in the Test Suite Browser. Herein SUT

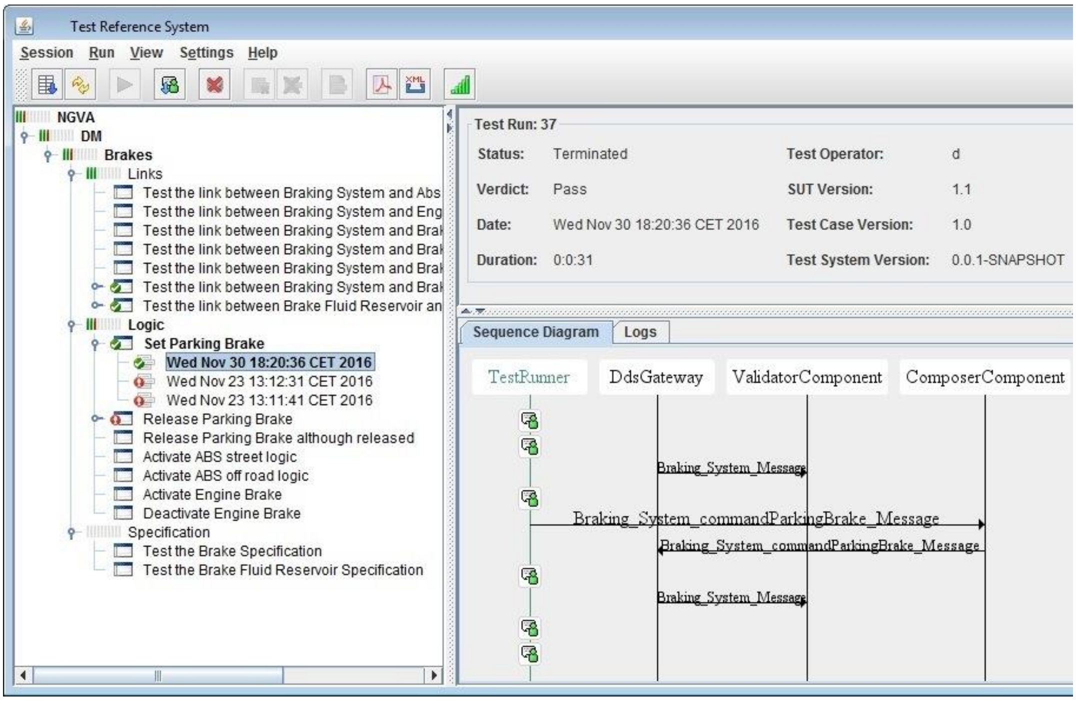


Figure 4.11: NTRS Test Client

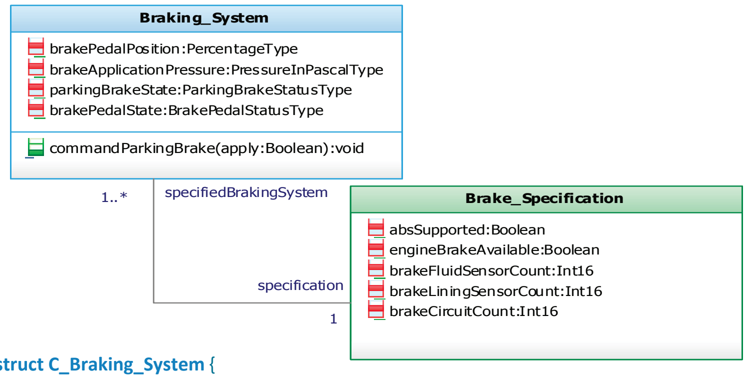
information is given as well. Finally, the Test Run Panel displays the NTRS message flow between components in form of a sequence diagram.

NGVA Conformance Test Suite

Since no specific compliance requirements are defined for NGVA DM testing so far, three test categories have been derived based on the DM structure and content; logic-based, link-based and specification-based testing. The different test categories also result in test groups for the conformance test suite (cf. Figure 4.11), which contain the actual test cases.

Link-Based Testing

The link-based testing category concerns verifying whether the topics correctly refer to each other. To better understand the scope, the notion of Topic Instances needs to be further examined. An Instance is unique for a specific Topic and is created by selecting one or more fields as key. For NGVA state and specification Topics, the *sourceId* is defined as a single key whereas commands have the *recipientId* as a second key attribute.




```
struct C_Braking_System {
        P_LDM_Common::T_IdentifierType                 A_sourceID; //@key
        P_LDM_Common::T_DateTimeType                   A_timeOfDataGeneration;
        P_LDM_Common::T_PercentageType                 A_brakePedalPosition;
        P_LDM_Common::T_PressureInPascalType           A_brakeApplicationPressure;
        T_ParkingBrakeStatusType                       A_parkingBrakeState;
        T_BrakePedalStatusType                         A_brakePedalState;
        P_LDM_Common::T_IdentifierType                 A_itsBrake_Specification_sourceID;
        P_LDM_Common::T_IdentifierType                 A_abs_sourceID;
        P_LDM_Common::T_IdentifierType                 A_optionalEngineBrake_sourceID;
        sequence <P_LDM_Common::T_IdentifierType>      A_brakeCircuit_sourceID;
        sequence <P_LDM_Common::T_IdentifierType>      A_brakeLining_sourceID;
        sequence <P_LDM_Common::T_IdentifierType>      A_brakeFluidReservoir_sourceID;
};

struct C_Brake_Specification  {
        P_LDM_Common::T_IdentifierType                 A_sourceID; //@key
        P_LDM_Common::T_DateTimeType                   A_timeOfDataGeneration;
        P_LDM_Common::T_Boolean                        A_absSupported;
        P_LDM_Common::T_Boolean                        A_engineBrakeAvailable;
        P_LDM_Common::T_Int16                          A_brakeCircuitCount;
        P_LDM_Common::T_Int16                          A_brakeFluidSensorCount;
        P_LDM_Common::T_Int16                          A_brakeLiningSensorCount;
        sequence <P_LDM_Common::T_IdentifierType>
        A_specifiedBrakingSystem_sourceID;
};
```

Figure 4.12: Link-Based Testing Example

Figure 4.12 depicts this concept by the example of two PIM classes and the resulting IDL code. The association between the *Braking System* and the *Brake Specification* class in the PIM is reflect by the Topics containing attributes for mutual *sourceIds* (depicted in red).

Since an Instance is unique per Topic type and provided that the latter is captured by naming convention, the correct referencing via *sourceId* is the objective of link-based testing. For instance, a *Braking_System* sample refers to a *Brake_Specification* Topic by the value stored in the *A_its_Brake_Specification_SourceID* variable. To link-test this reference, a *Brake_Specification* Topic with that identifier is expected. In case it is received, its list *A_specifiedBrakingSystem_SourceID* is checked for the *sourceId* of the *Braking_System* to be present.

Another type of link testing concerns the state-command (or class-operation) connection as depicted in Figure 4.13. Even though it is not captured in the PIM explicitly, this type of connection is present at the Topic level between a command and its corresponding state in the PSM and in the IDL file. The member *A_recipientId* of *C_Braking_System_commandParkingBrake* refers to *A_sourceId* of *C_Braking_System*. Therefore, the class-operation connectivity, which is being tested, is represented by this arrangement.

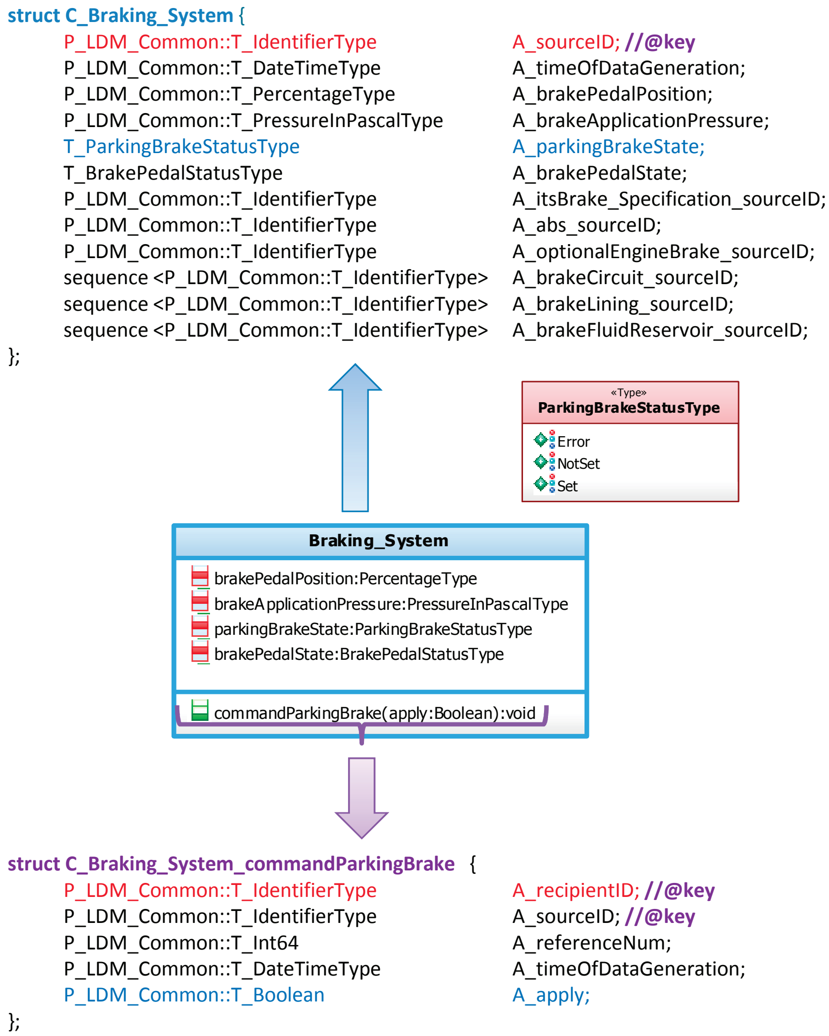


Figure 4.13: Command-State Link Testing Example

Logic-Based Testing

This type of testing originates from behavioural as well as structural diagrams specified in the PIM. It verifies whether the entities interact logically correct. Generally, it is based on the effect that a specific command sent by the NTRS, triggers an answer from the SUT and vice versa. Derived from the *Braking_System* class shown in Figure 4.13, a simple logical Test Case comprises the stimulation of the braking system through a command, which is afterwards triggering a new state topics caused by a state change.

For instance, a *C_Braking_System_commandParkingBrake* with the *A_apply value* (depicted in blue) set to *TRUE* sent by the NTRS, triggers the sending of the *C_Braking_System* state by the SUT, whose member *A_parkingBrakeState* must be *Set*. Any value other then *Set* would cause the test to fail.

Specification-Based Testing

This type of testing checks whether specified Topics are implemented as defined in the specification Topic. In Figure 4.12, for example, the *C_Brake_Specification* represents the specification Topic and *C_Braking_System* the specified Topic. In this case, the specified Topic is a state, but it could be a Specification Topic as well.

Considering engine braking is supported, which is reflected in the *C_Brake_Specification* by having its member *A_engineBrakeAvailable* set to *TRUE*, a simple specification-based test checks whether the *A_optionalEngineBrake_sourceID* in the *C_Braking_System* is not empty.

Similarly, if *A_brakeCircuitCount* in the *C_Brake_Specification* is set to a specific number, it should equal the sequence size of *A_brakeCircuit_sourceId* in *C_Braking_System*. It is worth noting that this type of testing generally tests the specified topics with respect to their structure and content. For example, in case ten *C_Brake_Circuit* Topics are expected but only nine are received, it may indicate a network failure as well.

Conformance Test Process

In order to initiate a conformance test of an actual Vetronics system, the system developer or Test Operator has to register for and configure the conformance test server.

Configuration can be done remotely via a Secure Shell connection, so it is independent of time and location.

