ISSN 2277-3061

Towards Standardized Conformance Test Suite for ISO Transport Layer Protocol

Hazem El-Gendy, Ihab E. Talkhan
Ph. D. & P. Eng. Faculty of Computer Science & IT Ahram Canadian University, 6th of October City, Egypt
H_ElGendy@masrawy.com
Ph. D. & P. Eng.Faculty of Engineering Cairo University, Giza, Egypt
italkhan@aucegypt.edu

ABSTRACT

In this paper, we develop a sound Conformance Test Suite for the Transport Layer Protocol Internationally standardized by both ISO and IEC. This is to test the implementations of the protocol, promote and facilitate standardized test suites, and promote the use of formal methods. We use formal methods for the generation of testing sequences to make the results sound. The protocol is formally specified in Lotos; the ISO/IEC Formal Description Technique for computer/communications protocols and distributed systems.

Indexing terms/Keywords

Testing, Computer/Communications Protocols, Protocol Data Unit (PDU), Standards, Conformance/Certification Testing, International Standards Organization (ISO), Open Systems Interconnections (OSI) Transport Layer.

Academic Discipline And Sub-Disciplines

Provide examples of relevant academic disciplines for this journal: E.g., History; Education; Sociology; Psychology; Cultural Studies;

SUBJECT CLASSIFICATION

E.g., Mathematics Subject Classification; Library of Congress Classification

TYPE (METHOD/APPROACH)

Provide examples of relevant research types, methods, and approaches for this field: E.g., Historical Inquiry; Quasi-Experimental; Literary Analysis; Survey/Interview

Council for Innovative Research
Peer Review Research Publishing System
Journal: International Journal of Computers & Technology
Vol 12, No.2
editor@cirworld.com
www.cirworld.com, member.cirworld.com

INTRODUCTION

Computer and Communications protocols are widely used in various areas. This widespread use necessitated standardization of them by major organizations such as International Standardization Organizations (ISO), International Electro Committee (IEC), and Telecommunications Standardization Section of the International Telecommunications Union (TSS/ITU). ISO and IEC have jointly developed a Transport Layer Protocol Standard in recognition to its importance [2,3,4,14].

Reliability of the implementations of these protocols is essential to the widespread use of such protocols and to networking. Here, comes the role of Conformance/Certification Testing. Conformance tests are "intended to provide as thorough testing of an implementation as is practical (as we cannot do exhaustive testing), over the full range of requirements specified in a standard" [3,4,10-14]. Conformance testing involves:

a) The design of tests appropriate to the claimed subsets (conformance claims) of the protocol standard (PS),

b) The unambiguous specification of these tests in a standard test language according to the test architecture chosen,

c) The implementation of the test specification and the execution of the executable tests,

d) The interpretation of test results to determine the degree of conformance of an implementation to the protocol specification based on the conformance claim, taking into account the limitations of the test architecture. The interpretation of test results involves:

i) An audit of test traces to verify that test specification was faithfully implemented.

ii) Determination of verdict (pass, fail, inconclusive, canceled) on every executed test.

iii)Rationalization of the verdict of every test by reference to relevant components of the protocol specification, and the conformance claim.

e) The validation of the tests based on the results of test execution coupled with rigorous targeted analysis of the protocol specification. This step may involve discovering and documenting errors in the protocol standard, the design, the implementation, or in the test suite (both abstract and executable forms).

So, the design of a test suite constitutes a very important part in the process of conformance testing. This is because the effectiveness of conformance testing in detecting non-conforming implementations depends heavily on the coverage of the test suite. By the coverage of the test suite, we mean the ability to detect different types of errors (faults) in the implementations by applying this test suite. Thus, formal methods for designing adequate test suites are required. The use of formal methods provides correct testing sequences (rather than error prone testing for intuitive testing), consistent results, better coverage, extendable coverage, and both time and cost effective test derivation methods.

In this paper, we develop Conformance Test Suite (CTS) for the ISO/IEC Transport Layer Protocol Class 2 (TP2). We use formal methods for the generation of testing sequences and formal specification of the protocol in the ISO/IEC Lotos standard Formal Description Technique for protocols. Both the control-flow aspects of the protocol and the data-flow aspects are tested.

