Extending Conformance Testing Concepts to Performance Testing

I. Schieferdecker and A. Rennoch

GMD Fokus, STEP
Hardenbergplatz 2, D-10623 Berlin

http://www.fokus.gmd.de/step

Abstract

The paper discusses basic concepts to performance test a protocol implementation. Performance testing is understood as an extension to classical conformance testing. We present a methodology and some results of its working steps, that were obtained from applying it to the test of the ATM signalling protocol.

Introduction

Conformance testing addresses the functional equivalence between an implementation and its specification based on the notion of implementation relations. The conformance assessment process bases on the development and application of test suites to the implementation under test (IUT), i.e. the abstract and executable test suites play the central role in this process. Important questions in this context are about the procedure for the generation of the test cases and on the quality of the tests. The latter aspect includes the coverage of the test w.r.t. the specification but also their validity (free of syntactic and static/dynamic semantic errors) and applicability (necessity of any adaptation to the IUT).

Currently, the production of industrial relevant test suites is most often done manually, tool support is restricted on the test suite editor, analyser and sometimes simulators. Since this work is very expensive, great efforts are undertaken to find algorithms for the (semi-)automatic derivation of test cases from a formal description of the implementation specification (see e.g. [FMCT]).

Due to e.g. new multimedia communication services over high-speed networks, time and performance issues of communication protocols are gaining in importance and implies new requirements on the implementations but also the related test suites. They are very often the keys to interoperability problems of protocol entities. In particular, the detailed questions tackle (wrong?) settings of timers which are e.g. responsible for correct frequencies of event streams (i.e. throughput, bandwidth, delay jitter etc.) or interstream synchronisation (e.g. voice / video). Also execution environments may decrease the protocol performance (and Quality of Service). The measurement of time consumption of single protocol entities helps in identifying inefficient implementations (resp. parts of them). Consequently timing and performance aspects have to be reflected in the protocol specification, but also the testing methodology has to be extended to address these issues.

In the next sections we describe our approach in a formally supported protocol conformance and performance testing methodology. We also present the results of the working steps of this methodology, which have been applied to the ATM signalling protocol [Q.2931].

Approach

Exhaustive testing of all heterogeneous peers of protocol entities is not feasible, due to practical (e.g. economic) and theoretical (e.g. infinite data value space) reasons. A practical solution to this problem has been given with the classical protocol conformance testing methodology [CTMF]. In this methodology no language prerequisites are specified for the protocol description, but the formal approach of [FMCT] starts from a protocol described in one of the standardized FDTs. We concentrate on SDL [Z.100] due to its wide acceptance and usage in the telecommunication industry.

For the introduction of time requirements in protocol descriptions two approaches are possible: There are valuable approaches for the integration of timing constructs into all of the standardized FDTs. Alternatively time constraints are separated from the functional (formal) protocol specification. In the latter case there are e.g. suggestions to add real-time temporal logic formulas in addition to state-transition FDTs [Leu95]. We also follow the second approach but aim to extend Message Sequence Charts (MSC [Z.120]) with time extensions (TMSC, i.e. timed MSC) and to use them in combination with SDL. This approach has been inspired by existing work on the usage of MSCs for (functional) test purpose description during the derivation of test cases from formal protocol specifications [GNSH94] and proposals to use MSC scenarios in combination with SDL, for monitoring system performance and system behaviour, which have been specified with SDL [DDL95].

We intend to allow the extension of the MSC signals by explicit / maximal delay constraints (see denotation “{t}” in figure 1a) and measurements of time (denoted by “?t” in figure 1b), which have the semantics of delay operators, life reducers and time measurements as introduced in TE-LOTOS [TE-L]. It is still necessary to provide a formal semantics for TMSC since timing issues in MSCs are restricted to explicit timers only [MSC96]. Similar to [Mau96] these semantics are planed to be defined with a process algebraic notation.

