Highlighting the Essentials of the Behaviour of Reactive Systems in Test Descriptions Using the Behavioural Atomic Element

Lars Ebrecht
German Aerospace Center (DLR)
Institute of Transportation Systems
Braunschweig, Germany
lars.ebrecht@dlr.de

Karsten Lemmer
German Aerospace Center (DLR)
Institute of Transportation Systems
Braunschweig, Germany
karsten.lemmer@dlr.de

Abstract **- The work described in this paper depicts an approach on how to close the lack of structure for test specifications, test descriptions and test data representations. On the example of a test data representation with the eXtensible Markup Language (XML), it will be shown how to form the structure using the behavioral atomic element and two testing patterns, i.e., for test stimuli and test reactions. The patterns are described using Petri net semantics. The scalability and flexibility of the approach, i.e., enabling the consideration of domain specific information, will also be shown on the example of the testing format. The pros and cons for using non-black box information in test descriptions and reference black box behavior specification for reactive systems, like actions, tasks and processes as well as system states in addition to the interface events, will be discussed in this paper at the end. The main achievement is a scalable, platform- and implementation-independent Test description using the meta-model described by the patterns.**

Keywords - Reactive systems; Validation and Verification, Real-time behaviour Pattern; Test Pattern; Validation Pattern; Test specification

I. INTRODUCTION

Black box test specifications mostly consist of stimuli and reactions mixed and alternated in any order according to the functional requirements to be covered. In some cases test steps are sporadically inserted in order to indicate actions, processes or changes inside the system under test (SUT), e.g., here a safety-critical distributed reactive system - a train control unit. Concerning the test creation and evaluation the practice shows that it is very useful to add this information. These non-black test steps enable a better understanding of certain situations as well as figuring out specific system features and functionalities. Consequently, no order or any order concerning the use and mixture of stimuli, reactions, internal processes and system states of a reactive system is fixed by the application of the behavioral atomic element as generic pattern for the behavior of reactive systems and black box tests. Further it will be shown that the behavioral atomic element does not restrict the test description too much. It creates moreover a scalable and concerning the consideration of domain specific information, flexible as well as consistent and transparent the behavior description for reactive systems. The introduction and demonstration of the approach is done on the example of a test description for a train control onboard unit.

A. *Context of the work and application example*

The test description that is used as application example in this paper, addresses the European Train Control System (ETCS). ETCS comprises two main safety-critical distributed real-time components, i.e., the ETCS onboard unit (OBU) located on each train and the Radio Block Center (RBC) on the trackside. The RBC mainly supervises the location of the trains for a certain area of a track in order to provide movement authorities to the trains in that area. The OBU uses the information got from the RBC mainly to supervise the train movement, i.e., the maximum allowed speed and distance allowed to travel. All the functional requirements telling how and when to do what are specified in the European wide standardized System Requirement Specification (SRS) (Subset-026 [5]). In order to check if an OBU fits the SRS, i.e., the behavior, all the interaction and way of exchanging information via different physical interfaces with the track, the functional requirements have been analyzed and assigned to features and test cases. The test cases have been selected, parameterized and combined to form virtual train trips, called test sequences, in order to emulate the start of mission, train movement with certain specific conditions as well as the end of the virtual trip. Thus, the technical conformity and interoperability of the OBU will be checked against SRS (Subset-076 [6]).

The conformity and interoperability tests of Subset-076 are used in independent laboratories like the Railway Simulation and Testing laboratory (RailSiTe® [14]). Only giving some rough figures to the modular and distributed test environment, the RailSiTe contains different modules for the simulation of train dynamics and track properties as well as different physical black box hardware interfaces, i.e., high frequency signals, digital I/O, TTL, GSM-modem-serial-connection, camera and robot for a touch screen display acting as man machine interface of the OBU [1].

Concerning the tests applied in this test environment there is a separation of the logical and functional behavior on the one hand and the physical interface behavior, i.e., the signal generation and emulation, on the other hand. The test

 ISBN: 978-1-61208-111-3

and reference description represent the logical and functional behavior that are used by different interface modules to encode the information and to build the physical signal in order to stimulate the SUT, i.e., the OBU. The emulated train dynamics, i.e., the acceleration and braking, are logically and mathematically described in the test and reference behavior description.