This is to provide sound CTS for TP2 as a step towards standardized CTSs for the various protocols, to check the easiness in applying formal methods as well as gaining experience from this experiment.

The rest of this paper is organized as follows. In the next Section, we provide an overview of TP2. In Section III, we develop the testing sequences for the control flow aspects. Sequences to test the data-flow aspects are developed in Section IV. We conclude the paper in Section V.

OVERVIEW OF ISO TP2

TP2 has 16 different conformance classes and the capabilities of negotiation and flow control. TP2 has the following capabilities:

1) Initiating, upon request from local user, the establishment of a connection over which data can be transferred;

2) RC: Responding to a request, from the remote peer entity, to establish a connection;

3) DT: transferring Data over an established connection;

4) Disconnection of a connection;

5) $NegO_{Res}$: Negotiation of options (e.g., parameter values), while responding to a request from remote entity to establish a connection;

6) $NegO_{Init}$: Negotiation of options, while requesting (upon request from local user) the initiation of a connection establishment;

7) FC: End-to-end flow control that allows the receiver to signal to the sender to stop sending data at a rate higher than what the receiver can consume;

8) Multiplexing: The use of a single Network connection for several Transport connections provided that the Network prefixes of the Transport addresses are the same;

9) Segmentation: The ability to handle arbitrarily long Transport service data units (SDUs).

Capabilities describe services to be implemented based on the protocol; so, they are described in terms of successful outcomes. TP(2) state diagram is given in [2]. The formalization of the standard in Lotos is as given next [2].

TPC2[ ] := IC >> (DT [> DC)

[] RC >> (DT [>DC)

Where:

IC := ?TConReq ?dest_address ?proposed_options;

!CR !dest_address !opt !R_credit; Wait-CC

Wait-CC := ?CC ?options ?seq_no; ([p5] ---> !TConConf ?opt; exit

[NOT(p5)] ---> !TDisInd !procedure_error;

!DR !procedure_error !false; Wait-DC)

[] ?DR ?disconnect_reason; !TDisInd !disconnect_reason; TPC2

Wait-DC := ?DC; !TDisConf; TPC2

[] ?DR; TPC2

RC := ?CR ?to_ad ?opt ?seq_no; ([p4] ---> (!TConInd !to_add !opt; Wait-TConResp

[] [NOT(p4)] ---> !DC; TPC2

Wait-TConResp := ?TConResp ?accepted_options; ([accepted_options ≤ opt] ---> !CC !opt !R_credit; exit

[] ?TDisReq; !DR !TS_user_initiated !true; Closing)

Closing := !TDisConf; TPC2

DT :=?TDataReq ?seq_no ?data ?EoSUD;[S_credit > 0]--->i; !dt !seq_no!data!EoSDU; DT

[] ?dt ?send_seq ?EoSDU; [R_credit ≠ 0 & send_seq = Trseq] ---> i; !TDataInd !data !EoSDU; DT

[] ?U_Ready ?credits; i; !AK !Trseq !R_credit; DT

[] ?AK ?seq_no ?creidt; i; !TSReady; DT

DC := ?TDisReq; !DT !TS_user_initiated !false; Wait-DC

[] [reason ≠ TS_user_initiated] ----> i; !TDisInd !reason; !DR !reason !false; Wait-DC

[] ?DR; !TDisInd !PDU_disconnect_reason; !DC; Closing

GENERATION OF SEQUENCES FOR TESTING CONTROL FLOW ASPECTS

In this Section, we apply the formal method to generate sequences for testing the control flow aspects of TP2 formally specified in Lotos as given in the previous section. Firstly, to focus on only control flow aspects, we ignore parameters in the Lotos-behavior expressions; consequently, we get the following version of the specification.

