An ISO/IEC 15118 Conformance Testing System Architecture

Kathleen Hänsch
Alexander Pelzer
Przemyslaw Komarnicki
Fraunhofer Institute for Factory Operation and Automation IFF
Magdeburg, Germany
{kathleen.haensch, alexander.pelzer, komarn}@iff.fraunhofer.de

Sven Gröning
Jens Schmutzler
Christian Wietfeld
TU Dortmund University
Communication Networks Institute (CNI)
Dortmund, Germany
{sven.groening, jens.schmutzler, christian.wietfeld}@tu-dortmund.de

Jörg Heuer
Reiner Müller
Siemens AG
Corporate Technology
Munich, Germany
{Joerg.Heuer, reiner.e.mueller}@siemens.com

***Abstract*— A major goal for the ISO/IEC15118 standard for controlling the charging process for electric vehicles is the simplicity and reliability of use. Severe threats for a simple use are implementations of the standard which are not interoperable to each other. A traditional approach to reduce this threat is the setup of testivals between all implementations to explicit test interoperability. However this approach is complex and expensive as it requires extensive test and the coordination between all implementers. The project eNterop takes a different approach to enable a broad acceptance of the ISO/IEC 15118 and IEC 61851 specification on the market and to facilitate a large number of implementations. It defines conformance tests which can be fully automated. This is a quite new approach for machine to machine interface specifications which not only cover communication but also power interfaces. With that implementers can independently test their protocols including the power flow control and increase the likelihood of interoperable implementations without expensive testivals. This paper presents the approach of conformance tests for combined communication and power interfaces.**

***Index Terms*-- Smart grid, Electric vehicles, Automatic testing, ISO IEC 15118, System testing**

I. INTRODUCTION

The German government intends to have one million electric vehicles (EV) on German roads by 2020. This will require broad market acceptance of electric vehicles. One important factor for the acceptance of electric cars is the frequency and ease with which drivers will be able to charge their electric vehicles anywhere. Thus standardization of the communication and power interfaces between electric vehicles and charging infrastructures is the fundamental prerequisite for broad market integration of electric vehicles. E.g. the European Commission has designated the type 2 plug as the standard for electric vehicles in Europe. What is more, European and American carmakers have agreed on the Combined Charging System (CCS) as the standard charging system for electric vehicles. ISO/IEC 15118 ([1],[2],[3]), which specifies the communications protocol between electric vehicles and charging infrastructures, is the standard for software interfaces between electric vehicles and charging infrastructures and the integration of electric vehicles in smart grids, in which, among other things, smart load management at low voltage level can be implemented [20]. Automatic payment procedures are also possible and will simplify and automate charging for electric vehicle users. Unfortunately, even though the standards for the interfaces are in place, the interoperability of the initial interface implementations is limited at present as the interpretation of the standards by the implementers often differs. To improve the situation the availability of conformance tests of the aforementioned standards is a fundamental prerequisite. The goal of this approach taken in the eNterop project is the interoperability of every electric vehicle with every charging station without extensive n-to-m interoperability test. For this a system that tests the conformance of electric vehicles and charging infrastructures with the standard ISO/IEC 15118 is being developed in the project.

II. TESTING SYSTEM ARCHITECTURE

The eNterop conformance testing setup is for black box testing of connected systems under test (SUT). The SUT is the system tested for conformance with ISO/IEC 15118 and can be either an EV or electric vehicle supply equipment (EVSE).

The eNterop conformance testing setup will test, whether the ISO/IEC 15118 protocol is correctly implemented and the IEC 61851-1 signaling and related power flow is handled correctly, which is a prerequisite for ISO/IEC 15118 communication.

In the first step, the complete eNterop conformance testing architecture was developed (see a detail of the SysML [4] block definition diagram in Fig. 1). The conformance testing system consists of the three main components of a test

This work was supported in part by the German Federal Ministry of Economics and Technology under Grant 01MX12060.

