An Overall View of the SPP IF Project: 'The Automatic Generation of Test Purposes'

Jens Grabowski*, Dieter Hogrefe*, and Daniel Toggweiler*

Abstract. The project *'The Automatic Generation of Test Purposes'* is part of the second series of the 'Priority Programme Informatics' (SPP IF), Module 1: 'Safe Distributed Systems'. It intends to improve the testing process of communication systems. The work is funded partially by the Siemens-Albis AG in Zürich. The project started on 1. October 1994 and will be carried out at the University of Berne. This project presentation describes the problems which shall be tackled and explains the planed line of actions in order to solve the mentioned problems.

1. THE PROBLEM AREA

The aim of testing is to protect users and customers against insecure, inappropriate, or even erroneous software products. In the telecommunication area special tests, so-called *conformance tests*, are demanded by the costumers. A conformance test should ensure the required functions of a component to communicate with other system components. The specification of these functions, in the following called *protocol specification*, can be found in standards or recommendations provided by international standardization organizations (e.g. ITU-TS, ISO/IEC, or ETSI).

A conformance test is a complex and error prone process during which several tasks have to be carried out and various documents have to be produced. In order to make test results comparable the entire conformance testing procedure is standardized by the international ISO/IEC standard 9646 *'Conformance Testing Methodology and Framework'* (CTMF) [1].

1.1 The CTMF Testing Phases

CTMF structures the conformance testing procedure in *test specification phase*, *test execution phase*, and *test result analysis phase* (Figure 1).

The test specification phase comprises all actions necessary to specify a set of test cases, a so-called *test suite*. This phase is based on a specification of the protocol which should be tested.

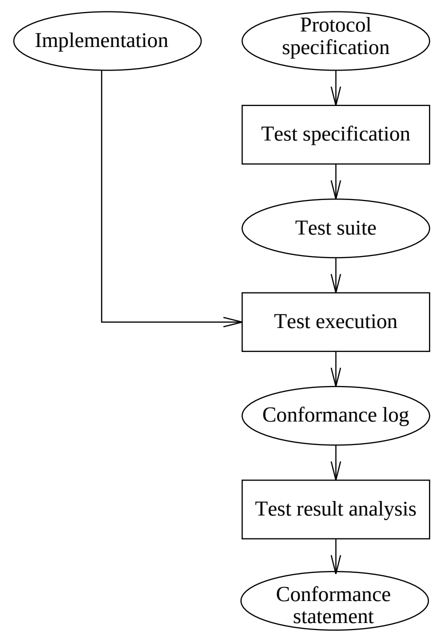

Figure 1: Conformance Testing Phases

In the test execution phase the test cases are applied to the implementation. The test runs are recorded in a *conformance log*.

During the test result analysis phase the conformance test log is analyzed and a *conformance statement* is produced. The conformance statement describes how good the implementation conforms to the specification.

CTMF describes the individual steps within the conformance testing procedure with a different degree of detail and formality. As a consequence, they also differ in the possibility to be automated.

Our project intends to improve the test specification phase. We describe this phase in more detail.

*University of Berne, Länggassstrasse 51, 3012 Bern, Switzerland, [grabowsk,hogrefe,toggweil]@iam.unibe.ch

1.2 The Test Specification Phase

During the test specification phase two main tasks have to be performed (Figure 2). In a first step a set of *test purposes* has to be deduced from the specification. CTMF defines a test purpose as an informal statement describing a behavior or a property which shall be proven by a test. The second step comprises the development of the test cases. Based on the specification for each test purpose at least one test case has to be developed.

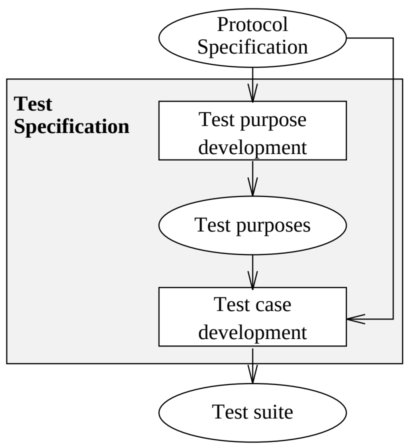

Figure 2: Test specification phase

Due to the missing formalization of the term *test purpose* in the test specification phase both mentioned tasks are described only informally. Protocol specification and test cases, i.e. the other input and output data of the two tasks (cf. Figure 2), can be assumed to be represented formally. Standardized formal description techniques (FDTs), i.e. SDL, Estelle, and LOTOS [2], can be used to specify protocols formally and the standardized test notation TTCN [3] can be used to represent test cases.

Within the project *'Conformance Testing - A Tool for the Generation of Test Cases'*, funded by Swiss PTT under contract no. 233/257, we developed the SaMsTaG[1] method which formalizes the term test purpose and allows an automatic generation of test cases [4].