TPC2[ ] := IC >> (DT [> DC)

[] RC >> (DT [> DC)

Where:

IC := ?TConReq; !CR; Wait-CC

Wait-CC := ?CC; (!TConConf; exit

[] !TDisInd; !DR; Wait-DC)

[] ?DR; !TDisInd; TPC2

Wait-DC := ?DC; !TDisConf; TPC2

[]?DR; TPC2

RC := ?CR; (!TConInd; Wait-TConResp

[] !DC; TPC2)

Wait-TConResp := ?TConResp; (!CC; exit

ISSN 2277-3061

[] ?TDisReq; !DR; Closing)

Closing := !TDisConf; TPC2

DT := ?TDataReq; !dt; DT

[] ?dt; !TDataInd; DT

[] ?U_Ready; !AK; DT

[] ?AK; !TSReady; DT

DC := ?TDisReq; !DR; Wait-DC

[] !TDisInd; !DR; Wait-DC

[] ?DR; !TDisInd; !DC; Closing

The testing method in [3,4] firstly generates from the given specification a Restricted Behavior Tree T that represents the control flow aspects of the system. Then, unique signatures for every node are derived to enhance the testing coverage. Applying the method [3,4], we get T shown in Figure 1 and the following unique signatures of its nodes.

UESeq(nl_0) = ?TConReq UESeq(nl_1) = !CR

UESeq(nl_2) = !TConInd UESeq(nl_3) = ?CC

UESeq(nl_5) = ?TconResp

UESeq(nl_6) = !TDisInd @ TConReq

UESeq(nl_7) = !TConConf UESeq(nl_8) = !CC

UESeq(nl_{10}) = ?TdataReq

UESeq(nl_{11}) = !DR ! ?DR

UESeq(nl_{13}) = !DR @ !TDisConf UESeq(nl_{14}) = !dt

UESeq(nl_{15}) = !DataInd UESeq(nl_{16}) = !AK

UESeq(nl_{17}) = !TSReady

UESeq(nl_{20}) = !TDisInd @ !DC

UESeq(nl_{21}) = ?DC UESeq(nl_{22}) = !TDisConf

UESeq(nl_{29}) = !DC

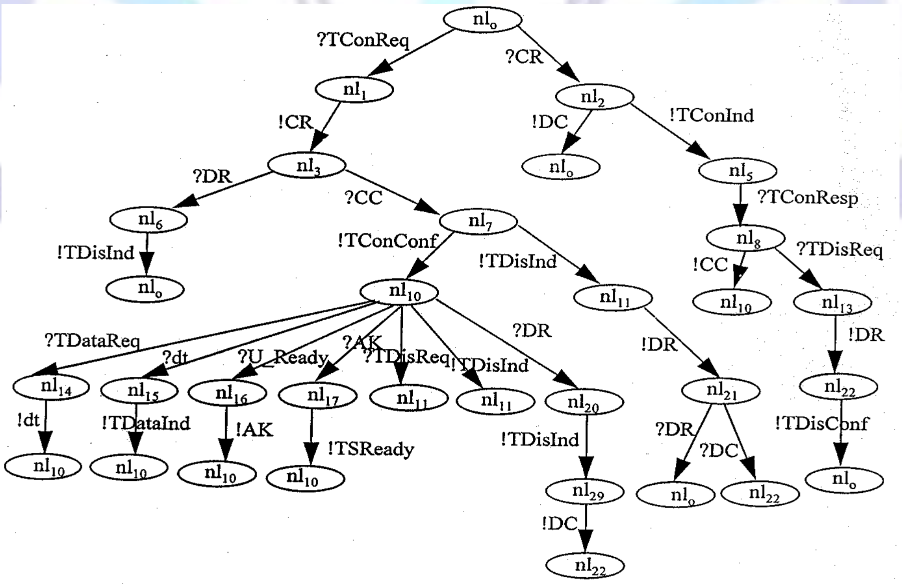

Figure 1. Restricted Behavior Tree of the ISO Transport Layer Protocol Class 2

The testing sequences are derived from T and the unique signatures [3,4]. So, the optimized[1] testing sequences are:

?TconReq !CR ?DR !TdisInd ?TConReq,

?TConReq !CR ?CC !TConConf ?TDataReq !dt ?TDataReq,

?TConReq !CR ?CC !TConConf ?dt !TDataInd ?TDataReq,

?TConReq !CR ?CC !TConConf ?U_Ready !AK ?TDataReq,

?TConReq !CR ?CC !TConConf ?AK !TSReady ?TDataReq,

?TConReq !CR ?CC !TConConf ?TDisReq !DR ?DR,

?TConReq !CR ?CC !TConConf !TDisInd !DR ?DR,

?TConReq !CR ?CC !TConConf ?DR !TDisInd !DC !TdisConf,

?TConReq !CR ?CC !TDisInd !DR ?DR ?TConReq,

?TConReq !CR ?CC !TDisInd !DR ?DC !TDisConf,

?CR !DC ?TConReq,

?CR !TConInd ?TConResp !CC ?TDataReq,

?CR !TConInd ?TConResp ?TDisReq !DR !TdisConf ?TConReq,

GENERATION OF SEQUENCES FOR TESTING DATA FLOW ASPECTS

In this Section, we apply the method presented in [13] to generate sequences for testing the data flow aspects of TP2. The method starts by generating a ‘Data Oriented’-Restricted Behavior Tree T’ that is an extension of T where data aspects are also presented. So, we get T’ shown in Figure 2.

Then, the method identifies from T’ the definitions and uses of the various parameters to test their in-dependencies. So, we get the following definitions/uses for TP2:

nl_0: Def(dest_address), Def(proposed_options), Def(to_add), Def(opt), Def(seq_no)

nl_1: Def(dest_address), Cuse(proposed_options),

nl_2: PUse(opt), CUse(opt), CUse(to_add)

nl_3: Def(options), Def(seq_no), Def(disconnect_reason)

nl_5: Def(accepted_options)

nl_7: PUse(options), CUse(options)

nl_8: PUse(options), CUse(options), CUse(credit)

nl_{10}: Def(seq_no), Def(data), Def(EoSDU), Def(credit), PUse(disconnect_reason)

nl_{11}: CUse(EoSDU), CUse(disconnect_reason)

nl_{13}: CUse(disc_reason), CUse(EoSDU)

nl_{14}: CUse(seq_no), CUse(data), CUse(EoSDU), PUse(credit)

nl_{15}: CUse(data), CUse(EoSDU), PUse(credit), PUse(seq_no)

nl_{16}: CUse(seq_no), CUse(credit)

nl_{20}: CUse(disconnect_reason)

---

[1] If a testing sequence is a sub-sequence of another testing sequence, we include only the latter (the longer). For example, the testing sequence to verify node nl7 is?” TConReq !CR ?CC !TConConf” which is a sub-sequence of the first testing sequence (TConReq !CR ?CC !TConConf ?TdataReq !dt). So, we include only the first testing sequence.

ISSN 2277-3061

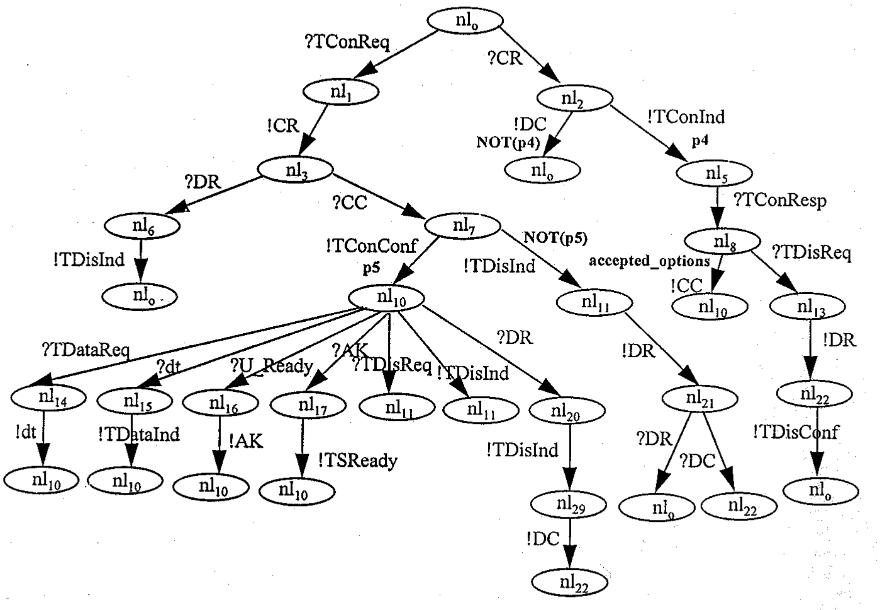

Figure 2. 'Data Oriented'-Restricted Behavior Tree T' of ISO Transport Layer Protocol Class 2

Then, we get the following testing sequences:

- Predicate dets_address: ?TConReq !CR for definition in nl_0 and Cuse in nl_1
- Predicate proposed_options: ?TConReq !CR for definition in nl_0 and Cuse in nl_1
- Predicate opt: ?CR !TConInd for definition in nl_0 and both PUse and CUse in nl_2, and ?CR !DC for definition in nl_0 and PUse in nl_2
- Predicate to_add: ?CR !TConInd for definition in nl_0 and CUse in nl_2
- Predicate seq_no: ?CR !TConInd ?TConResp !CC ?TDataReq for definition in nl_0 and CUse in nl_{10}
- Predicate options: ?CC !TConConf for definition in nl_3 and both PUse and CUse in nl_7, and ?CC !TDisInd for definition in nl_3 and PUse in nl_7
- Predicate accepted_options: ?TConResp !CC for definition in nl_5 and both PUse and and ?TConResp ?TDisReq definition in nl_5 and PUse in nl_8
- Predicate seq_no: ?TDataReq !dt for definition in nl_{10} and CUse in nl_{14}, ?dt !TDataInd for definition in nl_{10} and PUse in nl_{15}, and ?U_Ready !AK for definition in nl_{10} and CUse in nl_{16}
- Predicate data: ?TDataReq !dt for definition in nl_{10} and CUse in nl_{14}, and ?dt !TDataInd for definition in nl_{10} CUse in nl_{15}
- Predicate EoSDU: ?TDataReq !dt for definition in nl_{10} and both PUse and CUse in nl_{14}, and ?dt !TDataInd for definition in nl_{10} and CUse in nl_{15}
- Predicate credit: ?TDataReq !dt for definition in nl_{10} and both PUse and CUse in nl_{14}, and ?dt !TDataInd for definition in nl_{10} and PUse in nl_{15}, and ?U_Ready !AK for definition in nl_{10} and CUse in nl_{16}

CONCLUSIONS

The methods are quite applicable to the International Standards such as the ISO Transport Layer Protocol Class 2. The methods are applicable to systems specified in Lotos that describe finite behavior as well as infinite behavior and involve various Lotos operators such as enabling and disabling operators. They are also applicable to those systems that involve any form of non-determinism and internal events. The methods do not impose any restrictions on the number of Lotos processes composing the protocol nor on the Lotos constructs. So, they are applicable to various systems that can be specified in Lotos.

Given that the methods are both formal and supported by algorithms, the application of the methods was quite straightforward. This reduces the likelihood that there be errors. Also, the support of algorithms produces the same results even when different people are involved. Furthermore, the results can be analyzed and extended. The time required was much shorter than the usual time for intutively developing the testing sequences. This is assuming familiarity with the standard under-consideration, and availability of a formalization of such standard in Lotos.

As the method for testing control flow aspects verifies the processes, it has a coverage that is as good as any other formal method. Also, the method for testing the data flow aspects covers various forms of data dependency.

The formal methods for deriving testing sequences cover various aspects of real time systems. So, they are applicable and needed to test real time systems that involve various time constraints; this is for the following reasons:

1. The methods are applicable to systems that can be specified using Lotos. So, the methods' applicability is as wide as the scope of applicability of the ISO Lotos language. ISO has developed Lotos to be used in specifying EVERY communications protocol standard. Communications protocols are, by nature, real time systems. Due to the huge varieties of purposes/applications of communications protocols, they involve various forms of time constraints and time dependencies. Furthermore, ISO has been used in formally specifying various distributed real time systems. The Europeans in their collaborative research project on formal specification and verification of real time systems have chosen Lotos as the specification FDT [8,9].
2. The methods are capable of testing various forms of time constraints and interrelations among parameters whether these are reflected by any type of statement (a logical statement or a calculation statement) or any form of interaction with other systems (receiving an input or sending an output). Furthermore, the methods are not limited to any specific form for specifying such time constraints or parameters interrelations.