B. *Structure of the paper*

This paper is structured as follows. Section II depicts related work and the differentiation of the approach described in this paper to other present approaches for the test specification and behavioral description for reactive systems. The following Section III introduces the atomic element for the behavior of reactive systems and the two testing patterns, i.e., for test stimuli and test reactions that are described using Petri net semantics [13]. Section IV represents the main section of the paper introducing the structure of the test description and test data representation with the eXtensible Markup Language (XML) [8] followed by the demonstration of scalability and flexibility concerning the consideration of domain specific information. The last Section V will discuss and conclude the presented approach and results.

II. Related work

There exist various ways and standards for modeling and describing the behavior as well as tests for reactive systems. Beside the continuous signals in the electrical engineering or control theory domain, there are several standards coming from the software engineering domain, e.g., the Unified Modeling Language (UML), the System Modeling Language (SysML), the UML Testing Profile (UTP), the Testing and Test Control Notation (TTCN-3). The approach presented in the following does not touch continuous signals in the way they are described by Matlab / Simulink, LabView or similar tools or approaches.

The presented approach is more related to the formal mathematical and software engineering domain. In this domain the main standards are listed above. The UML [3] and SysML [2] specify how to describe systems and processes within a user-friendly representation in form of different diagrams with certain associated graphical symbols and elements. Both do not include a clear and formal concept for the behavioral description of reactive systems. More or less activities, operations, message exchanges and states are spread over the Activity, the Sequence and the State diagram without any concrete order and relationship specific for the reactive system behavior.

The UML Testing Profile (UTP) enables the description of black box tests [11]. You can describe test contexts, test configurations and test components. For the behavioral part of a test description the UTP offer test cases, test case parameters, stimuli, observations and test data as well as test routines and test procedures [11]. However, there is no comparable approach of a Meta model like it is defined by the behavioral atomic element. There are pre- and end-conditions of operations and invariant system properties without any concrete mapping to figure out the relation to the behavior of a reactive system as well as having an atomic element.

The Testing and Test Control Notation (TTCN-3) of the European Telecommunications Standards Institute (ETSI) is an additional standard for the black box test description. It mainly focuses on the implementation, execution and evaluation of black box tests [4][12]. The semantics of flow graphs, especially the flow graph frame that consists of start node, basic node and an end node [4], are not consequently used as generic basic concept for the behavioral description of reactive systems like it will be shown in the following with the atomic element.

III. The behavioural atomic element and related test pattern

In the following, the behavior of a reactive system is understood and described by the behavioral atomic element [10]. The element comprises the system configuration, i.e., system states (S_i), in- and out-going events (e.g., periodical, synchronous and asynchronous events) (E_j), sequential and parallel activities, operations or processes (A_k, AS_k).

In the middle of Figure 1. the pattern of the atomic element is depicted using the Petri net semantics [13]. The Petri net representation indicates the activity A_k as transition encapsulated by at least one in-coming event E_i, potential out-going events E_o as well as the starting state S_s and end state S_e of the functionality. The events and states are shown as places. The “b” represent a condition that enable the specification of further trigger conditions, e.g., temporal conditions, for the activation of the activity A_k. This is called the compact representation of the atomic element.

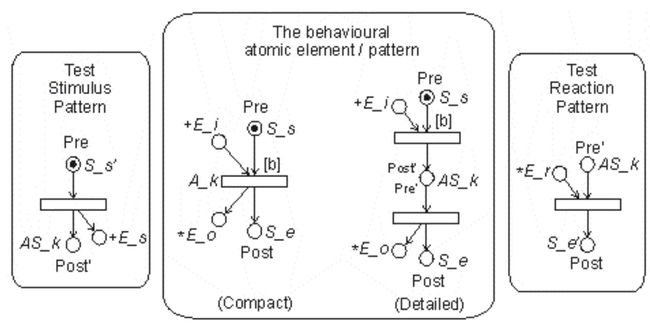

Figure 1. Basic Patterns describing the atomic behaviour of reactive systems, test stimuli and test reactions