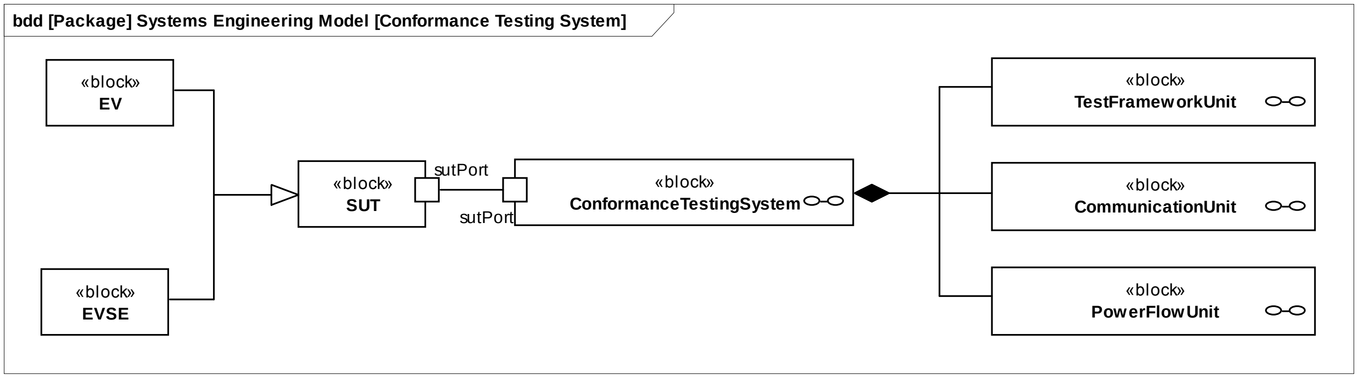

Figure 1. Conformance testing system architecture

framework unit, a communication unit and a power flow unit. The test framework unit is responsible for automatic execution of the conformance tests and report of the test results. It uses the communication unit to send and receive vehicle to grid (V2G) messages that are analyzed in the test framework.

The communication unit is an embedded system that transmits the messages generated by the test framework to the SUT and the responses from the SUT to the test framework. The communication unit also controls the power flow unit.

The power flow unit produces or emulates the electric connection like voltage and current between the SUT and the test system. This component has an interface used by the test hardware to control and measure the power flow (voltage and current) between the SUT and the test system.

The test framework unit and the communication unit consist also of several sub components. For example there is a test framework that executes the formally defined test cases and therefore it has an adapter to the communication unit in order to send and receive V2G messages. Additionally some codecs such as EXI (Efficient XML Interchange) codec are needed to decode or encode sent and received messages from the SUT.

To enable with this principle test setup the execution of conformance tests by independent implementations of this test setup, a language for modeling the test cases has to be selected.

III. Analysis, Evaluation and Selection of a Modeling Language

First, potential modeling languages were researched in order to be able to select a modeling language. The following well known modeling languages have been evaluated in the project:

- Message Sequence Chart (MSC)
- Unified Modeling Language (UML) / UML Testing Profile (UTP)
- Testing and Test Control Notation (TTCN-3)

MSC is a language for the specification of the behavior of distributed systems [5]. It is frequently used in telecommunications and the software industry to specify standards and to verify developed systems formally. It resembles a UML sequence diagram and models the sequence of message exchange [6], [7], [8], [9], [10]. MSC is frequently used in conjunction with other methods and languages such as TTCN-3 [5].

UML is the most widespread language for describing software systems' behavior and structure [11], [12], [14]. Since UML lacks modeling options relevant to testing, it was extended by UTP to provide options that specify testing systems and test sequences [13], [14].

TTCN-3 was developed to be a universally understandable, formal language for the description and specification of test behavior for black and gray box testing [15], [16]. TTCN-3 is a standardized, platform-independent testing technology [18].

The following requirements were imposed on the modeling of the test cases:

- Mapping between test cases and requirements has to be clear and use the requirement reference number.
- Test cases have to be describable both textually and graphically. Mapping between the graphical and the textual description has to be clear.
- The modeling language has to be standardized.
- The modeled test cases have to be convertible into other formats so that modeled test sequences can be used in as many test frameworks as possible.

A formal, textual representation is necessary because the project deliverables will be passed along to the ISO/IEC 15118 standardization bodies. A formal, textual description in the normative section of the standard will eliminate ambiguities in the interpretation of test case sequences. The graphical representation will make test case sequences easier to understand.