Examples of time constraints that are covered are:

- Time-out constraints: this is a typical parameter interrelation in networks where PDUs (Protocol Data Units) may be lost or encounter somewhat long delays. In such cases, the protocol designers include, in the protocol design, mechanisms for appropriate course of actions (e.g., the sender retransmits the PDU).
- Determination of fastest route to send on it a PDU: this is a possible case in certain multimedia network protocols where the PDUs are required to be delivered within pre-specified time limits. These networks usually have very low error rate and fast transmission. In such cases, the incoming PDUs contain, as part of their parameters, network delay indicators to estimate the current delay on various outgoing links. Also, it is always assumed that for every node, one of its outgoing links is fast enough to guarantee delivery of the PDU to the final destination within the allowed delay time. Then, the intermediate node, on receiving a PDU, uses these parameters' values to determine the outgoing link to send on it the PDU; the link with the shortest delay time (if a route that is likely to lead to receiving the PDU at the destination within a specific threshold does not exist, the node just deletes the PDU and ignores sending it). So, formal testing of such functionality of these protocols/systems require developing a formal method to generate testing sequences to test the following:
- Calculation statements where the estimated delays of the outgoing links are calculated based on the incoming PDUs parameters (delay indicators). These parameters may have been defined (supplied) before they are used.
- Logical statements as the transmission on a particular outgoing link is conditional to that the estimate of its associated delay is less than a threshold value. In such protocols, the intermediate node uses estimation parameters of the incoming PDUs to attempt to deliver the PDU within a time limit.

Incorporating the PICSP (Protocol Implementation's Conformance Statement Proforma) into the testing process; this may be by parameterizing the test generation as well as structuring the generated test sequences into a Test Suite;

REFERENCES

1. ISO/IEC 8807, "Information Processing Systems - Open Systems Interconnection - Lotos - A Formal Description Technique Based on Temporal Ordering of Observational Behaviour", 1992.
2. Hazem El-Gendy and Nabil El Kadhi, "Using Formal Methods: Importance, Experience, and Comparative Analysis", International Journal on Computing Methods in Science and Engineering, Published in Greece, 2005
3. El-Gendy, H.: A New Method for Deriving Test Sequences For Protocols Specified in LOTOS. Proceedings of the International Conference on Networks sponsored by the International Association of Science and Technology for Development (IASTED), Orlando, Florida, USA, (January 8-10, 1996) pp. 205-208.
4. El-Gendy, H., Abou-Rabia, O.: Derivation of Test Sequences for Protocols Specified in Lotos. Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, Amman, Jordan, (December 1995) pp. 220-225.
5. Hazem El-Gendy and Nabil El Kadhi, "Testing Data Flow Aspects of Communications Protocols, Software, and Systems Specified in Lotos", International Journal on Computing Methods in Science and Engineering, Published in Greece, 2005.
6. Hazem El-Gendy, "Formal Method for Automated Transformation of Lotos Specifications to Estelle Specifications", International Journal of Software Engineering & Knowledge Engineering, USA, Vol. 15, No. 5, October 2005, pp. 1-19.
7. Hazem El-Gendy, "Study of the Characteristics of CT-Equivalence with Proves", Journal of Computational Methods in Sciences and Engineering, Volume 6, Numbers 5, 6, 2006, pp. 171-179.

8. Steen, M., Bowman, H., Derrick, J.: Composition of Lotos Specifications. Proceedings. of IFIP 15th Int. Symposium on Protocol Specification, Testing and Verification, Specification of Concurrent Systems Session, 1996, Warsaw, Poland, (June 1995) pp. 87-102.

9. Sigireanu, M., Mateescu, R.: Validation of the Link Layer Protocol of the IEEE-1394 Serial Bus ("FierWire"): an Experiment with E-Lotos. INRIA Technical Report No. 3172, 1997. A short version of this report is also available in Ignac Lovrek, editor, Proceedings of COST 247 2nd International Workshop on Applied Formal Methods in System Design, Zagreb, Croatia, (June 1997).

10. Lee, D., Yannakakis, M.: Testing Finite State Machines: State Identification and Verification. IEEE Transactions on Computers, Vol. 43, No. 3, (March 1994) pp. 306-320.

11. Hazem El-Gendy, and Nabil El Kadhi, "Unique Input/Output Set: Universal Formal Method for Automated Testing FSM-Based Systems", Proceedings of the 8th World Multi-conference on Systemics, Cybernetics, and Informatics sponsored by the International Institute of Informatics and Systemics (IIIS), Orlando – Florida, USA, July 18-21, 2004.