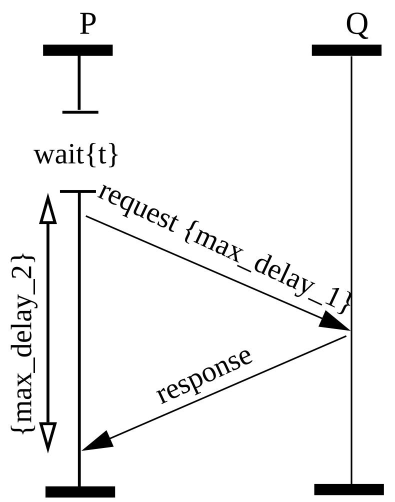

Figure 1a: Waiting and maximal delay constraints in TMSCs

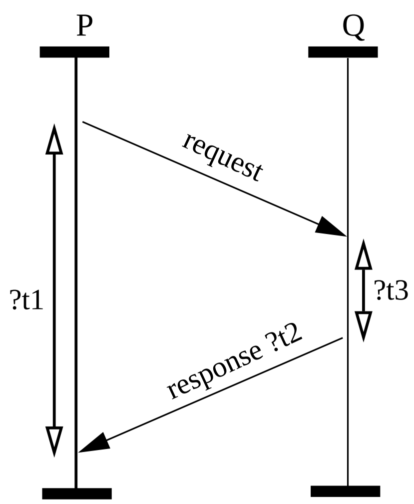

Figure 1b: Time measurements in TMSCs

To sum up, the testing methodology [SR96] is based on a functional protocol specification in SDL together with TMSC for the description of test purposes (including performance requirements). Performance-oriented test cases will be derived from the formal protocol specification where the path selection is according to MSCs.

TTCN is the only standardized test notation [CTMF, part 3]. It has also to be extended for the specification of time measurements and constraints. E.g. [GW96] proposes to extend concurrent TTCN to express non-functional requirements like time by introducing time windows for the enabling of test statements. Additionally we intend to allow the introduction of maximal time constraints even for test sequences with multiple test events. And we plan to include explicit waiting phases (fixed delays) and the control of load-generators or network monitors in the test specification, too.

Further steps in the methodology are the compilation of the test suite to executable code for a specific test equipment and the test execution in a given runtime test environment. This strict formal approach allows a test result analysis w.r.t. the origin protocol specification, including a full test suite validation and a coverage analysis.

A distributed test system architecture is needed taking into account multiple observation and measurement points of the System under Test (SUT) and several local testers together with a coordinating tester component. Although CTMF has been extended to cover multi-party testing, some generalization seems necessary, since the test environment should be capable to e.g. measure, control and evaluate QoS of the underlying communication service. A possible approach is illustrated in figure 2. This architecture has been taken into account for QoS testing [QoSMM], too.

Our overall premise is the priority of available tools in opposite to prototyping of new tools, i.e. we will try to reuse existing tools as far as possible. Own implementations are required for new language constructs and the test case derivation algorithms. In the latter case we will follow the approach, which includes the transformation of the (SDL) protocol specification to an Input/Output Labelled Transition System [Tre95]. There exist a number of tools for the development and validation of SDL specifications but less established tool support for TTCN test suite analysis and validation. We found the SDT/ITEX [TL] to be the only tool environment for consistency checks