Figure 1. shows a second detailed and unfolded representation of the atomic element that depicts the activity AS_k as place. In comparison to the compact representation the unfolded one highlights the activity. So, the activity is indicated as a meta-state, i.e., the system configuration might change or might be in-between two stable system states also having the possibility of further investigations concerning the description of in-coming and out-going events. Thus, the unfolded representation of the atomic element emphasizes the system activity enormously. The whole configuration enforces the understanding of the system behavior by

relating events together with states and the activity to a specific functionality.

The consideration of system states, events and activities improve the transparency and understandability of the system behavior. Even if this seems to be like a grey box test, this is not the case due to the fact testing the SUT via the black box interfaces and without referring to internal interfaces or structures like sub-classes or modules. On the very left side of Figure 1. the pattern describing the events stimulating the SUT (E_s) is shown in comparison to the pattern of managing reactions (E_r) coming from the SUT on the very right side of Figure 1.

Time is considered as a global discrete variable in addition to system states, event occurrences and process durations. Of course concerning non discrete real-time you will have to consider the WKS-Sampling-Theorem as well as accuracy aspects like jitter with proper tolerance borders and intervals.

A. Test scenario defined by the atomic element and the testing and validation patterns

For a better understanding one very easy scenario demonstrates the use and interaction of the three patterns mentioned before concerning a common situation, i.e., testing a SUT with a certain test environment (TE) (see Figure 2.). In the middle there is a UML [3] sequence diagram showing the TE and the SUT. From the TE the stimulation E_s is triggered stimulating the SUT in the sense of a certain in-coming event E_i. After the reception of this event the operation A is started maybe emitting the result or confirmation or response with an outgoing event(s) E_o that is taken into account as reaction E_r from the SUT. In the background the Petri net representation is shown separated in TE side on the left and the SUT side on the right.

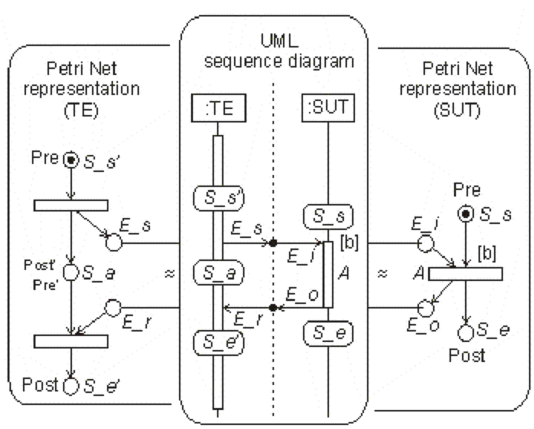

Figure 2. Test scenario: Test environment (TE) stimulating the system under test (SUT) in order to check function A

IV. APPLICATION OF THE ATOMIC ELEMENT TO THE TEST DESCRIPTION

In the previous section, the behavioral atomic element has been described that define the functional behavior of a reactive system as well as the other two test and validation patterns for the stimulation of a SUT as well as for receiving reactions from a SUT, here a train control unit (OBU). Now, it will be demonstrated how these patterns are used to form a test description and reference behavior on the example of a test data representation by an XML-schema [7]. The XML-schema defines the structure of an XML [8] file assigned acting as instance of that schema.

In the first subsection, the overall structure and content will be roughly depicted. The second subsection shows the application of the behavioral atomic element in detail. The last two subsections explain how the patterns enable a scalable and customizable use of the test description.