The configuration is conducted by means of the NTRS Test Client as depicted in Figure 4.11. When the Test Client is started, a SUT configuration dialogue (cf. Figure 4.14) pops up. It gathers data, which is relevant in order to connect the SUT. The collected data is transferred to the Test Server and is used to properly configure and execute test cases.

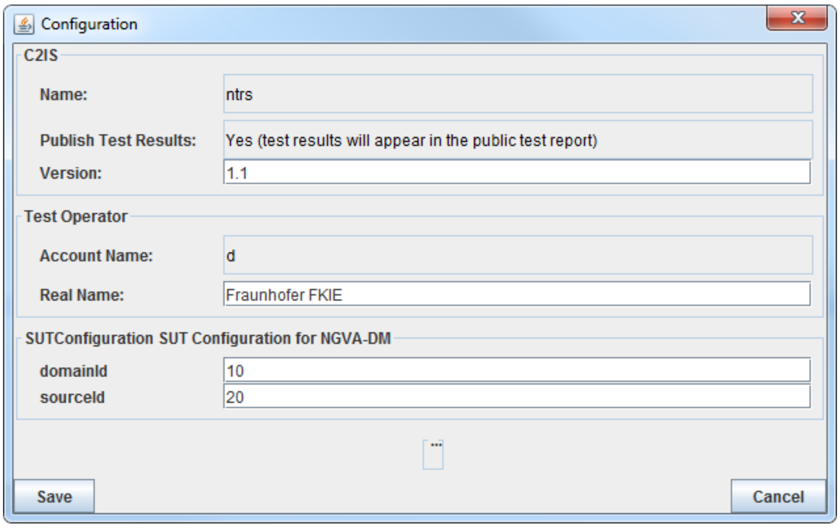


Figure 4.14: SUT Configuration Dialogue

Besides Test Operator information, the user should also provide information about the System Under Test (SUT) representing the current NGVA system or sub-system being tested. To test a specific SUT as well as to better outline the test results, the Test Server has to collect certain data about the SUT. This data includes parameters, which had been defined for the SUT. This includes the domainId, which represents the DDS domain in which the DDS entities of the NTRS communicate with those of the SUT, or the *sourceId*, which is identical to the source identifier of the topics originating from the SUT.

After configuring the test server, the Test Operator selects the test cases to be applied to the System under Test. After starting, the test sequence is automatically executed and the results are displayed to the operator and stored in the Test Result database.

4.5.3 Interoperability and Acceptance Test Laboratory

For conducting Interoperability and Acceptance Testing, a test laboratory has been created. The test lab was originally created on order to analyse the UK GVA and to accompany the NGVA standardisation process. Additionally, it was used to support research in the NGVA domain with respect to crew terminal integration [65] and the adaption of legacy hardware [66].

The test laboratory was designed based on existing GVA concepts and specifications. However, it was heavily extended and was continuously adapted to changing requirements during the NGVA standardisation process. The following sections give an overview of the architecture as well as the intended test applications and procedures.

Test Laboratory Architecture

An abstract overview of the Test Lab architecture and its components is given in Figure 4.15. The architecture considers that certain Vetronics components have to be simulated, since there is not necessarily a real vehicle connected to the test lab. In addition, management of the test laboratory and remote access is provided.

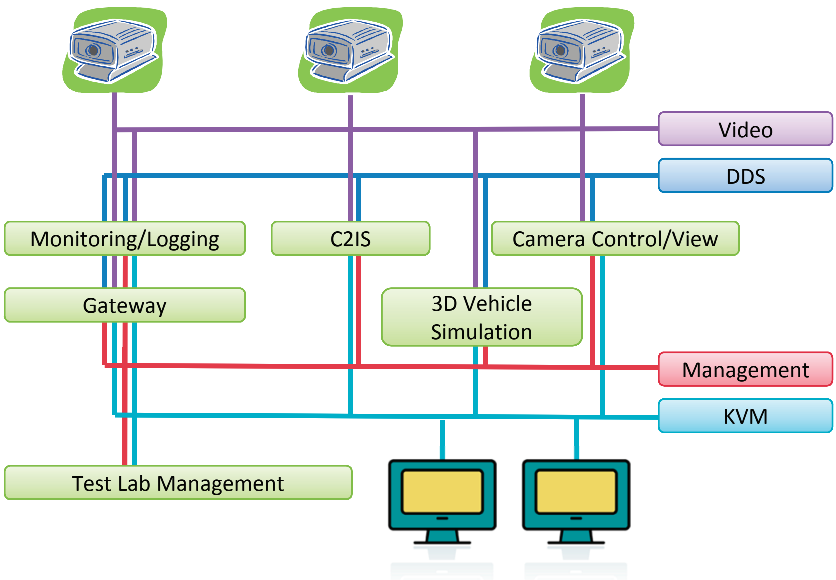


Figure 4.15: Simplified NGVA Interoperability Test Configuration

The basis of the test lab form four Ethernet networks. The first two subnets represent the actual test networks, while the other ones mainly serve test management purposes:

- The DDS subnet is used to exchange Vetronics data. It is the core network and allows to connect the (sub-) System Under Test. The components communicate only using NGVA DM messages based on the DDS middleware in this network.

- A video subnet is used to separate video sensors or other sub-systems requiring high data rates that can severely impact DDS Ethernet.

- The management subnet serves for the independent control of all components. This subnet allows to partly or completely set-up, update, and monitor the laboratory during operation. Further, it is used to start, stop, and reconfigure individual components of the test lab without affecting the rest of the lab. To accomplish this, a management component is used, which is as far as possible decoupled from the sub-networks of the components to be monitored by having no direct access to the DDS and video sub-networks.

- A Keyboard Video Mouse (KVM) subnet is used so that no separate workstation is needed for each additional lab component.

In order host the relevant vehicle services as well as simulated or real vehicle implementations, a number of processing nodes are connected to the test lab networks. The software deployment on these nodes includes for example:

- 3D simulators to simulate the movement and interactions of an NGVA vehicle in a synthetic environment

- Video display and control applications receiving video streams from real cameras connected to the video network

- C2ISs, which receive vehicle status information and display it on a map

- Software gateways allowing the integration of automotive, safety critical or vehicle external networks

- Applications to monitor and to log the data exchange among the Vetronics subsystem for interoperability or acceptance test purposes

In comparison to Conformance Testing, the test laboratory architecture is not restricted to testing a correct NGVA DM implementation only. The test laboratory allows to consider all three NCLs from testing the usage of correct connectors on Connectivity Level up to performance measurements on the Functional Compatibility Level.

Deriving Interoperability and Acceptance Test Cases

In contrast to the test suites derived for specification and conformance testing, test suites for interoperability or acceptance testing cannot be deduced from the standards or specifications only. Usually, the customer specifies additional requirements in the SRD addressing further functionality or demand specific QoS e.g. with respect to timing, which have to be part of acceptance testing, too.

Nevertheless, the portion of test cases originating from the NGVA specification can be similarly derived as introduced for earlier phases. Therefore, existing test groups and cases from these stages can be re-used. In addition, further test cases handling inter-module information exchange aspects need to be defined. This is not addressed by previous test activities, since these activities specify only test cases for one particular DM module.

In cases of more complex sub-systems like surveillance units, topics of several DM modules must be combined in order to receive the desired information – for example the orientation of the vehicle, the vertical and horizontal direction of the turret mounts, and the distance measurement of the laser range finder in combination results in target data required by a C2IS.

Interoperability and Acceptance Test Process using the NGVA Lab

The NGVA Test Lab is supposed to have the capability of simulating all Vetronics components of actual military land vehicles. However, the main function is to analyse if software interfaces are implemented according to the NGVA Data Infrastructure specification and the NGVA Data Model and are interoperable with existing Qualified Equipment. Further relevant aspects, such as the test of the power supply according to NGVA Power Infrastructure specification or the examination of electronic hardware interfaces also need to be addressed in the testing processes, but play a minor role.

The scenario of acceptance testing of separate NGVA components, for example a camera system, is the most probable one in the early phases of the standards life cycle. For NGVA compliance testing the following procedure has been derived in discussion with military personnel.

Test Preparation

In the test preparation phase, it must be specified, which dependencies the SUT has regarding other vehicle sub-systems. For example, many sub-systems rely on available GPS data in order to work properly, since they either need the vehicle position or need to extract date and time. Further, depending on the SUT design, NGVA services like a vehicle configuration service or Health and Usage Monitoring System (HUMS) might be required.

Next, it must be defined, which services and information are provided by the SUT in order to configure the test lab components to receive and display the results. This information is also relevant for the configuration of logging or evaluation components.

Before starting the tests, power supply and a connection between the System Under Test and the QE has to be established. The latter can be done by connecting the component to the NGVA test lab, start-up the component and executing the test equipment on the test lab nodes.

Test Execution

Since many sub-systems are uniquely designed for specific vehicles, testing these systems for the first time is often a manual effort. However, logging the test series – e.g. stimuli provided by the test operator – can provide means for (semi-) automatic testing in case regression tests are needed.

Based on the SUT capabilities, the test equipment needs to be configured. This can be manually done by the test operator, but should be supported as far as possible by *auto-detect* features – for example by recognising the type and number of NGVA DM modules and instances to be tested. E.g., for a pan-tilt 180° camera, this would include several instances of the *Video* and *Mount* modules. The test equipment needs to be adapted the version and to the extent of the NGVA DM modules that are implemented – for example, a camera does not have to implement the complete video module of the NGVA DM.

Then, the test operator needs to select supported messages for monitoring and is supposed to control the equipment manually according to the acceptance test criteria. In addition to manual testing, the operator might re-use previously generated test cases, e.g. in case of regression tests.

Finally, the test operator can determine if each test has been carried out successfully or if it has failed. The results are stored with date and time and might also be supplemented with free text notices of the operator.

Test Result Capturing

After executing the test cases, again a test report needs to be issued. The report can be structured as discussed in specification testing section 4.5.1.

4.6 Conclusion

Since not all aspects can be verified on a theoretical basis, verification procedures and tools to support the testing activities in the different stages of a specification life-cyle have been evaluated and designed. The approaches tackle especially the problem of early de-risking in the development of NGVA-based sub-systems. A test framework was derived allowing to conduct Specification, Interoperability and Conformance Testing of NGVA components with the added-value that a re-use of test artefacts among test phases is possible. The extent of re-using test tools and test suites is further evaluated in chapter 5.

The presented test framework has been published at an IEEE-listed conference [67]. The MML 5 specification test approach has been discussed with the NGVA DM community and was accepted as a common approach for conducting NGVA MML 5 accessment by all countries conducting NGVA DM MML 5 testing [68]. Further, the conformance test system and test suite was published at another IEEE-listed conference [69]. The design of the conformance test system and test suite was conducted in collaboration with Ditmir Hazizi under the research grant *NGVA Verification* supported by the German Ministry of Defence.

5 A Sub-System Verification Case Study

Chapter 4 outlined an integrated framework for Specification, Conformance and Interoperability/Acceptance Testing of NGVA-based (sub-)systems. The framework aimed at re-using test artefacts in later testing phases, which have already been realised during previous phases.

This chapter evaluates the framework by testing the NGVA Brakes Module PIM and by showing how artefacts can be forwarded to later phases as well as even be feed back to earlier phases in the case of testing further PIM modules. Additionally, future planned framework extensions and improvements are explained.

5.1 Brakes Module Maturity Testing

This section describes a case study analysing the specification testing approach described in section 4.5.1 by conducting MML 5 Testing of the Brakes Module as specified in the draft NGVA DM. MML 5 Testing primarily aims at ensuring that the information modelled in the class model of the Brakes Module is sufficient to execute all use cases and sequence diagram behaviour described in the Brakes Module. Therefore, test cases are specified, which are derived from the use cases and sequence diagrams of the module. Further, the test environment used to execute the tests is defined and the test results are captured.

As discussed in previous chapter 4, MML 5 Testing can be seen as a pre-step to Conformance and Interoperability Testing. Thus, the test specifications defined in this section aim primarily to conduct MML 5 testing, but also provide a basis for later testing phases of potential NGVA-based brakes systems. For this reason, terminology originating from the interoperability testing domain is already used as the basis for the test descriptions.

