# Formal Method to Derive Interoperability Requirements and Guarantees

Hazem El-Gendy,
Ph.D., P.Eng.
Faculty of CS &IT Ahram Canadian University

Magdi Amer,
Ph. D., P. Eng.
Faculty of Eng., Oum Kora Univ.,
Makka, Saudi Arabia,

Ihab Talkhan,
Ph. D., P. Eng.
Faculty of Eng, Cairo Univ.,
Giza, Egypt.

***Abstract*— Interoperability among telecommunications systems, possibly by different vendors, is essential for both the development of many telecommunications networks, and today's civilization development. Interoperability testing is very costly, as it has a complexity of (n**2) for n systems, and somewhat informal. In this paper, we develop a 'Conformance Testing (CT)'-based formal technique to determine interoperability requirements/guarantees. It allows automated derivation of the interoperability' requirements of various networks as well as the interoperability guarantees among different telecommunications systems. This is achieved using static analysis of the conformance classes of the standard and knowledge of the implementation's degree of conformance (DoC) of the telecommunications systems. Consequently, it results in a lot of cost saving in addition to being a formal technique.**

***Keywords—Computer/Communications Protocols and Standards; Conformance Requirements and Classe; Interoperability; Protocol Data Units (PDUs); Capabilities.***

## I. Introduction

Different telecommunications networks are typically by different vendors. This is often associated with many interoperability problems. This is the case even when these products are conforming products to an international standard. Also, different telecommunications products by the same vendor and that play different roles in the network may have interoperability problems in certain cases. Nevertheless, building a multi-vendor network has become a key requirement of every intelligent telecommunications user. This is to relief that user from dependency on any single telecommunications vendor. Such independence, when achieved, results in better economics of the network expansion and more importantly a feature richer network and continuous supply of telecommunications products.

This motivated the work on Interoperability Testing. Methods [8.9.10,11]. Interoperability testing methods developed so far are mainly informal methods that are both protocol dependent and product dependent. By protocol dependent, we mean that the method's applicability is limited to only one protocol. Product dependency means that the method has to be applied for every pair of products for which interoperability is required; consequently, the complexity of the "interoperability testing"-based methods for interoperability analysis is (n**2). So, for n different telecommunications products, all the following are required:

- Designing of n * n Interoperability Test Suites;
- Running the n * n Interoperability Tests Suites; and
- Conducting n * n Interoperability Test Results Analysis.

In this paper, we develop a CT-based method for interoperability analysis and guarantees. CT is the type of testing that aims at increasing the confidence in the correct implementation of the telecommunications products. As CT is also an ISO (which aims at facilitating Open Systems Interconnections) research work, it has to also increase the confidence that conforming implementations interoperate. The CT-based method has a complexity of just n for n different products. As the method depends on static analysis of the standard and the determination of the DoC of every product, no additional tests are required: this includes no design at all of any interoperability test suites. Furthermore, the method is formal and facilitates full automation.

## II. Testability-Directed Specification Of A Protocol

A Testability-Directed Standard of a communications Protocol (TDSP) [1] is a tuple:

TDSP := (P, I, BS, TDPICSP & ConfStat, ServDesc, NCTC)
where:

P: set of service parameters.

I: set of interactions

BS: Specification of allowed protocol behavior. From BS, all the allowed sequences of interactions (BSeq) can be derived. Also, the inter-dependencies between the interaction parameters can be extracted. Each sequence (trace) $BSeq_s = (s, P_s, C_s)$ represents:

*a) A syntactically allowed ordering s of interactions.*

*b) A number of constraints ($C_s$) on the parameters ($P_s$) of the interactions in the ordering s. From BS, it is possible (in principle) to derive all possible traces (BSeq) as follows:*

Let

- S be the set of syntactically allowed orderings of interactions as derived from BS;

- R : Constraints relation

- For each $s \in S$, $\exists\ C_s \subseteq R$ where $C_s$ represents a set of constraints on parameters $P_s$ in s.