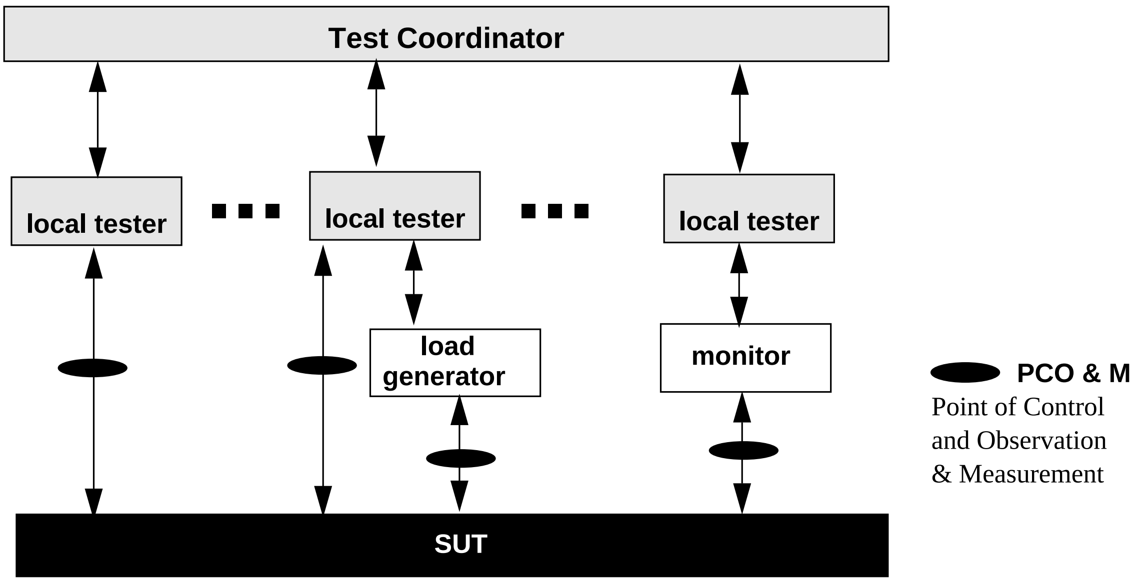

Figure 2: Performance Testing Architecture

between SDL and TTCN. The test suite derivation facilities of this tools allow an automatic static declaration generation but are limited by a semi-automated step-by-step simulation during the development of the dynamic test behaviour.

ATM Signalling Testing

For our project we have selected the ATM signalling protocol at the User-Network Interface. This protocol is a good candidate for performance testing since its virtual connection setup phase allows the analysis of the signalling mechanisms and timer behaviour. Based on this setup we also intend to check the (negotiated) QoS characteristics in different traffic classes (VBR, CBR).

The ITU-T provides an SDL specification [Q.2931] of the signalling protocol and the ATM Forum contributes the TTCN Abstract Test Suite for General Call, Clearing and Error Tests [AFT1, AFT2]. The first goal was the harmonization of both documents.

The ITU-T specification of the ATM signalling protocol has been slightly extended to be processable by SDT. This was due to informal description parts (which were even inconsistent) and minor defects with e.g. signal lists and primitive usage. In addition we found in the SDL specification some improper realisations (in the case of re-establishment of SAAL connections) of the (informal, i.e. english text) protocol description and important open issues of the protocol mechanisms (e.g. within the Restart procedure).

The SDL specification had to be adapted logically to the TTCN test suite. We had to introduce an adaptation process SAAL in order to specify the relation between SAAL service access point primitives (used in Q.2931) and the pure signalling message exchange (used in the ATM Forum test suites). In addition, an application process AP has been specified to serve the upper boundary interface of Q.2931. Last but not least, we took into account the slight differences between Q.2931 (defining public UNI) and the UNI 3.1 based test cases (addressing private UNI). In such cases, we followed the intention to modify our SDL system in order to validate the original ATM Forum test suites.

The final harmonization (dynamic requirements) is based on the cross simulation with a (automatic generated) simulator of the extended Q.2931 SDL specification and a (automatic generated) simulator of the ATM Forum test suite. For each test case run we obtained an MSC, which was verified against the SDL specification. The set of MSC traces is also useful to generate corresponding statistics of the test cases.

We discovered a lot of typos and missing or wrong type declarations in the TTCN tests, but in the behaviour parts also wrong applications of TTCN (e.g. wrong pattern of message parameters to be received), missing initialization of test suite variables, wrong indentations of test statements, incompatible message parameter values. From the coverage analysis we can see e.g. which system states have not been visited during the execution of the test suites. A detailed list of the errors and results has been reported to the ATM Forum [AF96].

Conclusions

The establishment of a complete tool-support for the presented performance testing methodology is the main goal. It will help to identify (semi-)automatically interoperability problems of protocol entities and addresses also performance bottlenecks or inefficiencies of protocol implementations.

Until now we've not generated performance-oriented test cases from the signalling specification. The strict use of the formal protocol description in the testing methodology leads to valuable results and gives customers more confidence to the testing assessment process.