A. Overview and structure of the test description format

The highest level of the overall structure includes four elements (General, StartingConditions, StepList and EndConditions). The element General contains the ID of the test sequence, a title, description, a substructure for the test sequence release and modification history table beside a link to a set of features and test cases. Out of the set of test cases some will be selected, instantiated and concatenated with others in a test sequence. Accordingly specific starting conditions will be fixed and stored in the StartingConditions element. As shown in Figure 3. this item comprise two generic and two domain specific sub-elements. The Sub-element Variable and Set allow defining variables and sets of variables with concrete values, intervals or enumerations in order to describe the system state and system variables. M_MODE and Location represent customized entries for the mode of an ETCS onboard unit (OBU), e.g., no power, standby, full supervision, shunting, etc., and the other comprises several variables defining the train location, i.e., the orientation of the train and travelled distance in relation to certain waypoint in the track. There exist further application domain specific entries in the starting that are not mentioned here.

The EndConditions element is similar to the StartingConditions element and describes accordingly the system state reached at the end of a test sequence. Between the starting and end conditions there is the StepList element located. This element takes the main role in the overall structure. As displayed in the Figure 3. it contains a list of steps that can be one of three following kinds, i.e., a state, an action event or an action. Besides, each step has a unique step ID and can be grouped, e.g., in the case of test case associations or sequential or parallel ordered steps. In Subsection IV.D it will be explained more in detail how the group element can be used to describe conditions and relationships over different steps for domain specific aspects and dependencies. Last but not least you can define variables in the SignatureVariableList that interconnect steps in order to exchange data among different steps.

The three kinds of the step as main element mentioned before are used to describe the behavior as defined by the atomic element. For each step it has to be chosen if the step describe a system state or an action event, i.e., a stimulus to be triggered by a test environment or a reaction coming from the SUT, or an action that is assumed to be active in the SUT. In the following section it will be explained how these

 ISBN: 978-1-61208-111-3

three different elements will be connected in order to satisfy the behavioral atomic element.

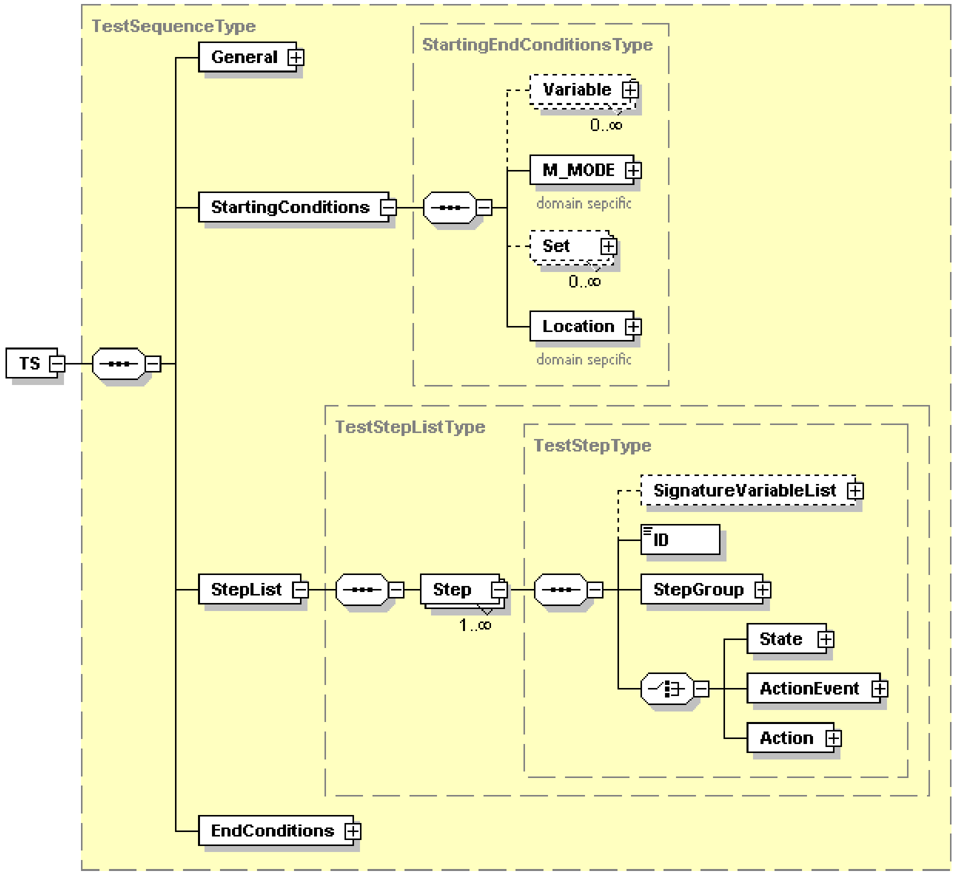

Figure 3. Overal structure of the test data representation defined by an XML-Schema [7]