Then, the set of all possible traces is denoted BSeq = {(s. $P_s$. $C_s$)} where

$$s\ (=<i,.i_2..\ .\ ,i_n>) \varepsilon\ S$$

$$P_s = \{(p_j,\ v\ (p_j)/\ \exists\ i(p_1,p_{2,}..\ .\ ,p_k) \in s,\ K \geq j \geq 1\}$$

$$C_s=\{(p_i.p_j)/p_i,p_j\ \varepsilon\ P_s\}$$

TDP1CSP: Testability-Directed Protocol's Implementation Confomance Statement Proforma [l].

ConfStat: Conformance Statement

ServDesc: Service Description of the various capabilities covered by the standard.

NCTC: Non-Conformance Testing Clause.

Meas of Specifying Components:

I, P, and BS can be formally specified using an FDT. TDPICSP is given in tabular form. ServDesc, ConfStat, and NCTC are given in a natural language.

## III. Conformance Classes

A Standard typically has capabilities whose faithful support is mandatory for conformance or optional or conditional [2.6.7]. This generates Conformance Classes. Thus, we have: Let M, 0, and C be the sets of mandatory, optional, and conditional capabilities respectively. Let also, R: C x C. *Interdependency Relation* where ($c_i$, $c_j$) is in R iff faithful support of $c_i$ requires faithful support of $c_i$. Then, ConfClass is defined to be Set of Conformance Classes =(MXOj) U DOj where Oj $\subseteq$ O and DOj = {$c_i$ /($c_l$,$c_i$) $\in$ R and $c_l$ $\in$ Oj}.

Every conformance class corresponds to a *self-contained* set of capabilities. Each conformance class identifies a unique set of conformance requirements that has to be satisfied by an implementation to be a conforming implementation to the conformance class.

Conforming implementations are more likely to perform the required functionality as well as interoperate than non-conforming implementations.

Interoperability problems typically involve:

- receiving an illegal PDU that the implementation cannot understand; or
- receiving a legal PDU but with an illegal parameter; or
- receiving a legal PDU in an unexpected state.

These problems may result from error in designing the protocol itself or in implementing the protocol. Protocol Verification and Validation aim is to resolve errors in designing the protocols. Conformance Testing may handle errors (uncover them) in implementing the protocols.

## IV. Interoperability Requirements And Guarantees

In this section, we develop a formal method to derive interoperability requirements and guarantees. The method is based on conformance testing and the use of a Testability-Directed protocol Standard.

A conformance class ConfClassj is *interworkable* with a conformance class ConfClass;. denoted by ConfClassj INTERW ConfClassj. iff ConfClass$_i$ meets all the requirements for interworking with ConfClass$_j$; i.e.. ConfClassj may not receive, from ConfClass$_i$ a request for a service for which ConfClassj cannot guarantee delivery as per the standard. However, ConfClassj may offer more services functions/capabilities than ConfClass$_i$. It is important to note that the relation INTERW is not symmetrical as anticipated. "Conformance Testing" of an IUT determines the IUT's "Degree of Conformance (DoC)".

Here, we consider the lUT's (maximum) degree of conformance to correspond to the largest conformance class that the IUT faithfully supports. The determination of the Conformance Classes and the DoCs facilitates static investigation and study of the interoperability between the implementations of the various conformance classes: this is illustrated in the next section. This is particularly important because it provides extensive information about the potential for interoperability between the various conforming systems without having to have any physical development of any of these systems. Such information can assist manufacturers in making decisions (marketing decisions) about what standard capabilities to support, and assist the users in making decisions on what capabilities to require faithful support for in the products they intend to purchase.

Interoperability among implementations: An implementation I is said to be[1] interoperable with an implementation I', denoted by I INTERO I', iff there is not a capability, out of those offered by the standard, that I' may request from I and I cannot faithfully offer.

Lemma: For two implementations I and I': I INTERO I' iff I CONF ConfClass$_I$, I' CONF ConfClass$_i$, and ConfClass$_I$ INTERW ConfClass$_{I'}$.

Proof: Follows from having I CONF ConfClass$_i$, I' CONF ConfClass$_{I'}$ and the definition of INTERW.