To improve and validate the maturity of the NGVA DM at MML 5, it needs to be proven for a PIM that the "Data interface model generated from module PIM has undergone testing of the complete set of classes for that PIM on a development rig which includes the simulation of operating applications that use the interface data structures" [64]. Since this requires an implementation of the entire DM module, MML 5 is a good candidate to combine early feedback into standardisation with providing an initial basis for realising CT and IOT.

5.1.1 Test Case Description Terminology

The MML 5 test case descriptions are provided in tables following a common structure. The test case header specifies a unique test identifier, the test objective, the test configuration to be used, and references to DM artefacts of the Brakes Module justifying the test cases. The pre-conditions define constraints that need to apply before starting the test.

The following different types of test operator actions are considered during the test execution:

- A *stimulus* corresponds to an event that enforces test equipment to proceed with a specific action or response, e.g. by sending a command topic sample.
- A *verify* consists of verifying that test equipment behaves according to the expected behaviour (for instance test equipment behaviour shows that the stimulus of a command topic is processed correctly).
- A *configure* corresponds to an action to modify the test equipment configuration, e.g. by requesting the test equipment to change into a specific mode.
- A *check* ensures the receipt of topic samples with valid content. A *check* event type corresponds to interoperability testing with conformance checks. For instance, in the context of MML 5 testing, it is relevant to check if information of specification, state and command topics by the (sub-)system are coherent.

During the execution of MML 5 testing sessions, every step of a test description should be captured with a monitoring tool.

5.1.2 Input Analysis

First, an input analysis of the Brakes Module was conducted analysing the use cases and sequence diagrams to generate the test specification consisting of test purposes and test cases as well as the test architecture and equipment configuration.

Figure 5.1 shows the class diagram of the Brakes Module consisting of eight classes with three operations. The six blue classes are translated into State Topics, the green ones into Specification Topics and the operations into Command Topics.

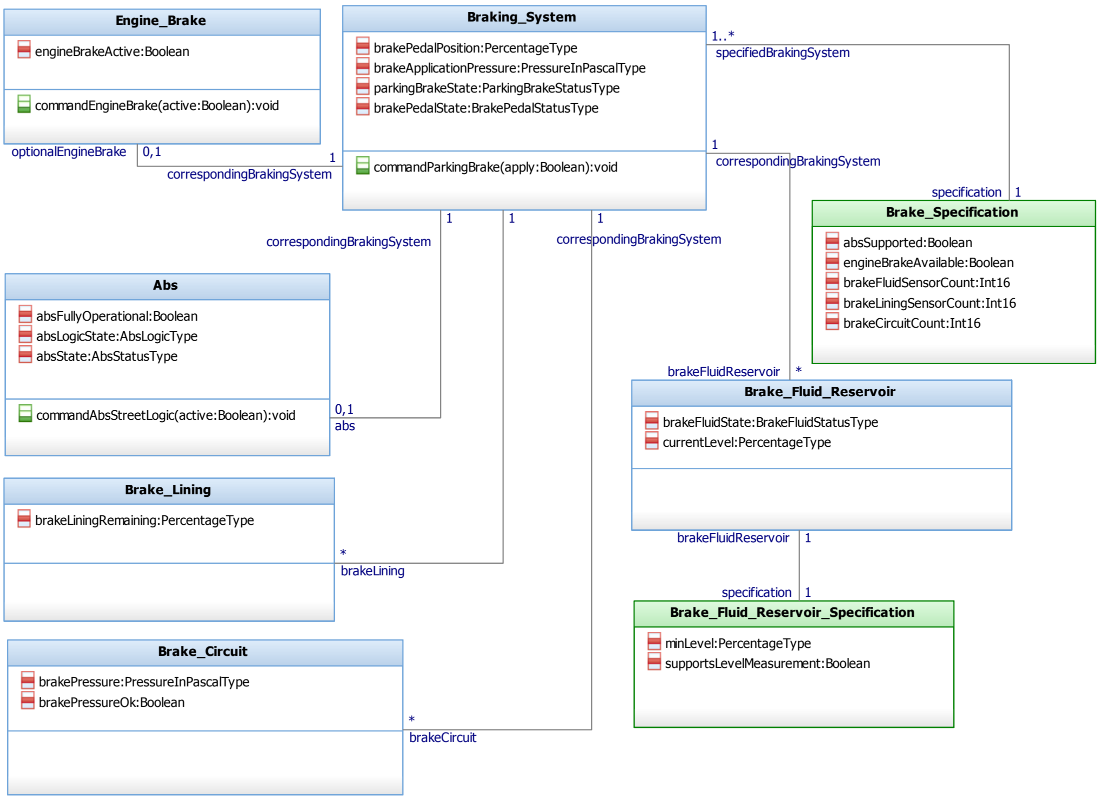


Figure 5.1: NGVA Brakes PIM Class Diagram

Furthermore, the Brakes Module specifies seven detailed use cases (cf. Figure 5.2), which are defined in detail by use case descriptions (Brks_UCxx). As an example, the use case *Monitor Brake System* is shown in Table 5.1.

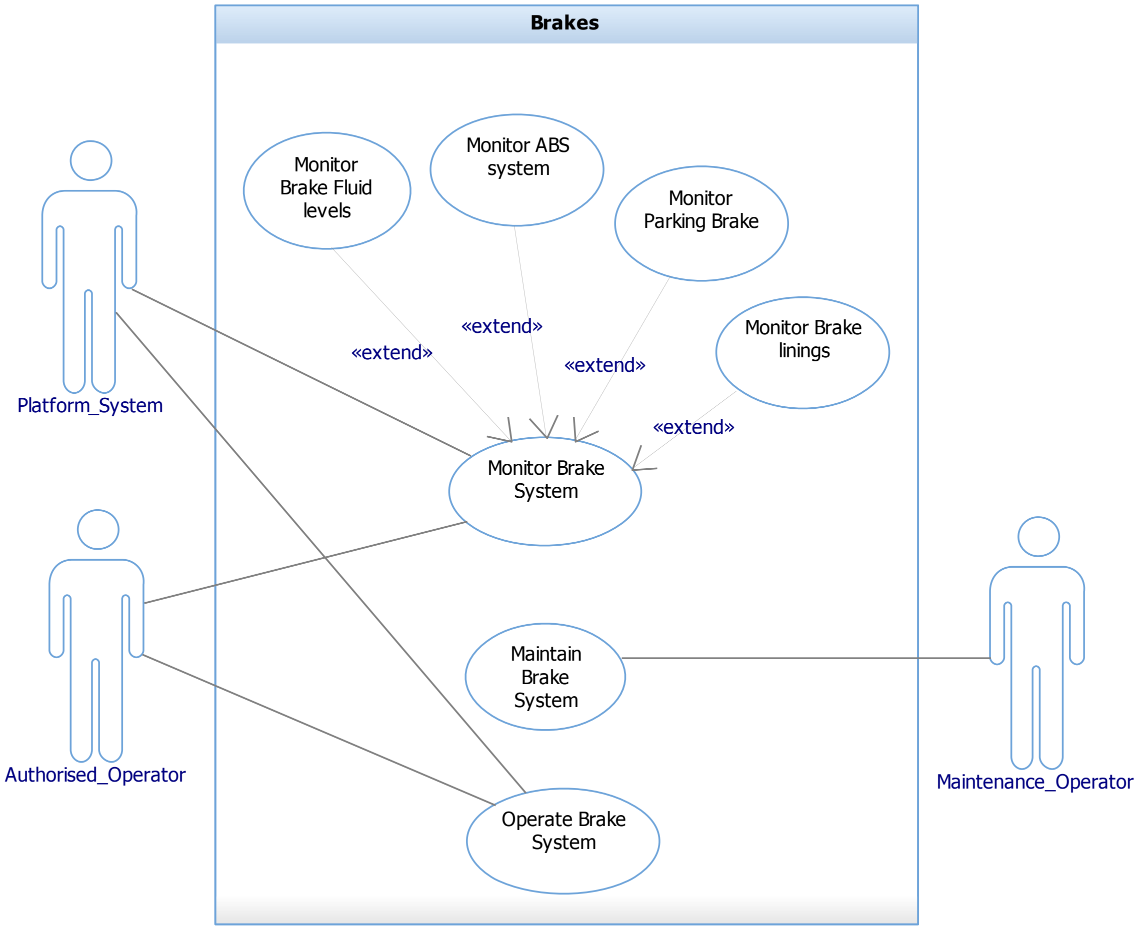


Figure 5.2: NGVA Brakes PIM Use Case Diagram

In addition to class and use case diagrams, the Brakes Module includes two sequence diagrams (Brks_SDxx) specifying to the response to braking commands shown in Figure 5.3.

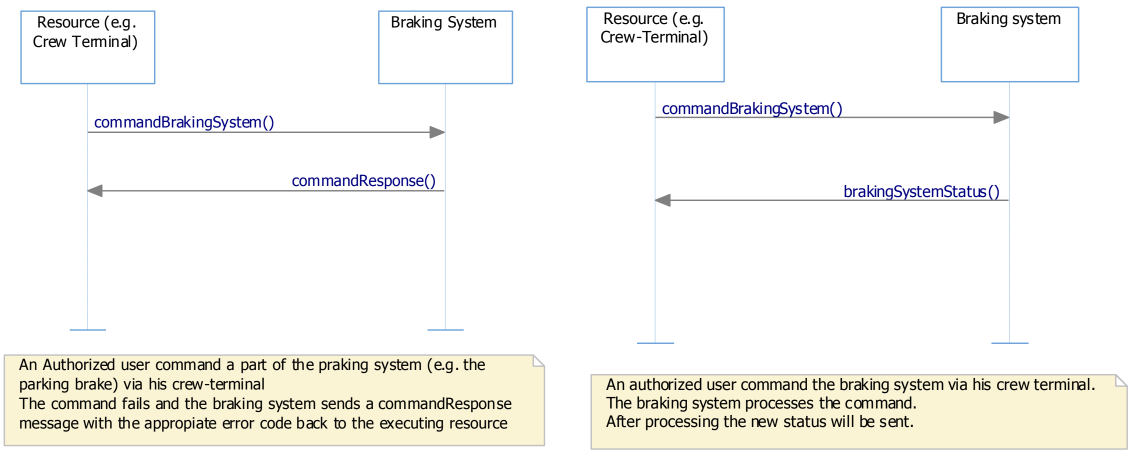
Generated IDL File

In order to specify the test cases and to prepare the basis for software implementations, the class diagram is translated to IDL (cf. Figure 2.5). This results in a couple of Specification (Spc), State (Stt) and Command (Cmd) topics:

- Spc01: C_Brake_Specification
- Spc02: C_Brake_Fluid_Reservoir_Specification

Table 5.1: Use Case Description to Monitor the Brake System

Use Case	Monitor Brake System
Author	Anonymous
Use Case Id	Brks_UC03
Description	An Authorized Operator monitors the braking system via a crew display screen.
Primary Actor	Authorized_Operator
Supporting Actors	Platform_System
Pre-Conditions	The braking system is operating correctly
Post Conditions	The braking system is operating correctly
Trigger	An Authorized Operator requests the status of the braking system via the user interface on his crew display.
Basic Flow	1. The operator requests the braking system information screen on his display.
	2. The operator scrolls through various screens in order to satisfy himself that the braking system is operating correctly.
	3. The operator exits the braking system information screen on his display and returns to previous operating conditions.
Alternative Flows	none



(a) Brks_SC01: Command_Brake_Failed

(b) Brks_SC02: Command_Brake_NoError

Figure 5.3: NGVA Brakes PIM Sequence Diagrams

- Stt01: C_Braking_System
- Stt02: C_Brake_Circuit
- Stt03: C_Brake_Fluid_Reservoir
- Stt04: C_Brake_Lining

- Stt05: C_Engine_Brake
- Stt06: C_Abs
- Cmd01: C_Braking_System_commandParkingBrake
- Cmd02: C_Engine_Brake_commandEngineBrake
- Cmd03: C_Abs_commandAbsStreetLogic

The following Listing 5.1 provides an excerpt of the generated topics.