B. *Details of the test data description according to the behavioural atomic element*

The beginning of a function should be a system state. As depicted in Figure 4. a state is defined by several variables contained in the VariableList substructure or by sets of variables in the Set element similar to the starting and end conditions described before. Furthermore a state has an identifier (ID), a name and a textual annotation.

A State is followed by an action event, i.e., one or more in-coming events stimulating the SUT. Action events are divided in in-coming, i.e. stimuli (E_i) and out-going events, i.e., reactions (E_o). All action events are described by an ID, a name, the event direction and interface, a description and related data (see Figure 4.). The Data sub-element comprises two alternatives, a reference to another action event and a set of elements composed of the action event trigger, duration, delay, function name and a function parameter list. The first alternative is used for the case if a step specifies for instance an optional packet that should be included in an event together with other packets, which are specified by other steps. In this case it is possible to include this packet together with the other ones in one action event whereby the data are only included in one step and the other refer to this action event. The second alternative represents the usual case where an event is triggered by, e.g., the location of the train, the time elapsed from the beginning of the simulation, an other action event or a set of variables defining a certain system state. The Duration and the Delay element contain a variable type describing the time for the generation and transmission and possible delays related to the trigger point. The function-name is used to determine runtime commands or routines of the test environment using all the variables and its values contained in the element ParameterList.

The last of the three main elements, the Action, depicts an action or task of the SUT. It is similar to the ActionEvent element. It has an ID, name and description and contains a substructure for the related data. Except from the ActionTrigger element the Data element is equal to the one of the action event. In comparison to the action event the action is triggered by one or more in-coming events that have to be received before.

Due to the fact that you want to violate the specified rules and conditions of a behavior in some cases testing a SUT, it is not forbidden to mix the three main elements also against the behavioral atomic element what represent an abnormal situation. You can ever check and identify these violations applying checks to the test and reference behavior descrip-

 ISBN: 978-1-61208-111-3

tion finding deviations in relation to the behavioral atomic element in order to check if you wanted to do so or not.