## V. Example

Automated derivation of interoperability requirements and guarantees are illustrated by a Testability-Directed version of the Transport Layer Class 2 protocol given as follows.

BEGIN_TDPICS_PROFORMA_TP(2)
Identifiers:

| | |
|---|---|
| Supplier | |
| Siandard Version | 15 8073 V 1 |
| Date of Statement | |
| Implementation Identification | |
| Extra Information for Testing | |

CAPABILITIES:

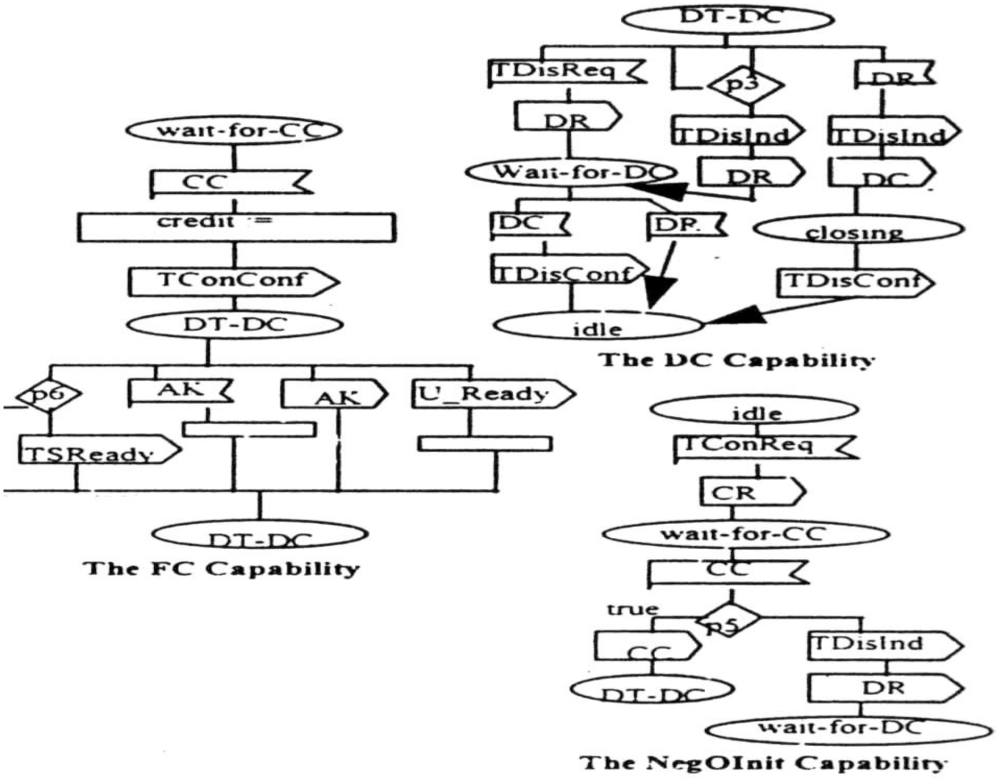
| # | Capability | Ref | Status | Su |
|---|---|---|---|---|
| A1 | Initiating the establishment of a connection | IC | $O_{IR}$.1 | |
| A2 | Responding to a request to a connection | RC | $O_{IR}$.1 | |
| A3 | Transferring Data | DT | $M_{iR}$ | |
| A4 | Disconnection of a connection | DC | $M_{iR}$ | |
| A5 | Flow Control | FC | A1:$O_{iR}$ | |
| A6 | Initiator negotiation of options | $N_I$ | A1:$O_I$ | |
| A7 | Responder negotiation of options | Nr | A2:$M_R$ | |