Listing 5.1: Brakes Module IDL Excerpt

```
module P_Brakes_PSM
{
[...]
   // This provides information about the brake system.
   struct C_Braking_System
   {
      T_IdentifierType A_sourceID; //@key
      T_DateTimeType A_timeOfDataGeneration;
      T_PercentageType A_brakePedalPosition;
      T_PressureInPascalType A_brakeApplicationPressure;
      T_ParkingBrakeStatusType A_parkingBrakeState;
      T_BrakePedalStatusType A_brakePedalState;
      T_IdentifierType A_itsBrake_Specification_sourceID;
      T_IdentifierType A_abs_sourceID;
      T_IdentifierType A_optionalEngineBrake_sourceID;
      T_IdentifierType A_specification_sourceID;
      sequence <T_IdentifierType, 10> A_brakeLining_sourceID;
      sequence <T_IdentifierType, 10> A_brakeFluidReservoir_sourceID;
      sequence <T_IdentifierType, 10> A_brakeCircuit_sourceID;
   };

   // This provides a specification for a Brake Fluid reservoir
   struct C_Brake_Fluid_Reservoir_Specification
   {
      T_IdentifierType A_sourceID; //@key
      T_DateTimeType A_timeOfDataGeneration;
      T_PercentageType A_minLevel;
      T_Boolean A_supportsLevelMeasurement;
      T_IdentifierType A_brakeFluidReservoir_sourceID;
   };

   // This provides information about the brake fluids within the braking system.
   struct C_Brake_Fluid_Reservoir
   {
      T_IdentifierType A_sourceID; //@key
      T_DateTimeType A_timeOfDataGeneration;
      T_BrakeFluidStatusType A_brakeFluidState;
      T_PercentageType A_currentLevel;
      T_IdentifierType A_correspondingBrakingSystem_sourceID;
      T_IdentifierType A_specification_sourceID;
   };
[...]
};
```

5.1.3 Test Specification

The following sections describe the test set-up and test cases derived from the Brakes Module. In order to derive the test cases and test set-up, the methodology discussed in section 4.5.1 was applied.

Test Configuration

After the input analysis, the test environment used to execute the tests was defined and developed. According to section 4.5.1, a brakes system is to be considered as terminal equipment. Therefore, the test configuration should consist of a terminal component representing the brakes system and a second test component controlling and monitoring the brakes system (cf. Figure 4.7).

For the test case execution, the brakes system is implement as a vehicle component in the 3D simulation environment *Virtual Battlespace 3* (VBS 3). A Commercial off-the-shelf (COTS) Command and Control Information System (C2IS) serves as the test application (cf. Figure 5.4). Additionally, a monitoring system is used to log the message exchange as proposed in Figure 4.3.

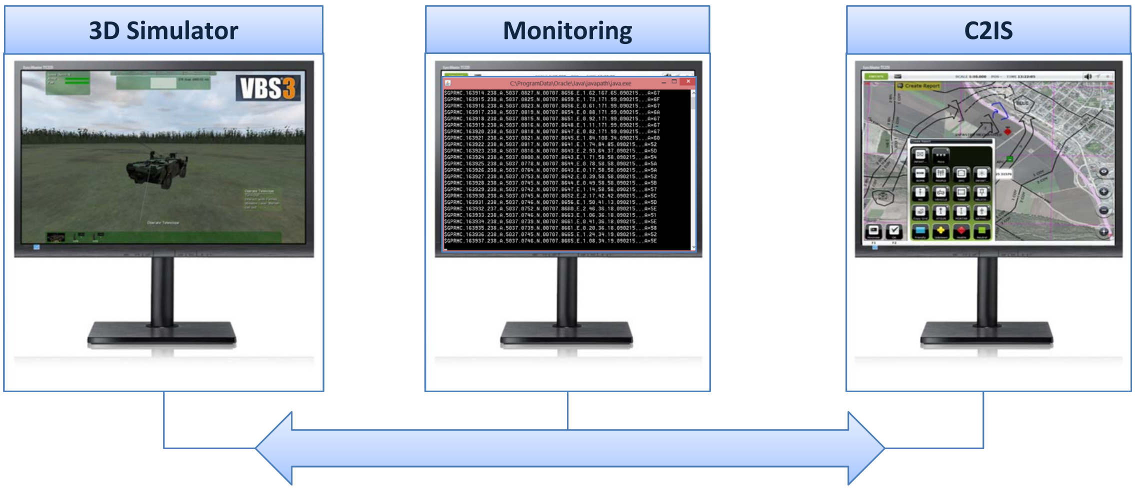


Figure 5.4: Brakes Module MML 5 Testing Configuration

Figure 5.4 shows the set-up and message flow. The Command Topics are published by the C2IS while the rest of the Topic samples is generated by the simulation.

Test Suite Structure

The test suite specified for MML 5 testing is organised by means of the naming convention given in Table 5.2.

Table 5.2: Test Case Naming Conventions

TC/⟨root⟩/⟨gr⟩/⟨nn⟩		
⟨**root**⟩ = root	BRKSTM	Brake System
	BRKCRCT	Brake Circuit
	BRKFLD	Brake Fluid Reservoir
	BRKLNG	Brake Lining
	ENGBRK	Engine Brake
	ABS	Anti-Blocking System
⟨**gr**⟩ = group	CONF	Configure Sub-System
	OPRT	Operate Sub-System
	MNTR	Monitor Sub-System
⟨**nn**⟩ = sequential number		01 to 99

First, a root group is created for each sub-system in order to structure the suite. Then, the test cases are grouped by whether they aim at configuring, monitoring or operating the Brakes system. Indicated by two digits, several test case per subgroup are possible.

Test Cases

After defining the Test Suite Structure, Test Cases for the different topics were derived from the use cases and sequence diagrams of the NGVA DM Brakes Module. In order to cover all use cases and topics, ten use cases have been specified. Table 5.3 provides a brief overview.

Table 5.4 shows an example test case in order to control the brake system. A second a test case for the monitoring use case (cf. Table 5.1) is given in Table 5.5.

Table 5.3: Test Cases Specified for Brakes Module MML 5 Testing

Test Case Identifier	**Test Case Name**
TC/BRKSTM/OPRT/01	Set Parking Brake
TC/BRKSTM/OPRT/02	Release Parking Brake
TC/BRKSTM/OPRT/03	Release Parking Brake although it is already released
TC/ABS/CONF/01	Activate ABS street logic
TC/ABS/CONF/02	Activate ABS off-road logic
TC/ENGBRK/OPRT/01	Activate Engine Brake
TC/ENGBRK/OPRT/02	Deactivate Engine Brake
TC/BRKCRCT/MNTR/01	Receive Brake Circuit states
TC/BRKFLD/MNTR/01	Receive Brake Fluid states
TC/BRKLNG/MNTR/01	Receive Brake Lining states

Table 5.4: Test Case to Control the Brake System

<table>
<tr><th colspan="4">Model Maturity Test Case Description TC/BRKSTM/OPRT/01</th></tr>
<tr><td colspan="2">Identifier:</td><td colspan="2">TC/BRKSTM/OPRT/01</td></tr>
<tr><td colspan="2">Objective:</td><td colspan="2">Set Parking Brake</td></tr>
<tr><td colspan="2">Configuration:</td><td colspan="2">CFG_01</td></tr>
<tr><td colspan="2">References:</td><td colspan="2">Brks_UC02, Brks_SD02, Brks_UC03d</td></tr>
<tr><td colspan="2">Pre-test conditions:</td><td colspan="2">Parking Brake is released.</td></tr>
<tr><td>Test Sequence:</td><td>Step</td><td>Type</td><td>Description</td></tr>
<tr><td rowspan="4"></td><td>1</td><td>Stimulus</td><td>Simulator publishes C_Braking_System.</td></tr>
<tr><td>2</td><td>Check</td><td>C2IS subscribes to C_Braking_System and operator checks if A_parkingBrakeState equals L_ParkingBrakeStatusType_NotSet.</td></tr>
<tr><td>3</td><td>Stimulus</td><td>C2IS publishes C_Braking_System_command ParkingBrake setting A_apply to TRUE.</td></tr>
<tr><td>4</td><td>Verify</td><td>C2IS receives new C_Braking_System sample and operator checks if A_parkingBrakeState equals L_ParkingBrakeStatusType_Set.</td></tr>
</table>

Table 5.5: Test Case to Monitor the Brake System

Model Maturity Test Case Description TC/BRKFLD/MNTR/01			
Identifier:	**TC/BRKFLD/MNTR/01**		
Objective:	Receive Brake Fluid states		
Configuration:	CFG_01		
References:	Brks_UC03		
Pre-test conditions:	none		
Test Sequence:	Step	Type	Description
	1	Stimulus	Simulator publishes C_Braking_System, C_Brake_Fluid_Reservoir, C_Brake_Fluid_Reservoir_Specification, and C_Brake_Specification.
	2	Check	C2IS subscribes to C_Braking_System, receives and displays the values. Operator checks if all values are as assumed.
	3	Check	C2IS subscribes to corresponding C_Brake_Specification, receives and displays the values. Operator checks if A_brakeFluidReservoir_sourceID sequence length of C_Braking_System equals A_brakeFluidSensorCount of C_Brake_Specification.
	4	Check	If at least one brake fluid reservoir exists, C2IS subscribes to C_Brake_Fluid_Reservoir and C_Brake_Fluid_Reservoir_Specification, receives and displays the values of the different reservoirs. Operator checks C_Brake_Fluid_Reservoir_Specification if A_supportsLevelMeasurement is TRUE. Operator checks if all A_currentLevel are about A_minLevel and if T_BrakeFluidStatusType is L_BrakeFluidStatusType_Sufficient.

5.1.4 Traceability Analysis

In order to provide traceability between the data model and the test suite, the use cases were mapped to the topics and it was documented how the generated topics have been covered by the test cases.

Table 5.6: Mapping of Test Cases to Use Cases and Sequence Diagrams

Test Cases	**Use Case/Sequence Diagrams**								
	UC01	UC02	UC03	UC03a	UC03b	UC03c	UC03d	SD01	SD02
TC/OPRT/BRKSTM/001		■	■				■		■
TC/OPRT/BRKSTM/002		■	■				■		■
TC/OPRT/BRKSTM/003		■	■				■	■	
TC/OPRT/ABS/001		■	■	■					■
TC/OPRT/ABS/002		■	■	■					■
TC/OPRT/ENGBRK/001		■	■						■
TC/OPRT/ENGBRK/002		■	■						■
TC/MNTR/BRKCRCT/001	■		■						
TC/MNTR/BRKFLD/001	■		■		■				
TC/MNTR/BRKLNG/001	■		■			■			

Table 5.7: Mapping of Test Cases to Topics

Test Cases	**Topics**										
	Spc01	Spc02	Stt01	Stt02	Stt03	Stt04	Stt05	Stt06	Cmd01	Cmd02	Cmd03
TC/OPRT/BRKSTM/001			■						■		
TC/OPRT/BRKSTM/002			■						■		
TC/OPRT/BRKSTM/003			■						■		
TC/OPRT/ABS/001	■		■					■			■
TC/OPRT/ABS/002	■		■					■			■
TC/OPRT/ENGBRK/001	■		■				■			■	
TC/OPRT/ENGBRK/002	■		■				■			■	
TC/MNTR/BRKCRCT/001	■		■	■							
TC/MNTR/BRKFLD/001	■	■	■		■						
TC/MNTR/BRKLNG/001	■		■			■					

Table 5.6 allows to check if there are any topics existing in the IDL, which are not covered by Use Cases or Sequence Diagrams. This either indicates that the set of use cases is not complete or that unnecessary topics have been defined.