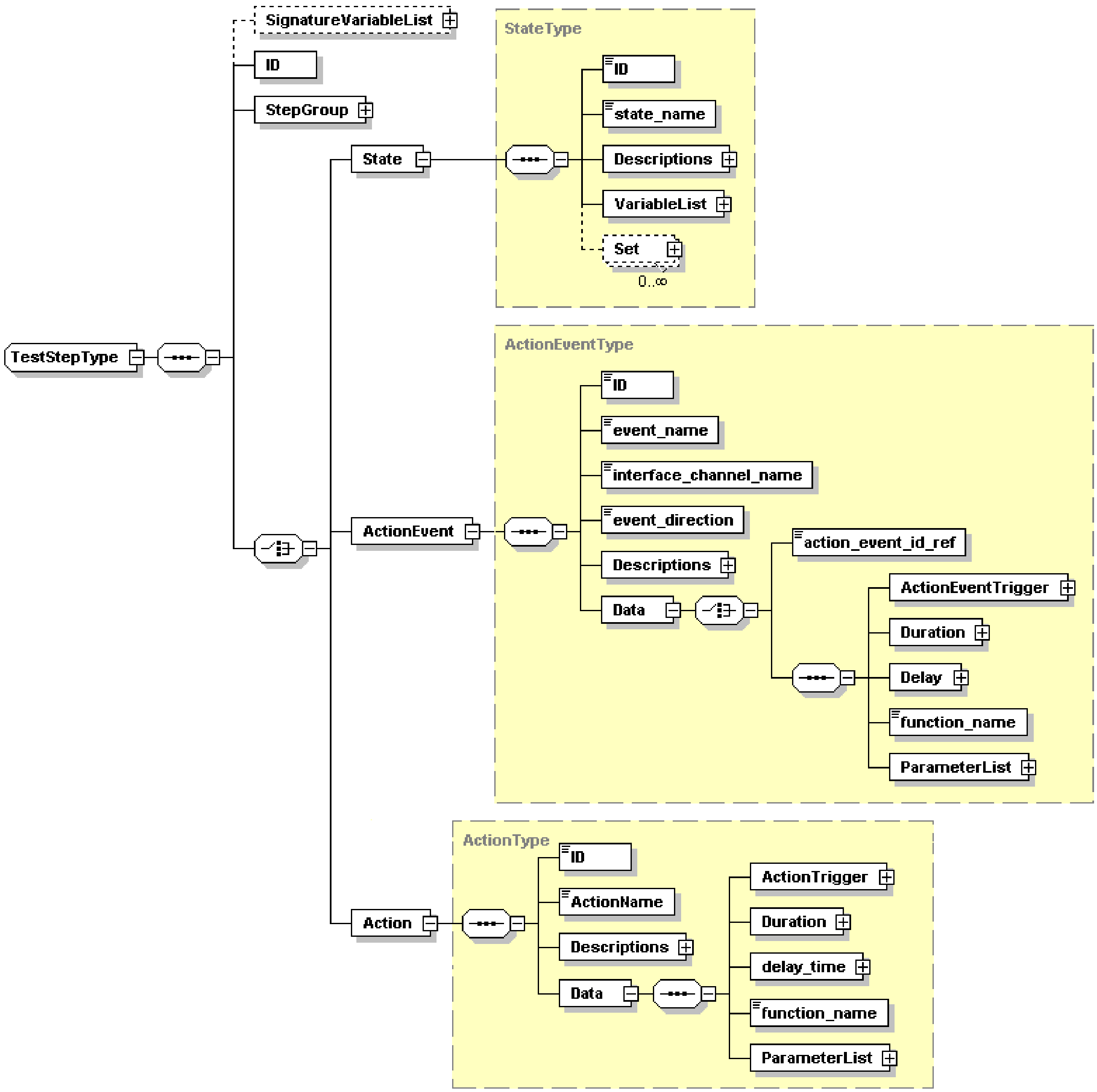

Figure 4. The atomic element builds the generic structure for a test and reference behaviour description

C. Scalability of the behaviuor description

The application of the atomic element enables on the higher abstract level as well as on the lower detailed level the possibility to combine test sequences with matching starting and end conditions [9], this not very astonishing, but also for concrete functionalities according to matching of post and pre states (compare with Figure 1.). So, small functional units can be consistently combined, i.e., an action belonging to a certain functionality including its trigger event(s) as well as the starting and end state. Test cases only containing starting and end conditions and black box message passing in between having a lack of building sup-units for smaller function inside bigger ones like the atomic element cramping events and an action by system states. Thus, there is the same consistent approach for test sequences over test cases until the steps included in a test case.

D. Differentiation of the atomic element considering domain and user specific relations and dependencies

The most popular group for test steps is the association to a test case. Black box stimuli and reactions are related to test cases and these for itself to features in order to manage the system complexity in principle. Beside the step order like sequential, parallel or strict execution or occurrence defined in the Control element (refer to Figure 5.), there is the need to specify domain and user specific relationships and dependencies. In the context of testing an OBU there is the need to describe a logical order for a certain type of events, i.e., Balise messages. One Balise message contains one up to eight Balise telegrams emitted by a passive discrete device that is in principle comparable with RFID tags only that additionally Balises include some more safety mechanisms in order to avoid copying telegrams. A Balise message has to be consistent, that means that the telegrams have to receive in the right and continuous order and completely, i.e., all of the telegrams. This association is described by the Special-Groups element. This element comprises an ID, name, group order, parent group for hierarchical dependencies and the group instance, in case that there more than one group occurrences.