PDUS/SERVICEPRIMITIVES:

| £ | PDt | Ref | Status | Su |
|---|---|---|---|---|
| Bl | Output CR | CR | (al v a6): $M_I$ | |
| B2 | Input CR | CR | (a2 v a7) $M_R$ | |
| B3 | Output CC | CC | a2$M_I$ | |
| B4 | Input CC | CC | (al v a5 v a6):$M_R$ | |
| B5 | Input DR | DR | (al v a4) $M_R$ | |
| B6 | Output DR | DR | *(a2* v a4 v a6):$M_I$ | |
| B7 | TransportData | DT | a3.$M_{IR}$ | |
| B8 | Disconnection | DC | (a4 va7):$M_R$ | |
| B9 | Acknowledgement | AK | A5:$M_{IR}$ | |
| B10 | TConnectionlndic | TCI | (A2 v A7):$M_I$ | |
| B11 | TConnectionRespon | tcr | a2:$M_R$ | |
| B12 | TDisconnectionReq | tdr | (a2 v a4):$M_R$ | |
| B13 | TDisconnectionConf | tdc | A4:$M_I$ | |
| B14 | TConnectionConfin | tcc | (al v a5):$M_I$ | |
| B15 | TConnectionRequest | tcr | (al v a6):$M_R$ | |
| B16 | TDisconnectionlndic | tdi | (al v a4 va6): $M_I$ | |
| B17 | User_Ready | UR | A5:$M_R$ | |
| B18 | TSReady | tsr | A5:$M_I$ | |

PARAMETERS:

| # | Parameter | Ref | Status | Su |
|---|---|---|---|---|
| C1 | Proposed_options | B10, B15 | (A6 v A7):M | |
| C2 | Accepted_options | Bl1, Bl4 | (A6 v A7):M | |
| C3 | Options_ind | Bl, B2, B3, B4 | (A6 v A7) M | |
| C4 | Credits | Bl,B2,B3,B4, B9 | A5:M | |
| C5 | Credit_value | B17 | A5:M | |
| C6 | User_data | B2, B3, B4, B5, B7, B8, B14 | $M_{1R}$ | |



The DC Capability

The FC Capability

The NegOInit Capability

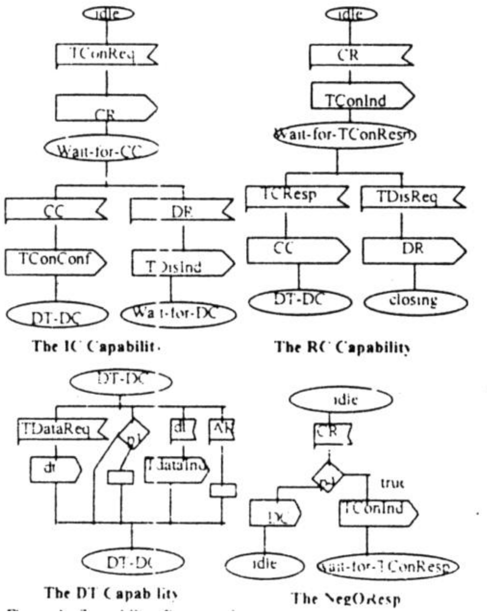


Figure 1: TD-BS of TP(2)

The Conformance Statement component of this testabiliry-directed standard is now constructed according to the guidelines in [1.6,7]. and the International Conformance Testing Standard ISO/IEC 9646.

END_Tcstability-Directed_TP_Class_2_Standard

---

Analysis with Respect to Interoperability Requirements/ Guarantees

The TDPICSP-based conformance requirements, as indicated by the TDPICSP. indicate that every implementation, to conform to the specification, has to support at least three capabilities: DT. DC. and either IC or RC; also, if the RC capability is supported, the support of the $NegO_R$ is mandatory. A conforming implementation supports up to seven capabilities. The support of the $NegO_I$ ($NecO_r$) capability requires the support of the IC (RC) capability. Consequently, there are nine conformance classes of implementations as follows (where the # field provides the conformance class identifier and the other field provides the capabilities that constitutes the conformance class):