Table 5.7 gives confirmation that each topic is covered by at least one test case, which is required by the definition of MML 5.

5.1.5 Test Summary

The execution of the different test cases led to topic-specific recommendations for improvements of the Brakes PIM. For example, with respect to the test case presented in Table 5.4, it was recommended to change the 1:1-relationship between *C_Brake_Fluid_Reservoir_Specification* and *C_Brake_ Fluid_Reservoir* into a 1:n-relationship since many reservoirs can share the same specification (cf. Figure 2.5).

These recommendations were captured in a test report consisting of organisational information and observations noticed during testing. The following subsections provide some excerpts of the findings.

Organisational Summary

- Organisation: MML 5 Testing has been conducted in the NGVA-WTD 41-Testlab
- Involved personnel: MML 5 Testing has been supervised by Daniel Ota, FKIE.
- Date: MML 5 Testing has been executed on XX September 2016
- Version of the Brakes PIM: LDM Revision 3019
- Translator Version: GVA PIM Translator, User Version 1.4 without modification of parameters/properties.
- DDS vendor and version: RTI Connext 5.2.3
- DDS QoS parameters: QoS patterns as specified in NGVA DM AEP volume

General Recommendations

Two types of observations have been noticed: issues regarding the translation process and recommendations to improve the Brakes Module. This subsection summarises some general observations noticed during the implementation and execution of the test cases within the previously described test scenarios.

The IDL was generated by means of the GVA PIM Translator Version 1.4 using default properties without modifications. The generation process resulted in two IDL files: a *LDM_Common.IDL* file containing types, which are jointly used by all domain specific modules, and a *Brakes_PSM.IDL* file, which is holding types, enumeration and topics that represent specific brake related features.

Recommendation R01:

In order to re-use the common types in the brakes IDL, the common file has to be included via *#include "LDM_Common.IDL"* at the beginning of the *Brakes_PSM.IDL* file:

Modify GVA PIM Translator to insert *#include "LDM_Common.IDL"* at the beginning of all domain specific IDL files generated.

Further improvements with regard to the structure of the resulting topics have been proposed:

Recommendation R02:

All relationships between specification and state classes are bidirectional. This induces a huge implementation overhead since sequences in both classes (or in the topics as their realisations) have to be maintained and induces also duplication of data to be transmitted. In fact, specification topics do not need to know about the states they are specifying.

Recommendation R03:

All commands obtain the attributes *A_recipientID*, *A_sourceID* and *A_instanceID*. It should be clarified and specified that recipient resource ID, source resource ID and recipient instance ID should be used for topic samples. Additionally, the purpose and usage of the attribute *A_referenceNum* is not specified.

Topic Specific Recommendations

The execution of the different test cases led to additional topic specific recommendations. This section presents findings related to the two previously presented test cases and to the topics presented in the IDL snippet.

TC/BRKSTM/OPRT/001-003

These test cases covered controlling the parking brake system as well as monitoring the brake system status.

Recommendation R04:

C_Braking_System contains the attributes *A_brakePedalPosition* for electric brakes and *A_brakeApplicationPressure* for air/fluid brakes to indicate the braking intensity. In order to inform, which attribute contains actual correct values the topic *C_Brake_Specification* should declare the type of braking system.

Recommendation R05:

C_Brake_Specification contains a sequence of instance IDs *A_specifiedBrakingSystem_instanceID* referring to different braking system instances, which are specified by the topic. It should be scrutinised if a single vehicle would contain more than two braking system instances defined by the same specification (all of them sharing the same *resourceID*) or these are rather 1:1..2 relationships between *C_Brake_Specification* and *C_Braking_System* topics.

Recommendation R06:

C_Braking_System contains an attribute *A_itsBrake_Specification_instanceID* and another one called *A_specification_instanceID*. Both seem to have the same purpose and thus one should be removed.

TC/BRKFLD/MNTR/001

This test case covered the monitoring of the brake fluids status.

Recommendation R09:

There is 1:1-relationship between *C_Brake_Fluid_Reservoir_Specification* and *C_Brake_Fluid_Reservoir*. It is recommended to change to 1:n since many reservoirs can share the same specification.

5.2 Brakes Module Conformance Testing

This section discusses conformance testing of the NGVA Brakes Module. Based on the approach designed in section 4.5.2, test cases from Data Model maturity testing in section 5.1 were re-used and extended for the development of NTRS Test Suite. Further, the MML 5 simulator implementation was applied for the first test execution of the newly developed NTRS. Obviously, there was nothing to re-use from MML 5 testing for the implementation of the actual conformance test system, since it was entirely based on the existing MTRS.

5.2.1 NGVA Test Suite

In order to test compliance of future NGVA Brakes Module implementations, an NGVA Test Suite has been developed, which so far includes 19 test cases for the NGVA DM Brakes Module. The test cases cover all three test categories: logic-, link-, and specification-based tests.

In the case of logic-based test cases, particular attention was paid to correctly implemented command-state interactions, which occur for example, when activating or deactivating different brake sub-systems (parking brake, anti-lock brake system, etc.). In this case, it is checked whether the status message sent afterwards corresponds to the expected response to the command. For link tests, the instance identities (sourceId and recipientId) of the referring brakes topics are compared. In the case of specification tests, it is checked whether the corresponding system status messages reflect the specification.

This includes, for example, the comparison of the number of specified brake circuits with the actual transmitted brake circuit instances.

Table 5.8: Realised NTRS Test Cases based on MML 5 Test Cases

Test Category	**NTRS Test Case**	**MML 5 Test Case ID**
Link-Based	Test link between Braking System and Abs	–
	Test link between Braking System and Engine Brake	–
	Test link between Braking System and Brake Specification	–
	Test link between Braking System and Brake Lining	TC/BRKLNG/MNTR/01
	Test link between Braking System and Brake Circuit	TC/BRKCRCT/MNTR/01
	Test link between Braking System and Brake Fluid Reservoir	TC/BRKFLD/MNTR/01
	Test link between Brake Fluid Reservoir and Brake Fluid Reservoir Specification	TC/BRKFLD/MNTR/01
Logic-Based	Set Parking Brake	TC/BRKSTM/OPRT/001
	Release Parking Brake	TC/BRKSTM/OPRT/002
	Release Parking Brake although already released	TC/BRKSTM/OPRT/003
	Activate ABS street logic	TC/ABS/CONF/001
	Activate ABS off-road logic	TC/ABS/CONF/002
	Activate Engine Brake	TC/ENGBRK/OPRT/001
	Deactivate Engine Brake	TC/ENGBRK/OPRT/002
	Reflect Set Parking Brake	TC/BRKSTM/OPRT/001
	Reflect Activate ABS Street Logic	TC/ABS/CONF/001
	Reflect Activate Engine Brake	TC/ENGBRK/OPRT/001
Specification-Based	Test Brake Specification	TC/BRKCRCT/MNTR/01
	Test Brake Fluid Reservoir Specification	TC/BRKFLD/MNTR/01

Table 5.8 provides an overview of the 19 realised test cases and – if based on – the re-used test cases from the previous section. It can be seen that the MML test cases 5 provided valuable input and a good basis for the specification of conformance test cases. All MML 5 test cases related to the operation, configuration or monitoring of the brake (sub-) systems resulted directly in logic- or specification-based CT test cases. This is logical, since for MML 5 testing mainly functional test cases have been defined.

The test cases defined in the link-based category check if two topic instances are correctly linked with the same resourceIDs and instanceIDs. Therefore, these tests cover correct usage of the data structures rather than testing functionality. If corresponding monitoring test cases have been defined for MML 5 testing, these can be re-used. If not additional, new test conformance test cases have to be created. However, this is not problematic at all, since link-based tests can be directly derived from the UML data model itself. Basically, each association between two classes results in link-based test cases.

One example of a re-used test case is presented in Figure 5.5. It provides a simplified flow chart of the *Test link between Brake Fluid Reservoir and Brake Fluid Reservoir Specification* test script of the NTRS Test Suite. It represents a link test based on the class association shown in Figure 2.5. The script was generated by detailing Step 3 of the brake monitoring Test Case *TC/BRKFLD/MNTR/01* presented in Table 5.5. It still checks the criticised 1:1-relationship since the test scripts were based on the draft NGVA Brakes Module used for MML 5 testing. The corresponding Test Case script is presented in Listing 5.2.