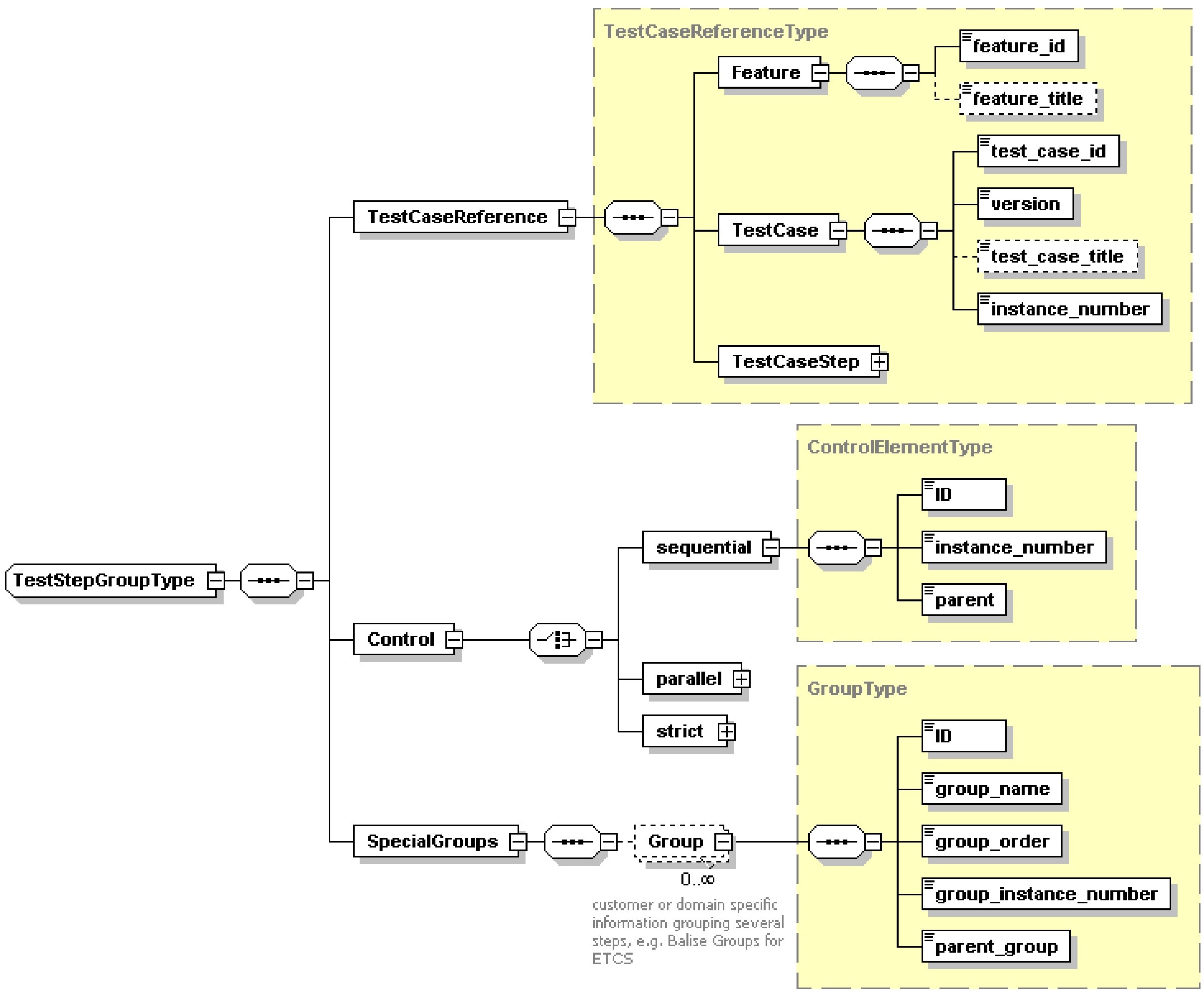

Figure 5. Vertical differentiation of the test steps

V. CONCLUSION

The contribution introduced the behavioral atomic element described as Petri net acting as pattern for a Meta model and forming a test data representation and reference behavior description. Beside, two other testing and validation patterns complete the scenario testing a reactive system with a certain test environment on a logical abstract level. Mostly, test descriptions and specification consider more or less mainly test stimuli and reactions of a system under test. The presented approach does not only depict the application of the atomic element to a test data representation, it also figures out how this reference behavior description can cope with domain specific requirements and dependencies as well as enabling a scalable, consistent and transparent reference behavior description highlighting the essentials of the behavior of a reactive system, i.e., system states, in-coming events or trigger events, actions or operations and out-going responses. Thus, the consistency and transparency of the test specification for any reactive system is improved by the definition of proper behavioral units and functions with the help of the behavioral atomic element. The presented XML format defined in the Schema is currently used to automate the test execution and evaluation in the rail simulation and testing laboratory RailSiTe®.

ACKNOWLEDGEMENT

Many thanks to the group rail technology, especially to C. Preuk and C. Torens for the feedback and review comments.

REFERENCES

[1] C. Torens and L. Ebrecht, Remote Test: A Framework for Testing Distributed Systems, Proc. International Conference on Software Engineering Advances (ICSEA2010), Aug. 2010.

[2] Object Management Group, „OMG Systems Modelling Language“, Ver. 1.2, June 2010, www.omgsysml.org/#Specification (23.08.10).

[3] Object Management Group, „Unified Modelling Language“, Superstructure Specification, Ver. 2.3, May 2010, www.omg.org-/spec/UML/2.3/ (23.08.10).