CONFORMANCE CLASSES OF TP(2)

| # | Capabilities of the cc# | # | Capabilities of the cc# |
|---|---|---|---|
| 1 | IC,DT, and DC | 2 | IC, DT, DC, and $NegO_I$ |
| 3 | IC,DT, DC, and FC | 4 | RC, DT, DC, and$NegO_R$ |
| 5 | IC, DT, DC, Neg Oj, FC | 6 | IC,RC,DT, DC, FC,$NegO_R$ |
| 7 | IC, RC, DT, DC, Neg $O_r$ | 8 | IC,RC,DT,DC,$NegO_R$, $NegO_I$ |
| 9 | IC, RC, DT, DC, FC, Neg $O_r$, Neg Oj | | |

Every conformance class corresponds to a unique degree of conformance of implementations; consequently, there are ten implementation's degrees of conformance (IDoC) (one degree of conformance corresponding to each conforming implementation class plus a zero value (Implementation's Degree of Conformance of Zero) corresponding to implementations that fail to faithfully support any of the mandatory requirements). The static analysis of these conformance classes along with the "Implementation's Degree of Conformance" of the various implementations provides a good basis for determining the chance of interoperability between the various conforming implementations: for example. two conforming implementations with Implementations Degree of Conformance = 4 (i.e.. faithfully supports conformance class 4) cannot interwork because neither of the two implementations can initiate the establishment of a connection (each of the two acts always as a responder); while two implementations with Implementation Degree of Conformance = 9 can interwork with each other. Such type of interworking, that is capability-based, is called, here, "Capability (functional) Interworking" and can be determined by static analysis of the testability-directed standard.

On the other hand, class-9-conforning implementations (implementations that faithfuly) supports every capability, in the standard) have the highest chance to successfully interwork as long as TP(2) standard is considered, with every other conforming implementation. Generally, these class 9 conforming implementations interwork with every other faithfully conforming implementation provided that the standard is error free: such interworking covers only those layers/protocols covered by the standard.

For the testability-directed standard. We have the following interworking guarantees, illustrated in Figure 3 in a lattice form, where every node represents implementations that faithfully support a valid class and the conforming implementations of a node can interwork with every conforming implementation in its sub-tree: the type o! interworking meant here is that the conforming implementations of a node may not initiate/invoke a capability that the conforming implementations of any of its parent nodes cannot positively (according to the standard) respond to it. For example, the IC capability of a conforming implementation interworks with an RC capability of another conforming implementation: consequently, implementations conforming to class 1 interwork with the implementations conforming to class 4 but not class 1: consequently, a class -4 node (but not a class 1 node) is a parent of a class 1 node. Also, class-4-conforming implementations can intenvork with class-8-conforming implementations because the latter may not initiate the invocation of a capability that the class-4-conforming implementations cannot faithfully respond to it (i.e., there is a class 4 node that is a parent to a class 8 node).

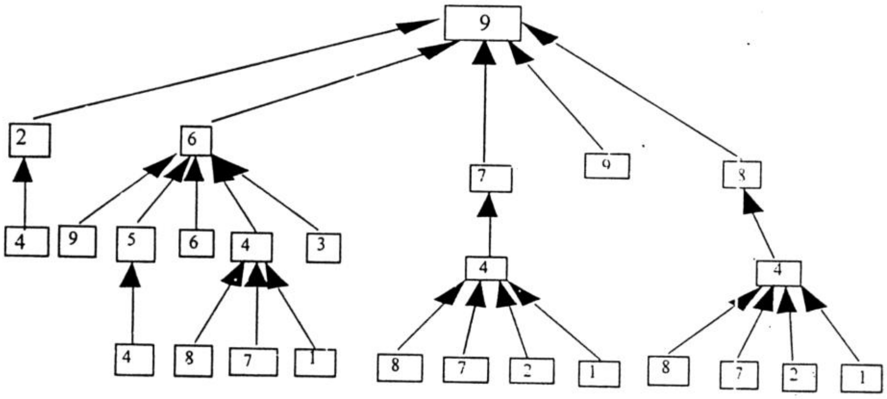


Figure 2: Lattice Representation of the interoperability guarantees between the various conforming classes.