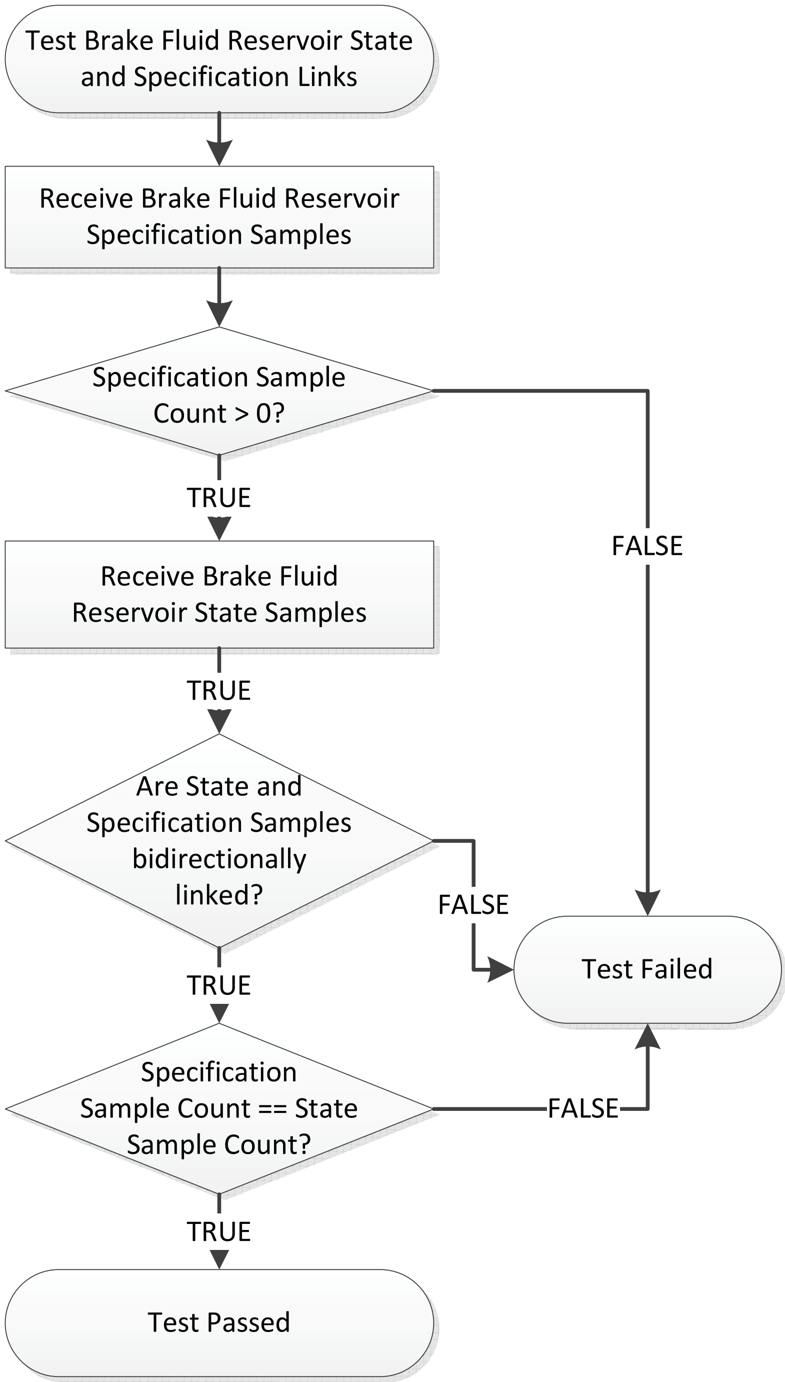


Figure 5.5: Flow Chart of an Example Test Case

Listing 5.2: Test Script for Links between Brake Fluid Reservoir and its Specification

```
/*
 * @Id
 *  3301010107
 * @Version
 *  1.0
 * @Purpose
 *  Test links between Brake_Fluid_Reservoir and
    Brake_Fluid_Reservoir_Specification
 * @Reference
 *  MML 5 Test Case TC/BRKFLD/MNTR/001
 */

 testcase Test_the_link_between_BFR_and_BFRS {

  Map<Integer, Brake_Fluid_Reservoir_Specification_Message>
      fluidReservoirSpecifications = new HashMap<Integer,
      Brake_Fluid_Reservoir_Specification_Message>();
  ddsGateway.createDataReader(Brake_Fluid_Reservoir_Specification,
      QoS.SPECIFICATION_PATTERN, 50, 20);  //wait 20sec for 50
      instances
  [?] validator.receive(Brake_Fluid_Reservoir_Specification_Message
      spec) from ddsGateway in 20000 {
   fluidReservoirSpecifications.put(spec.ddsSample.A_instanceID,
       spec);
   repeat;
  }
  on timeout {
   if (fluidReservoirSpecifications.size() == 0) {
    logInfo("No Brake Fluid Reservoir Specification Message received
        ");
    return Verdict.Fail;
   }
  }

  int instance_count = 0;
  int instance_expected = fluidReservoirSpecifications.size();
```

```
  ddsGateway.createDataReader(Brake_Fluid_Reservoir, QoS.
     STATE_PATTERN, instance_expected);
  [?] validator.receive(Brake_Fluid_Reservoir_Message mes) from
     ddsGateway  in 10000 {
   int correspondingSpec = mes.ddsSample.A_specification_instanceID;
   if (!fluidReservoirSpecifications.containsKey(correspondingSpec))
       {
    return Verdict.Fail;
   }
   Brake_Fluid_Reservoir_Specification_Message spec =
      fluidReservoirSpecifications.get(correspondingSpec);
   assertEquals("instanceID link", spec.ddsSample.
      A_brakeFluidReservoir_instanceID , mes.ddsSample.A_instanceID
      );
   ++instance_count;
   repeat;
  }

  on timeout {
   if (instance_count != instance_expected) {
    logInfo("Unexpected number of Brake_Fluid_Reservoir_Message 
       received");
    return Verdict.Fail;
   }
  }
  return Verdict.Pass;
 }
```

5.2.2 Initial NTRS Conformance Testing

The Brakes Module Reference Implementation for the virtual simulator and the C2IS presented in section 5.1 have been successfully tested with the NTRS. Therefore, the specified test cases have been tested against the 3D-simulation-based vehicle implementation of the Brakes system. The implementation was connected to the NTRS and all test cases were run in different scenarios. The conducted tests revealed different problems in the simulation implementation, which are summarized in Table 5.9.

Table 5.9: Examples of Identified Problems during Acceptance Testing

Test Category	**Identified Issue**
Link-Based	Referencing topics were not updated after a referenced topic has been deleted or disposed.
	Topics did not correctly refer to each other (i.e. by specifying the full IdentifierType).
Logic-Based	Topic members including senderId, timeOfDataGeneration, etc. were empty.
Specification-Based	The number of entities defined in the specification topic did not match with the number of linked topics instances.

By means of the NGVA DM Test Reference System, it was possible to identify weaknesses in the the MML 5-based simulation implementation in order to have a precise Reference Implementation for future usage in Interoperability and Acceptance Testing.

5.3 NGVA Interoperability Testing

This section gives an overview of the NGVA interoperability test laboratory designed according to the architecture described in section 4.5.3. Besides additional implementations from other studies, also all Reference Implementations developed for MML 5 testing are deployed there. In addition, the maturation tests as discussed in section 4.5.1 have been conducted in the NGVA interoperability test lab. For the purpose of future acceptance testing, existing monitoring and management tools have been installed and are re-used.

5.3.1 Re-Use of MML 5 Test Implementations and Configurations

Figure 5.6 provides an overview of the test configuration relevant for this thesis containing artefacts from the six MML 5 tests, which were carried out. The tests include the analysis of the following PIM modules:

- Brakes,
- Human Machine Interface (HMI) Input Devices,

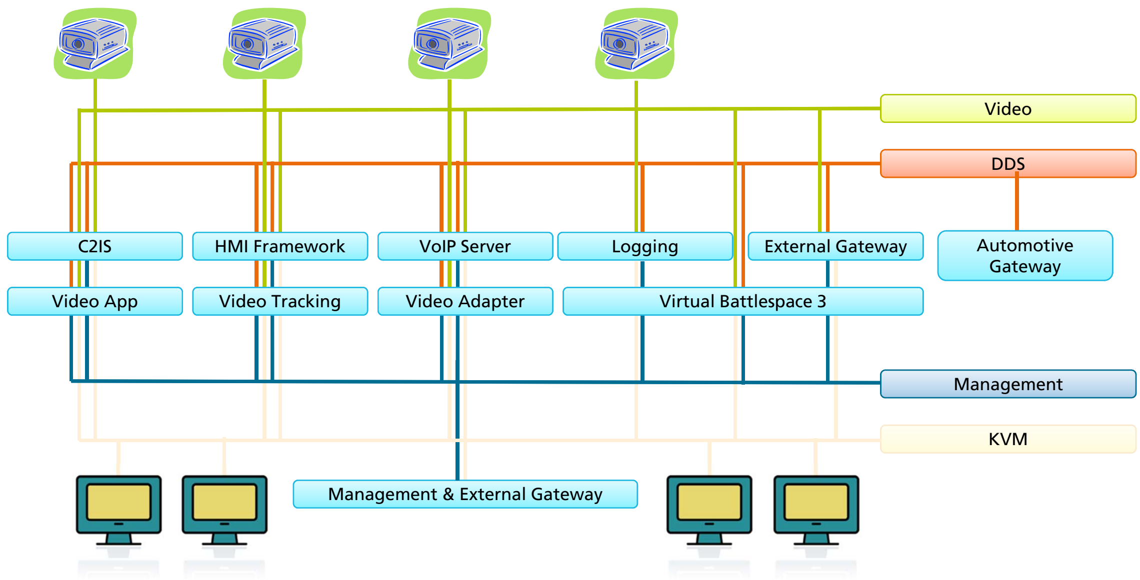


Figure 5.6: NGVA Interoperability and Acceptance Test Laboratory

- Human Machine Interface (HMI) Presentation,
- Laser Rage Finder,
- Video, and
- Video Tracking.

Depending on the intensity and scope of the particular MML 5 test activity, the integration was more or less complex.

Since the Brakes and Laser Range Finder artefacts are based on the same components, their integration was conducted in combination. The Brakes Module realisation – as implemented for MML 5 testing – was integrated into the existing vehicle simulation in Virtual Battlespace 3 and the C2IS. Thus, the simulated NGVA-based vehicle of the test lab provides now brake status information while driving around in the simulation. In contrast to the Brakes Module, integrating the LRF Module implementation was more challenging, since it was not possible to use the Laser Range Finder directly in the C2IS. This required the implementation of further NGVA DM modules such as *Navigation* and *Mount* in order to supply the C2IS with current vehicle position, vehicle bearing, and turret direction for calculating target data (cf. Figure 5.7).

The artefacts resulting from the two HMI tests were deployed to other network nodes and form now the basis to test applications realised according the NGVA CTSA AEP

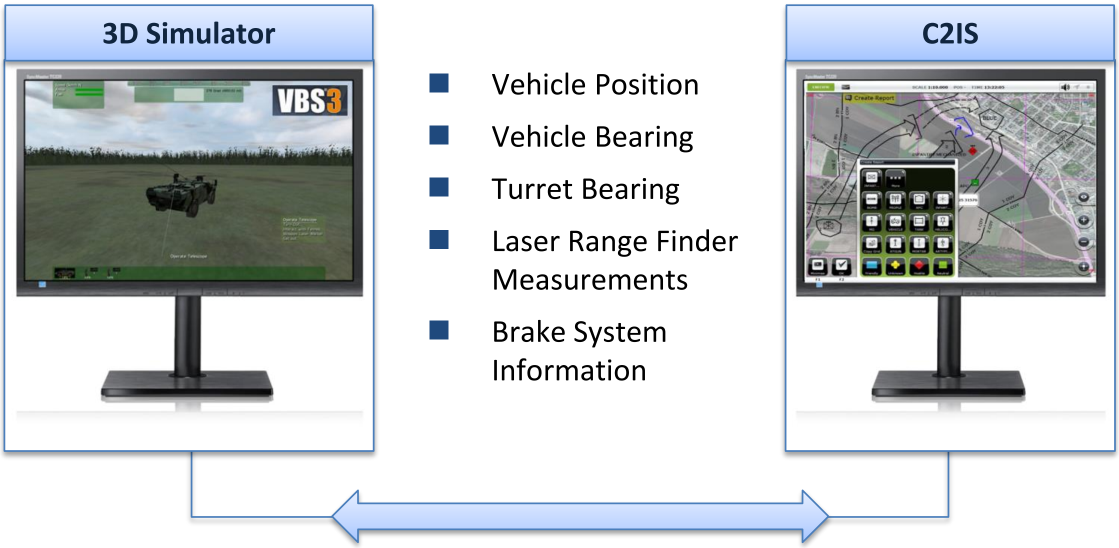


Figure 5.7: Integration of Laser Range Finder and Brakes Module Artefacts

volume. Video and Video Tracking applications are used along with Commercial off-the-shelf (COTS) cameras that stream video data according to the GigE Vision and PLEVID specifications, which is one option to integrate cameras NGVA-compliant (cf. Figure 2.1).

Beside actual artefacts from MML 5 testing, the test lab is equipped with gateways in order to connect to other networks. The external gateway allows Virtual Private Network connections to other NGVA test labs for conducting joint testing sessions. To connect automotive networks such as CAN-based ones, automotive gateway on the basis of a Raspberry Pi was provided. It allows the integration of COTS automotive components [66].

5.3.2 Conducting NGVA Interoperability and Acceptance Testing

In order to facilitate interoperability and acceptance testing of a wide range of NGVA-based components, the monitoring application from the MML 5 testing phase was seen as perfect starting point to develop a flexible test tool (cf. section 4.2.3). For determining the needed capabilities, use cases have been developed with personnel from the German test centre for military vehicles.