12. Hazem El-Gendy and Nabil ElKadhi, "New Method for Testing FSM-Based Systems", Proceedings of the 11th IEEE International Conference on Software, Telecommunications, and Computer Networks sponsored by IEEE Communications Society, Vinece-Italy, and Split-Croatia, October 7-10, 2003.

13. Bloor, M. S., Owen, J.: Learning Lessons from Conformance Testing. Journal of Computer Standards & Interfaces, Vol. 17, (1995) pp. 231-251.

14. El-Gendy, H., Probert, R. L.: Conformance Testing Requirements for Formal Development of Testability-Directed PICSP. Proceedings of the International Conference on Computers and Their Applications sponsored by the International Society for Computers and Their Applications, Honolloulou, Hawaai, USA, (March 25-27, 1998) pp. 85-88.

15. El-Gendy, H.: 'Conformance Testing' - based Approach to Interoperability Requirements and Guarantees. Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, Cairo, Egypt, (Dec. 15-18, 1997) pp. 715 - 719.

16. El-Gendy, H.: Testability-Directed Modeling of a Communications Protocol. Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, Rodous, Greece, (October 16-18, 1996) pp. 346 -349. Was also accepted for Publications in the Proceedings of the International Conference on Networks sponsored by the International Association of Science and Technology for Development (IASTED), Orlando, Florida, USA, (January 8-10, 1996).

17. El-Gendy, H.: Towards a Foundation for FDT-Based Conformance Testing Requirements. Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems, Rodous, Greece, (October 16-18, 1996) pp. 1182 - 1185. Was also accepted for Publication in the Proceedings of the International Conference on Networks sponsored by the International Association of Science and Technology for Development (IASTEDS), Orlando, Florida, USA, (January 8-10, 1996).

Authors' biographies

Biography

Dr. Hazem El-Gendy made his Ph. D. in Computer, his M. Sc. in Computer Engineering, and his B. Sc. in Electronics & Communications Engineering. He worked for Cairo University, Ottawa University, Bell-Northern Research, British Telecom, Gezera Academy for Computer & Management Information Systems, Misr International University, Ministry of Endowments, and Ahram-Canadian University. Dr. El-Gendy started the software & consulting firm EPEC Inc. of Ontario/Canada and the consulting firm GATIS Co. of Egypt. He has been engaged in R & D of Desktop Communications Services & Applications, Testing of Distributed Systems, Design of Computer/ Communications Systems for Testability & Interoperability, Intelligent Networks, Using Formal Methods in the lifecycle of protocols and software, Data Packet Networks, and ISDN. Dr. El-Gendy has over 150 contributions to International Standards and international papers (Journals and Conferences) in these areas. Dr. El-Gendy is currently ACU Acting Director of International & External Relations and Chair of the Department of Computer Science (CS) of the Faculty of CS & IT of Ahram-Canadian University and X-Assistant Minister of Endowments. Dr. El-Gendy has been selected for Marquis "Who's Who" in Science and Engineering Reference Book in 2010, USA. Dr. El-Gendy is currently the Manager of the Quality Unit of the Faculty of Computer Sc. & IT of Ahram Canadian University. He is also the General Manager of the International Master Program in Enterprise System Engineering, a joint project/program with a well established set of European Universities in Germany, Sweden, Spain, and Portugal.

Dr. Ihab Talkhan, IEEE member since 1984, he is currently an Associate Professor at both Faculty of Engineering, Cairo University, and Electronics Engineering Department, American University in Cairo, got his Ph.D. in 1993 jointly between Oakland University, Rochester, Michigan & Cairo University. Fields of interest: Computer Engineering, Microelectronics, Digital System Design and Digital Testing, VLSI Device modeling. Dr. Talkhan got a certificate of Merit for outstanding Achievement from Ministry of Health in 2007, and Certificate of Merit for outstanding Academic Achievement from the Mathematical Sciences Department, Oakland University in 1989. Dr. Talkhan is also the Deputy Director of the Computation Scientific Center, Cairo University Also, he is currently the IT advisor for both Minister of Health and Minister of Communication and Information Technology. He has been an IT consultant for different entities UNDP, Egyptian Electric Utility & Consumer Protection Regulatory Agency, Ministry of Electricity & Energy, Ministry of Foreign Trade, Ministry of Foreign Affairs, Ministry of Interior and Others.