[4] ETSI, ES 201 873-1, „Testing and Test Control Notation - Part 1: TTCN-3 Core Language,“ Ver. 4.2.1, May 2010, webapp.etsi.org-/workprogram/Report_WorkItem.asp?WKI_ID=29256 (23.08.10).

[5] ERA, „ERTMS/ETCS – Class 1, System Requirement Specification Subset-026 (SRS)“, Version 2.3.0d, 21 Apr. 2009.

[6] ERA, ERTMS/ETCS – Class 1, „Technical Specification for Interoperability: Subset-076-6-3 (Test Sequences)“, Version 2.3.1, 19 Oct. 2009.

[7] XML Schema, W3C, http://www.w3.org/XML/Schema (23.08.10).

[8] T. Bray et al., W3C, XML, www.w3.org/TR/2008/REC-xml-20081126, 26 November 2008 (23.08.10).

[9] L. Ebrecht, M. Meyer zu Hoerste and K. Lemmer, “The Basic Concept for the Formal Test Description - Horizontal Composition and Vertical Differentiation of the Atomic Element,” Proc. Formal Methods for Automation and Safety in Railway and Automotive Systems (Forms/Format), GZVB, Jan. 2007, pp. 447–457. ISBN 13: 978-3-937655-09-3.

[10] L. Ebrecht and K. Lemmer, “Konsistente Verknuepfung von Aktivitaets-, Sequenz- und Zustandsdiagrammen - Darstellungsunabhaengige und formale Semantik zur Verhaltensbeschreibung von Echtzeit-Systemen,” Proc. Mobilitaet und Echtzeit - PEARL. Informatik Aktuell. Springer, Dec. 2007, pp. 49–58. ISBN 978-3-540-74836-6.

[11] Object Management Group. (July 2005). UML Testing Profile, Ver. 1.0-05-07-07, www.omg.org/technology/documents/formal/test_profile.htm (23.08.10).

[12] J. Grabowski. TTCN-3 - A new Test Specification Language for Black-Box Testing of Distributed Systems. In: Pro-ceedings of the 17th International Conference and Exposition on Testing Computer Software (TCS 2000), Washington D.C., www.swe.informatik.uni-goettingen.de/publications/JG/Grabowski.pdf, Jun 2000 (23.08.10).

[13] Billington, Jonathan ; Reisig, Wolfgang: *Application and Theory of Petri Nets*, Springer, 1996, ISBN 3-540-61363-3

[14] RailSiTe DLR homepage, www.dlr.de/fs/en/desktopdefault.aspx/-tabid-1235/1688_read-3254/ (23.08.10)