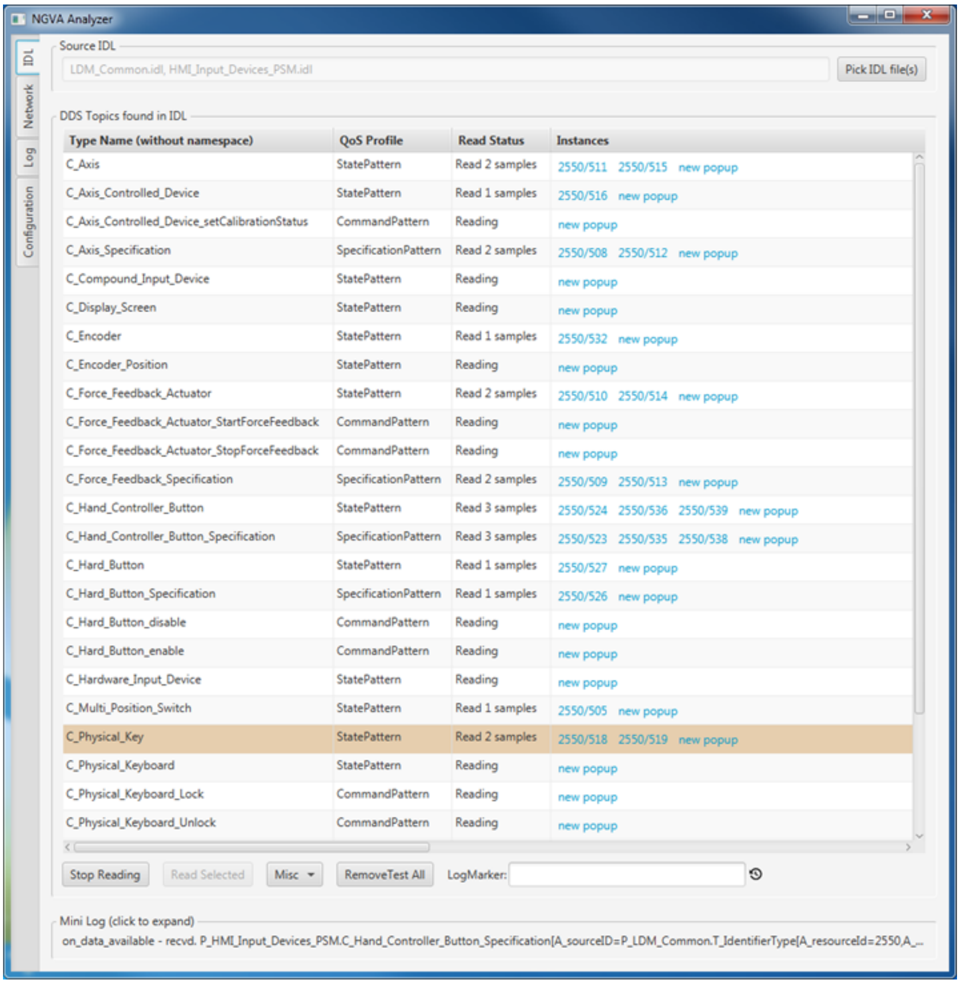


Figure 5.8: Extended NGVA Acceptance Test Tool

On this basis, the MML 5 monitoring tool was extended to allow a manual testing of NGVA compliance (cf. Figure 5.8). After loading the IDL files implemented in the SUT, NGVA-compliant DDS entities such as data readers and data writers can be dynamically created by the test operator. Received DDS samples can be inspected and modified if needed. Additionally, the tool analyses QoS matchings of tool internal and external DDS entities (e.g. of the SUT) to detect incorrect settings.

During the test of a SUT component, any event happening is recorded. This includes DDS network events such as data available in readers and changed liveliness status or application internal events and process messages such as IDL compilation or entity creation. Besides, the test operator can annotate the tests with custom log entries.

By means of scripting support, all DDS data exchange or only the exchange initiated by the test operator can be recorded. Thereby, a playback functionality is realised, which allows saving, loading, executing and stopping (sub-) system specific test scripts. These functionalities are especially helpful, if regression testing due to incorrect implementations is necessary during acceptance testing.

As discussed in section 4.5.3, automated acceptance testing is almost impossible, since it does not only rely on the specification but also on specific user requirements with respect to functionality or QoS – typically defined in accompanying System Requirements Documents. Nonetheless, the tool is a huge step forward towards efficient acceptance testing.

Thus, it is planned to extend it further with a comprehensive test suite environment, more (semi-) automated testing and test report generation functionalities. Besides, realised extensions refer only to the exchange of the messages defined by the NGVA Data Model so far. Further data exchange formats defined in the NGVA, such as audio or video standards to be implemented are not yet covered. In order to carry out these functional or performance tests, additional tools are necessary. In the case of a camera, for example, additional tools supporting the testing might support

- Accessing and assessing the video stream (sharpness, contrast, etc.),
- Measuring the latency of the camera and the control / status messages, or
- Recording and evaluating the network load of the video streams.

Beyond, it is also worth mentioning, that the new monitoring tool functionalities can be used out of the box for MML 5 testing of further DM modules. Additionally, future MML 5 test case sequences and test logs can be directly re-used as a basis acceptance tests again. Thus, the improved monitoring tool is a good example how artefacts can be shared between MML 5 and acceptance testing.

5.4 Conclusion

This chapter analysed to which extent an artefact re-use for the Specification, Interoperability and Conformance Testing phases is possible and reasonably practical. By example of the NGVA Brakes Module, it was shown that test case specifications can be adapted and transferred between specification and conformance testing. Similarly, Reference Implementations can be re-used with little effort between specification and interoperability/acceptance testing phases. Further, all phases support each other by gaining experience and reducing the probability of poor sub-system realisations.

The presented MML 5 case study was supported by the German Ministry of Defence under the research grants *NGVA Verification* and *Interoperability Test Methods for Future Military Land Vehicles*. The case study has been published at an IEEE-listed conference [67]. The evaluation of the conformance test suite was conducted in collaboration with Ditmir Hazizi at Fraunhofer FKIE. The requirements analysis and implementation of the NGVA monitoring tool improvements were supported by Reinhard Claus.

6 Conclusions

The complexity of military systems in the land domain gains rapidly. The number of sensors and effectors installed on them and the amount of data exchanged is constantly growing. In contrast, the interconnectedness restricts more and more the upgradability, since dependencies between sub-systems intensified.

In order to overcome these problems, Open System Architectures have been introduced to standardise sub-system interfaces and behaviour. This facilitates the exchangeability of components, since interfaces are no longer proprietary and controlled by a single manufacturer or contractor.

Rising complexity and increasing dependencies also complicate the systems Verification and Validation. Side-effects are more likely to occur during upgrades in data centric architectures, since information exchange is no longer pre-determined. Thus, adapted V&V approaches need to be defined. In the case of internationally agreed standards and specifications, these test procedures also need to be harmonised to ensure equal level of testing.

Furthermore, the earlier V&V is applied to the system development, the more financial costs and time to market can be reduced. In order to support this, verification tools need to be on hand as early as possible.

On the example of the NGVA STANAG, the contributions of the thesis provide concepts, tools and test suites to de-risk system implementations and to verify systems as early as possible.

6.1 Thesis Contributions

The first contribution of the thesis is the first internationally agreed V&V approach for military land vehicles based on Open System Architectures. It provides guidance to develop a verification plan for sub-systems to be evaluated. Therefore, best practices and standards from the V&V domain have been analysed and adapted for alignment with the NGVA specification. NGVA Compatibility Levels have been developed in order to support ordering the requirements for verification execution. Following this methodology allows gradual evaluation of NGVA sub-systems and ensures that requirements providing the basis for other requirements are verified beforehand. Based on the NASA Systems Engineering Handbook, a verification process considering the verification plan and the NCLs was developed. The results of the first contribution form the main input of the NGVA Verification and Validation AEP. The AEP was approved by NATO nations and is part of the official NGVA STANAG release.

The second contribution comprises a test framework supporting verification activities during the full standard and system development process. It addresses the three stages: Specification, Conformance and Interoperability Testing. The framework incorporates the development of testing processes and test specifications including test architectures and test suites. Based on the draft NGVA standard, first test procedures for the specification – especially the NGVA DM – were derived and described in form of a data model maturation process. Afterwards, it is analysed and indicated, which artefacts can be re-used and shared between the three different verification phases. This approach particularly lowers the development risks of early systems based on the standard. Following the approach, first test tools and suites are already available when the specification is approved, since resulting artefacts from specification testing can be re-used.

The third contribution is a case study analysing the extent of artefact re-usability. Therefore, the framework is applied to the NGVA Brakes Module. After input analysis, the actual test architecture and test suite for the DM module under test is derived. In the subsequent phases of the case study, it was shown that the test suite can be re-used for conformance testing. Based on the ten test cases from maturation process, a conformance test suite containing 19 test cases has been derived. For interoperability testing, in particular the test architecture and software artefacts have been re-used as Reference Implementations. Further, the re-use and enhancement of the original logging and monitoring tool was particularly found valuable.

6.2 Future Work

The NGVA STANAG and the accompanying DM have been released in February 2018. So far, no actual NGVA-compliant (sub-)systems have been realised and could be tested with the elaborated V&V approach. Even though interoperability testing of a GPS receiver has been conducted, the applicability of the entire AEP volume to real systems needs further evaluation. In addition, the presented approach is mainly concentrating on verification, since there are no operational requirements defined in NGVA. As soon as this changes, more focus on validation is needed.

The existing NGVA requirements are expected to change in the future. As discussed, they are not yet fully specified and contain no verification methods, measures of performance or even justification. In addition, further requirements from other AEPs such as Safety, Security, and HUMS will be added to the STANAG. Old and new requirements must be classified and ordered to match the NCLs. Thus, guidance needs to be developed for requirements grouping.

The specified MML 5 test approach is accepted and proven. It has already been applied for the maturating of six NGVA DM modules by the author and it has further been used by other nations to mature their NGVA DM baseline modules. This is different for the Conformance and Interoperability Testing approaches. Similar to MML 5 testing, action has to be taken for the generation of conformance test suites for further modules concerning analysis and prove of the discussed re-usage of test suites.

As presented, some link-based conformance test cases could not be derived from the module maturation phase, but they can directly be derived from the UML model. In order to efficiently derive test suites, it should be examined if this process can be a automated.

Since GVA and NGVA share the same basic concepts, all thesis contributions can be applied to GVA as well. Besides, the GVA Data Model provides further basis for adjacent domains such as soldier and base architectures. The thesis contributions provide potential to be exploited for these domains, too. Although probably using other data exchange standards and protocols, the principal approach to early de-risk data model implementations and to re-use developed artefacts might be adopted. This needs to be checked in further research.

Bibliography

[1] Ministry of Defence. *Defence Standard 23-09 – Generic Vehicle Architecture. Issue 1.* Defence Equipment and Support, UK Defence Standardization, Aug. 2010.

[2] NATO. *STANAG 4754, NATO Generic Systems Architecture (NGVA) for Land Systems, Edition 1.* NATO Standardization Office (NSO), Feb. 2018.

[3] NATO. *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Volume III: Data Infrastructure, Edition A, Version 1.* NATO Standardization Office (NSO), Feb. 2018.

[4] NATO. *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Volume V: Data Model, Edition A, Version 1.* NATO Standardization Office (NSO), Feb. 2018.

[5] Open Systems Joint Task Force. *Modular Open Systems Approach to Acquisition, Version 2.* US Department of Defence, Sept. 2004.

[6] Command and Control Initiative. *Open Mission Standard, Version 1.0.* Apr. 2014.

[7] Thomas Gaska. 'Lockheed Martin Common Cockpit Avionics Suite – A Modular Open Systems Approach (MOSA) Driven Architecture'. In: *American Helicopter Society (AHS) Forum 63* (May 2007).

[8] The Open Group. *Technical Standard for Future Airborne Capability Environment (FACE), Edition 2.1.* Reference C145. The Open Group, May 2014.

[9] Mike Southworth. *Delivering VICTORY and PNT Hub Services for Tactical Ground Vehicle Architecture.* White Paper. Curtiss Wright Defense Solutions, Oct. 2015.

[10] Ministry of Defence. *Defence Standard 23-09 – Generic Vehicle Architecture. Issue 3.* Defence Equipment and Support, UK Defence Standardization, Mar. 2013.

[11] IEEE. 'IEEE Standard for System and Software Verification and Validation'. In: *IEEE Std 1012-2012 (Revision of IEEE Std 1012-2004)* (May 2012), pp. 1–223. DOI: 10.1109/IEEESTD.2012.6204026.

[12] ISO. *ISO/IEC/IEEE 15288:2008, Systems and software engineering – System life cycle processes*. International Organization for Standardization, 2008. URL: http://www.iso.org/iso/catalogue_detail?csnumber=63711.

[13] ISO. *ISO/IEC/IEEE 29148:2011, Systems and software engineering – Life cycle processes – Requirements engineering*. International Organization for Standardization, 2011. URL: http://www.iso.org/iso/catalogue_detail.htm?csnumber=45171.

[14] ISO. *ISO/IEC 17000:2004, Conformity assessment – Vocabulary and general principles*. International Organization for Standardization, 2004. URL: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=29316.

[15] AUTOSAR. *AUTOSAR BSW & RTE Conformance Test Specification*. AUTOSAR GbR, Apr. 2011.

[16] Alain Gilberg et al. 'Conformance Testing for the AUTOSAR Standard'. In: *Embedded Real Time Software and Systems*. May 2010.