These requirements served as the basis for the specification of evaluation criteria for the modeling languages and for the evaluation of the modeling languages (see TABLE I).

TABLE I COMPARISON OF THE ANALYZED MODELING LANGUAGES

	MSC	TTCN-3	UML/UTP
Modeling Approach and Representation Format	• Textual or graphical modeling support. • Graphical modeling approach typically used (depending on tool support).	• Textual or graphical modeling support • Textual modeling approach typically used (depending on tool support)	• Textual or graphical modeling support. • Graphical modeling approach typically used (textual description for model exchange between different modeling tools).
Data Type Support for XML Schema	No XML Schema binding	Standardized XML Schema binding (TTCN-3 Part 9: Using XML schema with TTCN-3)	(non- standardized) XML Schema binding in most tools
Test Data Representation	No	Template mechanism to match messages	Template mechanism to match messages
Test Result Representation	No	Concept of verdicts and verdict resolution	Concept of verdicts and verdict resolution
Standardized Language	Yes (ITU-T Z.120)	Yes (ETSI ES 201 873 Series)	Yes (ISO/IEC 19505 Series), additional profiles as OMG specification available

All of the modeling languages have options for textual and graphical modeling and are standardized. TTCN-3 and UML/UTP have an option for explicitly modeling the results of test cases with the aid of verdicts. Test results are specified indirectly in MSC. Only a modeled test sequence is considered to be a successfully executed test. Any deviations from this are considered to be unsuccessful.

Parallel to this, the following modeling tools, which model in the identified languages and are already used by the project partners, were identified:

- TTWorkbench (TTCN-3)
- MSC Editor (MSC)
- EXAM Modeler (based on UML)
- Enterprise Architect (UML)
- Magic Draw (UML)
- Modelio (UML)

TTWorkbench and EXAM not only have options for modeling but also for test management and test case execution.

Evaluation criteria were also defined in order to evaluate the modeling tools. These criteria were assigned to individual evaluation categories and weighted. An evaluation scale was defined for every evaluation criterion and the semantics of valuation were specified.

The individual criteria were evaluated to select the modeling tool. To this end, Internet searches were performed and modeling tools were presented by their vendors. In addition, a model test case that analyzes the session setup of ISO/IEC 15118 was selected and modeled in the individual tools.

TTCN-3 meets every requirement. Since no limitations were identified, even when modeling the model test case for session setup, TTCN-3 and TTWorkbench were selected to model the test cases in this application.

IV. MODELING AND IMPLEMENTATION OF TEST CASES IN TTCN-3

A TTCN-3 test configuration had to be developed for the conformance testing system before test cases could be modeled in TTWorkbench and TTCN-3 (see Fig. 2). This test configuration based upon experiences previous researches in the field of interoperability testing for V2G CI [19].

The eNterop test configuration consists of the main test component (mtc) and two parallel test components (ptc) and the TSI. Two types, one for EV and one for EVSE, were defined for mtc. The type of SUT determines which one is used. One parallel test component additionally verifies that IEC 61851-1 functions are used correctly. Another parallel test component verifies that V2G CI messages are used correctly.

An abstract test system interface (TSI) is specified as a collection of ports. A TSI has no local timers, constants or variables. Only ports are assigned to it. During test case execution, test components ports can be mapped dynamically to the TSI ports to establish a communication channel to the real test system interface. The TSI in the eNterop test configuration uses one of two types depending on the type of SUT.

Four ports, a V2G port, two IEC 61851-1 port, one for each ptc, and one SDP port were defined. The V2G port sends and receives messages defined in ISO/IEC 15118 on application layer level. The IEC 61851-1 port sends messages to the communication unit. The communication unit interprets these messages and sets the IEC 61851-1 parameters at the interface to the SUT. This port also receives messages from the communication unit. The communication unit sends messages whenever any IEC 61851-1 parameters are changed at the interface to the SUT.

Once the ISO/IEC 15118 data types have been imported, templates can be created for the individual messages in TTCN-3 notation.