References

[AFT1] S. Yoo (Ed.): Abstract Test Suite for the UNI 3.1 Signalling for the User Side: General Outgoing Call and Incoming Call Tests, ATM Forum Technical Committee, Testing Subworking Group, ATM Forum/95 - 0584R1, Toronto Aug. 1995.

[AFT2] S. Yoo (Ed.): Abstract Test Suite for the UNI 3.1 Signalling for the User Side: General Clearing and Error (general) Test Section, ATM Forum Technical Committee, Testing Subworking Group, ATM Forum/95 - 0858, Toronto Aug. 1995.

[AF96] I. Schieferdecker, A. Rennoch: Validation and Correction of ATM Signalling Abstract Test Suites, Proposal at the ATM Forum Technical Meeting, Orlando (Florida) June 1996.

[CTMF] ISO/IEC: IS 9646 Information technology - OSI - Conformance testing methodology and framework, Geneva 1996.

[DDL95] P. Dauphin, W. Dulz, F. Lemmen: Specification-driven performance monitoring of SDL/MSC-specified protocols. In: [PTS95], p. 409 - 424.

[FMCT] ISO/IEC JTC1/SC21: Information technology - OSI - Formal methods in conformance testing. Working Draft, 1995.

[GI96] U. Herzog, H. Hermanns (Hrsg.): Formale Beschreibungstechniken für verteilte Systeme, Arbeitsberichte des Instituts für Mathematische Maschinen und Datenverarbeitung, Band 29, Nummer 9 (http://www7.informatik.uni-erlangen.de/tree/IMMD-VII/Announcements/fachgespraech96/programm.html). Erlangen 1996.

[GNSH94] J. Grabowski, R. Nahm, A. Spichiger, D. Hogrefe: Die SAMSTAG Methode und ihre Rolle im Konformitätstesten. PIK 17(4), p. 214-224, 1994.

[GW96] J. Grabowski, T. Walter: Quality-of-Service Testing. Specifying Functional QoS Testing Requirements by using Message Sequence Charts and TTCN. In: [GI96], p. 109 - 118.

[Mau96] S. Mauw: The formalization of Message Sequence Charts. Computer Networks and ISDN Systems, 28(12):1643-1657, 1996.

[MSC96] ITU-T, Study Group 10: Draft Recommendation Z.120, Apr. 1996.

[Leu95] S. Leue: Specifying Real-Time Requirements for SDL Specifications - A Temporal Logic-Based Approach. In: [PSTV95], p. 19 - 34.

[PSTV95] P. Dembinski, M. Sredniawa: Protocol Specification, Testing and Verification, XV. Chapman & Hall, London 1995.

[PTS95] A. Cavalli, S. Budkowski (Ed.): IWPTS'95 Conference Proceedings, Evry (F) Sept. 1995.

[Q.2931] ITU-T Recommendation Q.2931: Broadband Integrated Services Digital network (B-ISDN). Digital Subscriber Signalling System No.2 (DSS 2). User-Network Interface (UNI) Layer 3 Specification for Basic Call/Connection Control, Feb. 1995.

[QoSMM] ISO/IEC JTC1/SC21: Quality of Service - Methods and Mechanisms. Working Draft, 1996.

[SR96] I. Schieferdecker, A. Rennoch: Formal based testing of ATM signalling. In: [GI96], p. 119 - 128.

[TE-L] Time Extended LOTOS LOTOS, ISO draft working paper, source Belgium and Spain, Sept. 1995.

[TL] Telelogic AB: SDL Development Tool (SDT), Interactive TTCN Editor and Executor Development Environment (ITEX-DE). Manuals version 3.02, Malmö (Sweden), Nov. 1995.

[Tre95] J. Tretmans: Testing Labelled Transition Systems with Inputs and Outputs. In: [PTS95], p. 461 - 476.

[Z.100] ITU: Specification and Description Language (SDL), Geneva, 1989.

[Z.120] ITU: Message Sequence Charts, Geneva, 1994.