The aim of the SPP IF project *'The Automatic Generation of Test Purposes'* is to automate the remaining task, i.e. the test purpose development in Figure 2, of the test specification phase.

[1]SaMsTaG is an abbreviation for *Sdl And Msc bAsed Test cAse Generation.*

2. FORMALIZING THE PROBLEM

In order to automate the test purpose production we have to formalize the problem. At first we need some formal representation for the protocol specification and the test purposes. Secondly, from literature (e.g. [5]) we know that a conformance relation and additional assumptions and restrictions also influence the test purpose generation. A refined view of the test purpose generation problem is depicted in Figure 3.

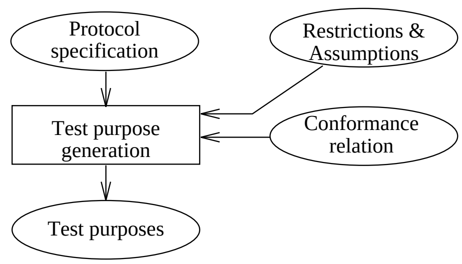

Figure 3: Test purpose generation

2.1 Protocol specification

Generally, a telecommunication protocol can be looked at as a reactive system, i.e. a system which is in some state and which reacts on stimuli from its environment by responses and by changing to a new state. Such a reactive system can be specified by means of the standardized FDTs SDL, Estelle and LOTOS. We abstract from the used FDT by assuming that a reactive system is given as a labeled transition system (LTS), i.e. as an infinite automaton. The concrete LTS can be gained partially by simulating the FDT specification.

2.2 Test Purposes

The application of a test case proves the existence of a property by forcing the implementation to behave in a prescribed way. The property which should be tested is called *test purpose*. A test case should be finite in order to gain a test result in a reasonable time. A test purpose selects the (finite) part of the test case behavior which is required by the property. In the SaMsTaG approach test purposes are formalized by finite automata (FA) [6]. We intend to follow this approach.

2.3 Conformance Relation

The aim of conformance testing is to prove a relation between the traces of the specification and the traces of the implementation. The relation is called *conformance relation*.

A chosen conformance relation influences the test purpose and test case generation heavily. For example, the proof of behavioral equivalence between specification and implementation requires that all state transitions have to be checked. Roughly spoken, a conformance relation defines the coverage of specification and implementation by the test cases.

2.4 Restrictions and Assumptions

It is not possible to prove all possible conformance relations for arbitrary specifications and implementations. For example, a behavioral equivalence can only be tested by a finite test if specification and implementation behave like FSMs with some special properties. This means that the chosen conformance relation restricts the class of testable specifications an implementations.

Besides *restrictions* which are implied by the conformance relation may exist further information, in the following called *assumptions*, which is not part of the specification but which may be useful to facilitate the test purpose generation, e.g. known length restrictions on signal buffers. Therefore we intend to make restrictions and assumptions explicit in order to use them as a tool for controlling the test purpose generation.

3. PROCEDURE

We structured the procedure of generating test purposes in two distinct steps (Figure 4). The first step concerns the reduction of the specification given by an LTS to a testable specification. The second step comprises the concrete test purpose generation.

3.1 Reduction of the LTS

The first step of the test purpose generation is the reduction of the LTS to a *testable* specification. Testable means that the result of the reduction is a specification for which we are able to generate a complete set of test purposes and, afterwards, a complete test suite. Complete means that the application of the test suite proves the chosen conformance relation between reduced specification and implementation. The reduction may be performed by four different techniques:

1. Behavior which is inconsistent with restrictions and assumptions may be omitted.

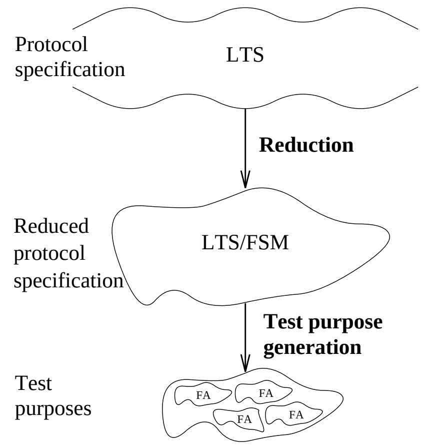

Figure 4: A view of the procedure

2. Heuristics which are applied when the LTS is simulated may be used to abstract from unimportant states and state transitions.

3. The state space of an LTS may be reduced by building equivalence classes of states, i.e. a set of states is folded onto one state.

4. If the specification is already reduced to a finite state machine some well known reduction algorithms for FSMs may be applied.

According to the chosen techniques the reduced specification may be be representable by an LTS or an FSM. The applied reduction techniques have to be selected carefully. Experts knowledge may be necessary to shrink the original specification in a reasonable manner.