Then, test cases for the messages can be created, which verify which requirements from ISO/IEC 15118 are met by SUT.

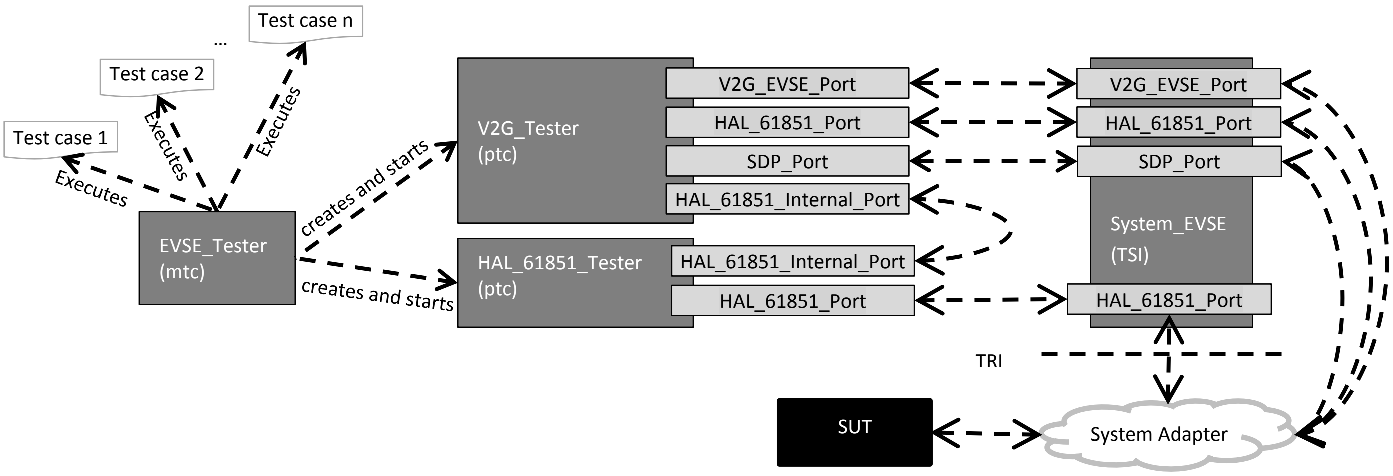

Figure 2. eNterop test configuration for EVSE tests

Once the test configuration has been defined in TTCN-3, the ISO/IEC 15118 data types have to be generated. TTWorkbench provides a plugin for generating TTCN-3 data types from XSD files. Definitions of all of the data types required to describe the corresponding V2G messages can be generated on the basis of the defined ISO/IEC DIS 15118-2 schema. Data types for other ISO/IEC 15118 protocol components (SDP, SLAC, IEC 61851-1) can be defined directly in TTCN-3

One test case executes a specific test behavior, thus verifying the behavior of the SUT based on transmitted messages. Every test case always begins and ends in a defined and recoverable state of the SUT and the tester in order to ensure the reproducibility of the test cases. A modeled test case consists of several elements [18]:

1. Preconditions and behavior (preamble)
2. Test behavior
3. Postconditions and behavior (postamble)

In the preconditions and behavior area all needed settings and variables have to be instantiated or imported. The test system shall start in a state so that a communication or data exchange with SUT is possible. In addition all needed ports should be connected with each other and the parallel test component will be created. This has to be done in each test case. Each test case should start in IEC 61851-1 state A. The SUT must be brought in the necessary IEC 61851-1 state that is required for the test case.

The testing system has to be started in a defined initial state so that communication with the SUT is possible. The precondition and behavior phase prepare the SUT for the actual test and bring it into the necessary state. To this end, all of the variables and settings must be imported and initialized the precondition and behavior phase. In addition, all of the ports are interconnected (see Fig. 2) and the parallel test components are generated and started in this phase. In the final step, the SUT must be put in the necessary state.