[17] One Box Consortium ACPO ITS Working Group. *One Box: Single Vehicle Architecture Criteria*. Home Office Centre for Applied Science and Technology, Sept. 2012.

[18] John Rushby. *Modular Certification*. NASA Contractor Report CR-2002-212130. NASA Langley Research Center, 2002.

[19] National Aeronautics and Space Administration. *NASA Systems Engineering Handbook*. SP-610S. PPMI, June 1995.

[20] The Open Group. *Future Airborne Capability Environment (FACE): Conformance Policy, Version 1.1*. Reference X1406. The Open Group, July 2014.

[21] Stephen Walters and Cara Perks. *Test, Evaluation & Accpetance Best Practice*. White Paper. BMT Group, 2011.

[22] ETSI Guide. *Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Generic approach to interoperability testing*. ETSI EG 202 237 V1.2.1. European Telecommunications Standards Institute, Aug. 2010. URL: http://www.etsi.org/deliver/etsi_eg/202200_202299/202237/01.02.01_60/eg_202237v010201p.pdf.

[23] ETSI Guide. *Intelligent Transport Systems (ITS); Testing; Framework for conformance and interoperability testing*. ETSI EG 202 798 V1.1.1 (2011-01). European Telecommunications Standards Institute, Jan. 2011. URL: http://www.etsi.org/deliver/etsi_eg/202700_202799/202798/01.01.01_60/eg_202798v010101p.pdf.

[24] Feng Luo et al. 'Research On CAN controller conformance test system'. In: *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*. Aug. 2009, pp. 582–585. DOI: 10.1109/ICCSIT.2009.5234629.

[25] FlexRay Consortium. *FlexRay Communications System Protocol Conformance Test Specification*. Version 3.0.1. Oct. 2010.

[26] W. Lawrenz. 'Communication Protocol Conformance Testing - Example LIN'. In: *Vehicular Electronics and Safety, 2006. ICVES 2006. IEEE International Conference on*. Dec. 2006, pp. 155–162. DOI: 10.1109/ICVES.2006.371574.

[27] D. Wang, J. Kuang and W. Tan. 'Conformance testing for the car lights system based on AUTOSAR standard'. In: *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*. May 2011, pp. 345–348. DOI: 10.1109/ICCSN.2011.6013608.

[28] AUTOSAR. *Overview of Acceptance Tests*. AUTOSAR GbR, Oct. 2015.

[29] Nico Bau, Michael Gerz and Michael Glauer. 'Testing C2 Interoperability – Advancements in Testing of the MIP Baseline 3 Solution'. In: *ICCRTS 2008: 13th International Command and Control Research and Technology Symposium*. June 2008.

[30] Christian Weiss. 'V2X communication in Europe – From research projects towards standardization and field testing of vehicle communication technology'. In: *Computer Networks* 55.14 (2011), pp. 3103–3119. ISSN: 1389-1286. DOI: http://dx.doi.org/10.1016/j.comnet.2011.03.016. URL: http://www.sciencedirect.com/science/article/pii/S1389128611001198.

[31] H. Stübing et al. 'simTD: a car-to-X system architecture for field operational tests [Topics in Automotive Networking]'. In: *IEEE Communications Magazine* 48.5 (May 2010), pp. 148–154. ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5458376.

[32] Andreas Festag, Long Le and Maria Goleva. 'Field Operational Tests for Cooperative Systems: A Tussle Between Research, Standardization and Deployment'. In: *Proceedings of the Eighth ACM International Workshop on Vehicular Inter-networking*. VANET '11. Las Vegas, Nevada, USA: ACM, 2011, pp. 73–78. ISBN: 978-1-4503-0869-4. DOI: 10.1145/2030698.2030710. URL: http://doi.acm.org/10.1145/2030698.2030710.

[33] Nikia Williams et al. 'TARDEC's VICTORY Systems Integration Laboratory (SIL) is a Key Tool for Advancing Standardized Ground Vehicle Electronic Architecture'. In: *2012 Ground Vehicle Systems Engineering and Technology Symposium*. 2012.

[34] ISO. *ISO/IEC 9646:1994, Information technology – Open Systems Interconnection – Conformance testing methodology and framework*. International Organization for Standardization, 1994. URL: http://www.iso.org/iso/catalogue_detail.htm?csnumber=17473.

[35] ITU. *ITU-T Recommendation X.290, OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts*. International Organization for Standardization, 1995. URL: https://www.itu.int/rec/T-REC-X.290-199504-I/en.

[36] ETSI Technical Specification. *Methods for Testing and Specification (MTS); Conformance Test Specification for SIP (IETF RFC 3261)*. ETSI TS 102 027-1 V4.1.1 (2006-07). European Telecommunications Standards Institute, July 2006. URL: http://www.etsi.org/deliver/etsi_ts/102000_102099/10202701/04.01.01_60/ts_10202701v040101p.pdf.

[37] ETSI Technical Specification. *Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT): IPv6 Core Protocol; Conformance Test Suite Structure and Test Purposes (TSS&TP)*. ETSI TS 102 515 V1.1.2 (2008-01). European Telecommunications Standards Institute, Jan. 2008. URL: http://www.etsi.org/deliver/etsi_ts/102500_102599/102515/01.01.02_60/ts_102515v010102p.pdf.

[38] S. Moseley, S. Randall and A. Wiles. 'Experience within ETSI of the combined roles of conformance testing and interoperability testing'. In: *Standardization and Innovation in Information Technology, 2003. The 3rd Conference on*. Oct. 2003, pp. 177–189. DOI: 10.1109/SIIT.2003.1251206.

[39] Interoperability OASIS ebXML Implementation and Conformance Technical Committee. *ebXML Test Framework, Committee Specification Version 1.1 Draft*. Organization for the Advancement of Structured Information Standards, Oct. 2004.

[40] James Bret Michael et al. 'Verification and validation for trustworthy software systems'. In: *IEEE software* 28.6 (2011), pp. 86–92.

[41] Doron Drusinsky, James Bret Michael and Man-Tak Shing. 'A framework for computer-aided validation'. In: *Innovations in Systems and Software Engineering* 4.2 (2008), pp. 161–168.

[42] Pathanjali Sastri Akella and K Nageswara Rao. 'Effective Independent Quality Assessment Using IV&V'. In: *International Journal of Computer Science and Information Technology (IJCSIT)* 3.3 (2011), pp. 320–332.

[43] NATO. *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Volume I: Architecture Approach, Edition A, Version 1*. NATO Standardization Office (NSO), Feb. 2018.

[44] NATO. *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Volume II: Power Infrastructure, Edition A, Version 1*. NATO Standardization Office (NSO), Feb. 2018.

[45] NATO. *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Volume IV: Crew Terminal Software Architecture, Edition A, Version 1*. NATO Standardization Office (NSO), Feb. 2018.

[46] NATO. *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Volume VI: Safety, Edition A, Version 1*. NATO Standardization Office (NSO), Feb. 2018.

[47] NATO. *AEP-4754 NATO Generic Systems Architecture (NGVA) for Land Systems, Volume VII: Verification and Validation, Edition A, Version 1*. NATO Standardization Office (NSO), Feb. 2018.

[48] OMG. *Data Distribution Service (DDS), Version 1.2*. Object Management Group, 2007. URL: http://www.omg.org/spec/DDS/1.2.

[49] OMG. *The Real-time Publish-Subscribe Protocol (RTPS) DDS Interoperability Wire Protocol Specification, Version 2.2*. Object Management Group, 2014.

[50] NATO. *STANAG 4697, Platform Extended Video Standard (PLEVID), Edition 1, Ratification Draft*. NATO Standardization Office (NSO), 2013.

[51] ISO. *ISO 9000:2005, Quality management systems – Fundamentals and vocabulary.* International Organization for Standardization, 2005. URL: http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=42180.

[52] US Department of Defence. *MIL-STD-1512B, Technical Reviews and Audits for Systems, Equipments and Computer Software.* US Department of Defence, June 1995.

[53] N. Navet et al. 'Trends in Automotive Communication Systems'. In: *Proceedings of the IEEE* 93.6 (June 2005), pp. 1204–1223. ISSN: 0018-9219. DOI: 10.1109/JPROC.2005.849725.

[54] James D. Kindrick, John A. Sauter and Robert S. Matthews. 'Improving Conformance and Interoperability Testing'. In: *StandardView* 4.1 (Mar. 1996), pp. 61–68. ISSN: 1067-9936. DOI: 10.1145/230871.230883. URL: http://doi.acm.org/10.1145/230871.230883.

[55] ETSI Special Report. *Testing interoperability and conformity activities to be run during the implementation and promotion of the framework of digital signatures.* ETSI SR 003 186 V2.1.1 (2016-04). European Telecommunications Standards Institute, Apr. 2016. URL: http://www.etsi.org/deliver/etsi_sr/003100_003199/003186/01.01.01_60/sr_003186v010101p.pdf.

[56] Sungwon Kang. 'Relating interoperability testing with conformance testing'. In: *Global Telecommunications Conference, 1998. GLOBECOM 1998. The Bridge to Global Integration. IEEE.* Vol. 6. 1998, 3768–3773 vol.6. DOI: 10.1109/GLOCOM.1998.776013.

[57] Jianping Wu and Jianguo Wang. 'Internet protocol conformance testing by using a TTCN based protocol integrated test system'. In: *Communications, 1999. ICC'99. 1999 IEEE International Conference on.* Vol. 1. IEEE. 1999, pp. 646–650.

[58] Alex Vallejo et al. 'State of the art of IPv6 conformance and interoperability testing'. In: *IEEE Communications Magazine* 45.10 (2007), pp. 140–146.

[59] Steve Bishop et al. 'Rigorous specification and conformance testing techniques for network protocols, as applied to TCP, UDP, and sockets'. In: *ACM SIGCOMM Computer Communication Review.* Vol. 35. 4. ACM. 2005, pp. 265–276.

[60] Gerardo Pardo-Castellote. *OMG DDS Interoperability Demo March 2013.* OMG Technical Meeting, March 2013, Washington DC.

[61] Gerardo Pardo-Castellote. *OMG DDS Interoperability Demo June 2013*. OMG Technical Meeting, June 2013, Berlin.

[62] The Open Group. *Future Airborne Capability Environment (FACE): Conformance Certification Guide*. Reference X1606. The Open Group, Aug. 2016.

[63] Kyoungho An et al. 'Model-driven generative framework for automated OMG DDS performance testing in the cloud'. In: *ACM Sigplan Notices* 49.3 (2014), pp. 179–182.

[64] QinetiQ. *Model Maturity Levels, QinetiQ/MS/WD/GU1600085, Issue 2.0*. QinetiQ, Jan. 2016.

[65] Manas Pradhan and Daniel Ota. 'An Adaptable Multimodal Crew Assistance System for NATO Generic Vehicle Architecture'. In: *2016 International Conference on Military Communications and Information Systems (ICMCIS)*. May 2016, pp. 1–8. DOI: 10.1109/ICMCIS.2016.7496556.

[66] Manas Pradhan and Daniel Ota. 'Integrating Automotive Bus-based Networks in the NATO Generic Vehicle Architecture'. In: *ICCRTS 2016: 21st International Command and Control Research and Technology Symposium*. Sept. 2016.

[67] Daniel Ota, Periklis Charchalakis and Elias Stipidis. 'Towards a Verification and Validation Test Framework for Open System Architectures'. In: *2017 International Conference on Military Technologies (ICMT)*. May 2017, pp. 115–122. DOI: 10.1109/MILTECHS.2017.7988742.

[68] Daniel Ota. *Model Maturity Level 4 and 5 Testing Approaches*. NGVA Data Model Meeting, 22^{nd} Nov. 2016, London, Germany.

[69] Daniel Ota and Ditmir Hazizi. 'Interface Conformance Testing for Future Military Land Platforms'. In: *2017 International Conference on Military Communications and Information Systems (ICMCIS)*. May 2017, pp. 1–7. DOI: 10.1109/ICMCIS.2017.7956496.