Often the test process is influenced by time and cost constraints. Such constraints can be considered during the reduction step. In most cases the size of an automatically generated test suite is directly related to the size of the specification. A series of different reductions can be used to make a specification testable (cf. Figure 5).

3.2 The generation of test purposes

The concrete test purpose generation depends on the chosen conformance relation. For example, the proof of a behavioral equivalence between two FSMs requires the test of all possible state transitions. This means that for each state transition a single test purpose, describing one state transition of the specification, has to be generated. Other conformance relations may require other kinds of test purposes.

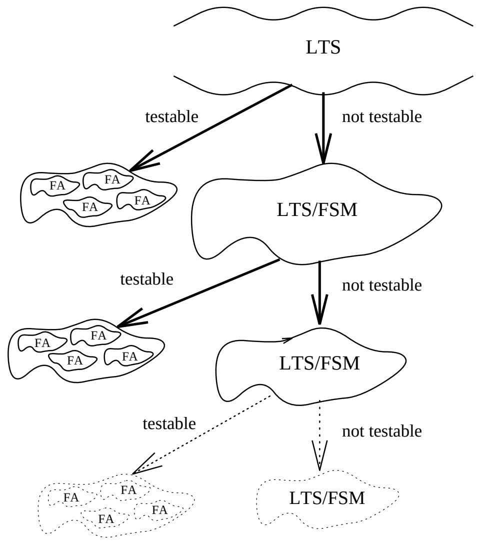

Figure 5: Making a specification testable

The acceptance of an automatically generated test purpose also depends on its representation. For example, test purposes which focus on the message exchange between different entities of a communication system can be specified adequately by means of the frequently used graphical language Message Sequence Chart (MSC) [7, 8]. Other kinds of test purposes may require other formalisms for their representation.

Our research on the generation of test purposes will consider the character of test purposes for different conformance relations and their user-friendly representation. This work will start by analyzing conformance relations and test purposes of well known test case generation methods for FSMs. A summary of these methods can be found in [5].

4. PROJECT STRUCTURE

The work on the project is structured into the tasks (1) *System reduction*, (2) *Test purpose generation*, (3) *Case study*, (4) *Literature study*, and (5) *Publications* (cf. Figure 6).

The problems to be tackled by the Tasks 1 and 2 have been sketched in Section 3. We intend to finish the work on Task 1 in the first project year and to focus on Task 2 in the second year.

The aim of Task 3 is to show the usability of all developed methods and algorithms by applying them to a real world example. The work on Task 3 will be carried out in parallel with the Tasks 1 and 2.

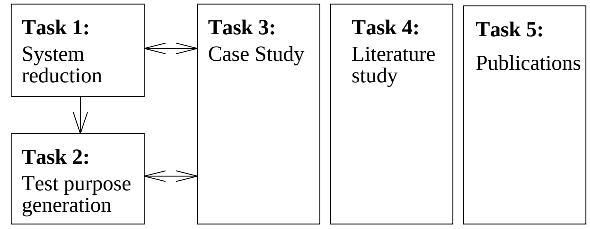

Figure 6: The project tasks

The Tasks 4 and 5 are necessary to compare our work with existing and forthcoming methods. They will be carried out in parallel with the other tasks throughout the whole project.

REFERENCES

[1] ISO/IEC JTC 1/SC 21 N. *Information Technology - OSI - Conformance Testing Methodology and Framework.* International Multipart Standard 9646, ISO/IEC, 1992.

[2] D. Hogrefe. *Estelle, LOTOS und SDL - Standard Spezifikationssprachen für verteilte Systeme.* Springer Verlag, 1989.

[3] J. Kroon, A. Wiles. *A Tutorial on TTCN.* In Proceedings of the 11th International IFIP WG 6.1 Symposium on Protocol, Specification, Testing and Verification, 1991.

[4] J. Grabowski, D. Hogrefe, R. Nahm. *Test Case Generation with Test Purpose Specification by MSCs.* In O. Faergemand and A. Sarma, editors, SDL'93 - Using Objects. North-Holland, October 1993.

[5] G. J. Holzmann. *Design and Validation of Computer Protocols.* Prentice-Hall, 1991.

[6] R. Nahm. *Conformance Testing Based on Formal Description Techniques and Message Sequence Charts.* PhD thesis, University of Berne, February 1994.

[7] J. Grabowski, P. Graubmann, E. Rudolph. *The Standardization of Message Sequence Charts.* In Proceedings of the IEEE Software Engineering Standards Symposium 1993, September 1993.

[8] ITU Telecommunication Standards Sector SG 10. *ITU-T Recommendation Z.120: Message Sequence Chart (MSC).* Geneva, June 1992.