The test behavior describes the actual test in order to test, the behavior of the SUT. An example for the test behavior is shown in Fig. 3 in Graphical Presentation Format (GFT). The test behavior is defined for example using alt steps and starting and stopping timers. The result of the evaluation of a test behavior is defined by using verdicts. Verdict types are pass, fail, inconc or error. Pass means that the SUT performs the expected behavior triggered in the test case. If the SUT doesn't perform the expected behavior the test case fails. The verdict inconc is used when it isn't possible to decide whether the observed behavior is a pass or a fail, for example because the preconditions of the test case can't be achieved. Whenever a run-time error occurs the run-time system assigns an error verdict, for example division by zero.

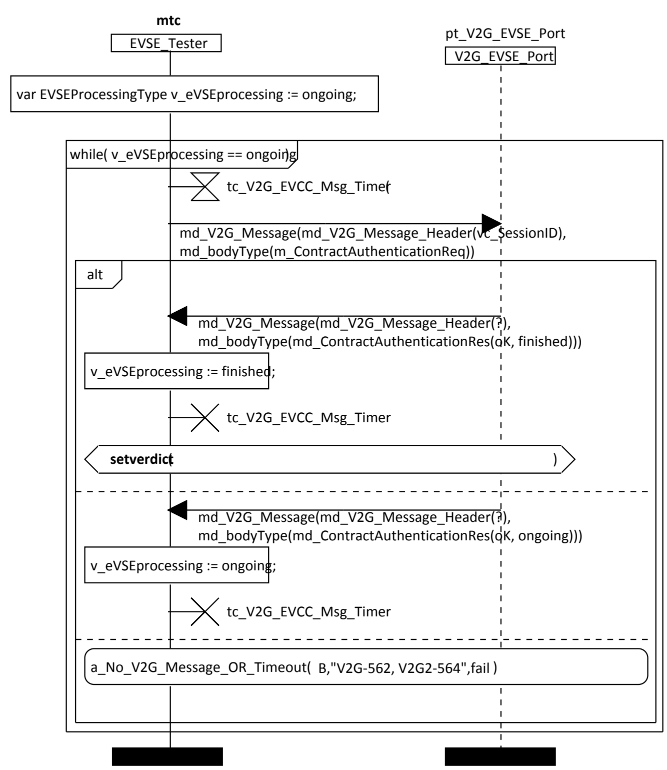

Figure 3. Test behaviour in GFT

The different behavior of the SUT is verified in altsteps. Each altstep includes a possible behavior of the SUT and described what should happen next. For example each testcase contains usually an altstep that checks whether this timer has expired and if this is the case it sets the verdict to fail. When waiting for a reaction from the SUT, one typically describes which messages. Another possibility is that one altsteps contains unexpected messages. If such message is sent by the SUT the test case is also failed.

All of the steps necessary to return the SUT and the testing system to the defined final state are executed in the post-condition and behavior phase. Afterward, both systems are ready to execute a new test case.

Since IEC 61851-1 is a prerequisite for communication in compliance with ISO/IEC 15118, appropriate functions have to be integrated for the IEC 61851-1. During test case execution, all necessary IEC 61851-1 instructions are initiated by the mtc.

At present, the individual test cases are being modeled and alternatives for modeling a particular test behavior are being identified and discussed. It is important that an alternative for modeling certain test behaviors is selected, which assures that the behaviors are modeled identically in every test case and modeling is consistent. The selected modeling approaches will be documented in modeling guidelines.

V. Modeling and Implementation of Test Cases in TTCN-3

This paper presents an approach to a V2G CI testing system architecture with which the conformance of EV and EVSE with ISO/IEC 15118 can be tested. First, the complete test bench architecture was described and the individual components were explained.

Then, a modeling language and a modeling tool with which the test cases can be modeled were selected. First, potential modeling languages and tools were investigated and evaluated using weighted criteria. Based on this evaluation, TTCN-3 was selected as the modeling language and a test framework was defined and set up.

In the next step, the test configuration was specified in TTCN-3. Necessary components and ports were also specified.

The requisite data types were imported directly from ISO/IEC 15118-2.

Then, the general configuration of the test cases was presented. Since the preconditions from IEC 61851-1 Annex A also have to be verified during test case execution, a parallel test component that does this was defined in the approach presented here.