The "interwork with" relation as defined on the conformance classes is not reflexive because, for example, the class-1-conforming implementations cannot interwork with the class-1-conforming implementations (they deadlock). Also, "intenvork with" relation is not transitive. For example, the class-1-conforming implementations interwork with class-4-conforming implementations and class-4-conforming Implementations interwork with the class-2-conforming implementations. But, class-1-conforming implementations do not interwork with class-2-conforming implementations.

However, a testability-directed standard facilitates the static analysis of the specification for interworking guarantees based on the implementation's degree of conformance which, in turn, is very useful for resolving many conformance testing issues and interoperability issues as well as assisting industry in making intelligent decisions regarding what capabilities to require the implementations to faithfully support.

## VI. CONCLUSIONS

In this paper, we have developed a CT-based technique to analyize interoperabiliy requirements and guarantees. The technique is a formal technique which makes it not error prone. Also, the technique *is* general enough to be applied to various communications protocols as far as they are formally specified using an International FDT: Lotos, Estelle, or SDL. The analysis is a static analysis which saves a lot of cost in performing interoperability testing. For n implementations of a standard, the method requires running only n conformance tests rather than n times n interoperability tests. Furthermore, the n conformance tests are needed in any way. Consequently, the method does not require conducting any additional tests and therefore saves a lot of cost. The applicability and practicality of the method has been demonstrated by a real large protocol standard. Finally, the method can be fully automated.

## REFERENCES

[1] Hazem El-Gendy, Magdi Amer, and Osman Ibrahim, “Towards Modeling of Communications Protocols“, IEEE 20th Telecommunications Forum (TELFOR), Nov. 20-22, 2012, Belgrade, Serbia, pp.95-99.

[2] Hazem El-Gindy. "Towards a Foundation for Conformance Testing Requirements: FDT-Based CTRs". Proceedings of th; IEEE International Conference on Electronics. Circuit;, and Systems. Rodous. Greece, 16-18 Sept. 1996. Wa[1] also accepted for Publication in the Proceedings of the International Conference on Networks sponsored by the nternational Association of Science and Technology for Development (IASTED). Or'ando. Florida. USA. January 8-10. 1996.

[3] Hazem El-Gendy. "A New Theory for Equivalence of Process Specifications". Proceedings of the IEEE International Conference on Electronics. Circuits, and Systems. Rodous. Greece. 16-18 Sept. 1996. 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 (IASTED). Orlando. Florida. USA. January 8-10,1996.

[4] Hazem El-Gendy, "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.

[5] Hazem El-Gendy. and Osman Abou-Rabia. "Automated Derivation of Test Sequences for Protocols and Software Specified in Lotos", Proceedings of the IEEE 2nd International Conference on Electronics, Circuits, and Systems. Amman. Jordan. December 18-21. 1995. pp. 220-225.

[6] Hazem El-Gendy and Robert Probert. "Relevant Testability Aspects of a Communications Protocol Standard". Proceedings of the 4th IEEE Maylaysia International Conference on Communications (IEEE M1CC95). Langkawi Island. Nov. 20-23, 1995. pp. 14.2.1-14.2.6.

[7] Hazem El-Gendy and Robert Probert. "Towards a Framework and Methodology for Conformance Testing Requirements". Proceedings of the 4th IEEE Maylaysia International Conference on Communications (IEEE MICC'95). Langkawi Island. Nov. 20-23. 1995, pp. 14.1.1 -14.1.8.

[8] Stephen Castro. "The Relationship Between Conformance Testing of and Interoperability- Between OS1 Systems". Computer Standards & Interfaces, Vol. 12. No. 1. 1991. pp. 3-11.

[9] Jay Gadre. Chris Rohrer, Catherine Summers, and Susan Symington, "A COS Study of OSI Interoperability". Computer Standards & Interfaces. Vol. 9. No. 3, 1989. pp. 217-237.

[10] L. Legrand and G. Cesarine, "Network Management Interoperability Testing". Preceding of the llth IFlP's International Workshop on Protocol Specification. Testing, and Verification. 1991.

[11] Pierre de Saqui-Sannes. Jean-Pierre Courtiat. and Rene Casadessus. "Verification by Abstraction as a Preamble for Interoperability Test Suite Generation". Proceedings of the 14th IFIP WG6.1 International Symposium on Protocol Specification. Testing and Verification. Vancouver. BC. Canada. June 1994. pp. 152-166.

[12] Mohamed Wagdi Youssef and Hazem El-Gendy, “Scrampling and Encrypting-Based Authontication for for Open Networks Communications”, International Journal of Computer Science and Network Security, June 2011, pp. 24-29.

[13] Hazem El-Gendy, Nabil El-Zant El-Kadhi, Osman Ibrahim, and Narayan Debnath, “Complete Management Scheme for Intelligent Terminals for Information Super-highways and Its Design”, International Journal of Computational Methods in Sciences and Engineering, Volume 10, No.s 1,2, 2010, pp. 247-257.

[14] Nabil El-Kadhi and Hazem El-Gendy, "An Intelligent Bidirectional Authentication Method", The International Journal on Computer and Network Security, published in Vienna, Austria, 2010, pp.64-70.

[15] Hazem El-Gendy, Nahla El Zant El Kadhi, Osman Ibrahim, and Narayan Debnath, "Complete Management Scheme for Intelligent Terminals for the Information Super-highways and its design", Journal of Computational Methods in Sciences and Engineering, Vol. 10, Number 1,2, 2010, IOS Press, Netherland.

[16] Hazem El_Gendy, "Towards ISO Interface Protocol for Interactive Multimedia Intelligent Terminals", Proceedings of the 9th International Conference on Computer Systems and Applications, Hammamet Tunisia, May 16-19, 2010 sponsored by ACS, IEEE, and IEEE Computer Society.

[17] Hazem El-Gendy, Osman Ibrahim, Nabil El Kadhi, and Narayan Debnath, "Management Scheme for Intelligent Terminals for the Information Super-highways", Proceedings of the 9th International Conference on Computer Systems and Applications, Hammamet Tunisia, May 16-19, 2010 sponsored by ACS, IEEE, and IEEE Computer Society.

[18] Hazem El-Gendy, "Formal Development of Universal Protocol Implementations Conformance Statement Proforma", Proceedings of the 7th ACS/IEEE International Conference on Computer Systems and Applications sponsored by ACS, IEEE, and IEEE Computer Society, Rabat, Morocco, May 10-13, 2009.

[19] Hazem El-Gendy, Mohamed Gebriel, Ahmed Samir, Narrayan Debnath, and Nabil El-Kadhi, "Towards Mosques Management Information System", Accepted for Publications in the 7th ACS/IEEE International Conference on Computer Systems and Applications sponsored by ACS, IEEE, and IEEE Computer Society, Rabat, Morocco, May 10-13, 2009.

[20] Hazem El-Gendy, Nabil ElKadhi, and Narrayan Debnath, "Towards Sound Development of PIXITP, Conformance Test Suites, and Conforming Implementations For Various Formal Description Techniques", Proceedings of the 12th IEEE International Symposium on Computers & Communications sponsored by both IEEE Computer Society and IEEE Communications Society, Morocco, July 6-9, 2008.

[21] Hazem El-Gendy, Nabil ElKadhi, and Narrayan Debnath, "Formal Automated Transformation of SDL Specifications to Lotos Specifications", Proceedings of the 12th IEEE International Symposium on Computers & Communications sponsored by both IEEE Computer Society and IEEE Communications Society, Morocco, July 6-9, 2008.

[22] Hazem El-Gendy, Nabil ElKadhi, and Narayan Debnath, "Formal Automated Transformation of SDL Specifications to Estelle Specifications", Proceedings of the 23rd International Conference on Computers And Their Applications (CATA'08) sponsored by the International Society for Computers & their Applications (ISCA), April 8-10, 2008, Cancun, Mexico.

[23] 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.

[24] Nabil El-Kadhi and Hazem El-Gendy, "Advanced Method for Cryptographic Protocol Verification", Journal of Computational Methods in Sciences and Engineering, Volume 6, Numbers 5, 6, 2006, pp. 109-119.