In the future, the individual test cases will have to be modeled and specified as test sequences in the form of test suites and campaigns and the interface to the communication unit and the SUT will have to be finalized. The results of this work will be contributed to the joint working group between ISO and IEC which is responsible for the conformance test specification in ISO/IEC15118-4.

References

[1] *Road vehicles -- Vehicle to grid communication interface -- Part 1: General information and use-case definition*, ISO Standard 15118-1:2013, April 2013.

[2] *Road vehicles -- Vehicle to grid communication interface -- Part 2: Network and application protocol requirements*, ISO/FDIS Standard 15118-2, August 2013.

[3] *Road vehicles -- Vehicle to grid communication interface -- Part 3: Physical and data link layer requirements, ISO/FDIS Standard 15118-3*, August 2013.

[4] *OMG Systems Modeling Language (OMG SysML), Version 1.3*, OMG specification, June 2012.

[5] *Message Sequence Chart (MSC)*, ITU-T Standard Z.120, April 2004.

[6] S. Mauw, M.A. Reniers and T.A.C. Willemse, "Message Sequence Charts in the software engineering process," in *Handbook of Software Engineering and Knowledge Engineering*, 1st ed., vol. 1, S.K. Chang, Ed., Singapore: World Scientific Publishing, 2002, pp. 437-464.

[7] Ø. Haugen, "MSC-2000 interaction diagrams for the new millennium," *Computer Networks*, vol. 35, pp. 721-732, 2001.

[8] R. Alur, G.J. Holzmann, and D. Peled, "An analyzer for message sequence charts," *Software Concepts and Tools*, vol. 17, pp. 70-77, 1996.

[9] A. Muscholl, D. Peled, and Z. Su, "Deciding properties for message sequence charts" in *Proceedings of the 1st Conference on Foundations of Software Science and Computation Structures*, Lisbon, Portugal, pp. 226-242, 1998.

[10] H. Dan and R.M. Hierons, "Conformance Testing from Message Sequence Charts" in *Proc. 2011 IEEE International Conference on Software Testing, Verification and Validation*, pp. 279-288.

[11] *OMG Unified Modeling Language (OMG UML) Infrastructure, Version 2.4.1*, OMG specification, August 2011.

[12] *OMG Unified Modeling Language (OMG UML) Superstructure, Version 2.4.1*, OMG specification, August 2011.

[13] *OMG UML Testing Profile (UTP)*, Version 1.2, OMG specification, April 2013.

[14] A. Bagnato, A. Sadovykh, E. Brosse and T.E.J. Vos, "The OMG UML Testing Profile in use - An industrial case study for the Future Internet testing" in *Proc. 2013 17th European Conference on Software Maintenance and Reengineering*, pp. 457-460.

[15] *Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language*, ETSI Standard ES 201 873-1, April 2013.

[16] I. Schieferdecker and A. Vouffo-Feudjio, "The Testing and Test Control Notation TTCN-3 and its use," in *Formal Methods for Industrial Critical Systems: A Survey of Applications*, 1st ed., S. Gnesi, Ed., T. Margaria, Ed., New Jersey: John Wiley & Sons, Inc., 2013, pp. 207-235.

[17] Muscholl, D. Peled, and Z. Su, "Using TTCN-3 for testing the interoperability of HL7v3 based applications" in *Proceedings of the 1st Conference on Foundations of Software Science and Computation Structures*, Lisbon, Portugal, pp. 226-242, 1998.

[18] AC.Willcock, T. Deiß, S. Tobis, S. Keil, F. Engler, and S. Schulz, *An Introduction to TTCN-3*, vol. 2. New York: Wiley, 2011.

[19] S. Gröning, C. Lewandowski, J. Schmutzler, and C. Wietfeld, "Interoperability Testing based on TTCN-3 for V2G Communication Interfaces" in *International Conference on Connected Vehicles and Expo (ICCVE)*, Beijing, China, pp. 298-303, 2012.

[20] J. Schmutzler, C. Wietfeld, S. Jundel, and S. Voit, "A Mutual Charge Schedule Information Model for the Vehicle-to-Grid Communication Interface" in *7th IEEE Vehicle Power and Propulsion Conference (VPPC)*, Chicago, USA, pp. 1-6, 2011.