A Fuzzy Model for Test Results Certification

M. Bohus
Department of Computer Science
József Attila University
Árpád, Szeged, Hungary
bohus@inf.u-szeged.hu

L. T. Kóczy
Department of Telecommunications and Telematics
Technical University
Stocek, Budapest, Hungary
koczy@ttt-202.ttt.bme.hu

K. Tarnay
Central Research Institute for Physics
Hungarian Academy of Sciences
Konkoly Thege, Budapest, Hungary
tarnay@sunserv.kfki.hu

Abstract

Certification as an exhaustive type analysis is a long and time consuming process. One of the main activities of certification is the conformance testing. Some methods are used to shorten and simplify it, but in this case the validity of the test results is uncertain. We develop a fuzzy model (CERtification MODel, CERMOD) to decrease this uncertainty. The relational model induces a fuzzy-optimal test selection. The evaluation of the test suites, the description of the testing goals and constraints make possible the optimal decision and serve as the elements of the adaptive and intelligent control system. Finally, the coverage of the simplified testing according to the validity of test results is analyzed. Our results are demonstrated on the SS7 MTP-2 tests.

1 Background

We introduce the background information and motivation of our work. We should like *to support* the certification procedure. The main activity of the certification is the mostly non-exhaustive conformance testing. Testing drivers give the control of the testing. Nowadays, the connection between academic and industrial testing is getting increasingly actual and a lot of people agree with the assessment both sides. Academic approach has to prepare more appropriate formal models attending the critical areas of the testing, the optimal efforts and the reasonable amount of resources. Industrial testers are forced to use formal specifications in the design. It is desirable to define the goals, the constraints and the completion criteria of the testing. To support these activities, the pure analytical tools are not applicable (different space problem). In the simplification and the abstraction of some testing phase it seems the *fuzzy theory* makes the management of the validity of the test results easier. In this paper we introduce our model for supporting the certification (CERMOD). It is based on fuzzy theory. The first steps of proving of CERMOD model were made on MTP-2 protocol.

1.1 Certification

The *certification* is one part of the life-cycle of the development of protocols [9]. The other elements are the recommendations, the standards, the test sets, the specification, the implementation. One of the main activities of the certification is the conformance testing. Certification has many steps, checking the documents, choosing the certification method, selection of test suites and parameters, conformance testing, interoperability testing and preparing the result documents of the certification. We investigated this process, analyzed the earlier results and give some hints for formalization and optimization of certification.

1.2 Conformance testing

It is well known that the general problem of testing communication implementations is insoluble. The number of states of an implementation of a significant protocol is so large that no testing strategy will result in an exhaustive test. The general testing problem is therefore partitioned into a number of areas, each of which addresses a particular aspect of the general problem. Much of the current literature on communication protocol testing focuses on two aspects of the general testing problem: conformance testing methodology and the algorithms for generating test sequences [5, 11]. There are many other aspects of the testing problems. The terms robustness, reliability, interoperability, interworking, performance, connectivity and stress are all commonly used to qualify the word "testing". Each defines a field of protocol testing that is incompletely understood and deserves more attention.

Conformance testing is a formal procedure whose aim is to demonstrate that the external behavior of an implementation of a protocol corresponds to that defined by the specification of the protocol. A comprehensive set of test cases is defined, each having a particular purpose, the testing of a specific protocol function. The implementation under test (UIT) is considered to comply with the protocol specification if its responses to the stimuli defined in the test cases correspond to the behavior defined by the specification from which the test cases are derived. The quality of test is dependent on the content of the executed test cases. Ideally, test case should be automatically derived from a precise, formal specification of the protocol and defined in a general form that can be used in many environments. Test cases that are not generated automatically are limited by the inventiveness of the test designer. It is not possible to test what the test designer has not defined. Developing a set of test cases is a challenging and time-consuming task for an expert.

In paper [13], the experience with a methodology for testing implementations of communication protocols that uses autonomous test drivers to exercise the protocol service interface is described. The test drivers only generate random event sequences that the service interface specifies should be accepted by a correct implementation. Event rejection therefore indicates a discrepancy between the service specification and the implementation under test. The methodology has proved to be a powerful and cost-effective complement to other testing techniques.

1.3 Assessment between academic and industrial testing

Academic research has recently made significant adventures in the generation of test sequences from formal specifications and in the development of computer-aided test tools, with the aims of improving the effectiveness of testing [8]. The industry, with its own method of testing, is shipping to the customer communication software that still has bugs, even after it has been tested in the factory. It could therefore be expected that academic testing methods would be used to help to improve its effectiveness. Surprisingly, these methods are seldom used in the industry. This paper describes the gap between the two classes of methods, and the ways in which they can converge. It is intended to open a discussion on these rarely addressed topics. An effective academic testing method applicable to the industry must deal with the problems encountered in industrial testing. Academic research on protocol testing needs to be more conscious of these problems. At the same time, the industry needs to move towards using formal methods in order to improve the efficiency and effectiveness of testing. It is therefore concluded that neither the academic nor the industrial methods provide the most proper means of testing, but rather that there should be a combination of these two: the judgment and the experience of the testers, together with formal methods, that can address the practical problems of industrial testing. It is essential for academia to keep a closer watch at the real issues facing the industry. Both academia and the industry need to move towards each other in order for an improvement in real-life testing to be achieved. This remains a challenging area of research for both academia and communication specialists in the industry. There is no reason to believe that *academic formal methods* will not flourish in the industry in the future.

1.4 Fuzzy theory

In CERMOD model the fuzzy theory gives the main part of the theoretical foundation. After more than two decades from the foundation of *fuzzy theory*, nowadays there are more and more new applications of this theory [7]. It covers many topics, namely fuzzy sets and relations, fuzzy logic and reasoning, fuzzy control, adaptive fuzzy control, and fuzzy logic in knowledge-based systems.

1.5 Analyzing test results of certification(SS7, MTP-2)

In the preparation phase we *analyze* the results of the certification of Signaling System #7 (SS7) [10] performed by test labs and pay more attention to details of Message Transfer Part level 2 protocol (MTP-2). The general experiences were that the planning is not enough supported by computer, the testing is pre-defined without adaptive and intelligent control, which means the missing of history of previous steps, and the control is manual [2].

2 Supporting the certification

We develop *a certification model* (CERMOD) for test result certification. The aim of this approach is to prepare the integrated support for all range of certification activity. In this article the error detection of conformance testing is more elaborated using fuzzy sets and fuzzy decisions.

2.1 Weak points of certification

We have found some weak points of certification. Without using formal methods and models, one can hardly believe that the specification of more difficult protocols is well-defined. It is useful to describe the *goals and constraints* of the certification in an explicit way. Every testing process has the goals and constraints described informally. It may be inconvenient to describe them formally. It is evident that the evaluation and the *classification of the test case*, the test suites and the test list is necessary for the decisions of the testing control and in the measuring of the validity of the certification, which is also missing. Control is mostly static, manual, nonadaptive and nonintelligent. If we can solve the representation of the knowledge of the testers, we will take a great jump in the automation, too.

2.2 Elements of CERMOD model

The most important parts of the model:

- Evaluation of the test suite.
- Description of goals and constraints.
- Optimal selection of the next test suite.
- Memorizing of test history.
- Intelligent control of testing.
- Analyzing of test results.

2.3 Parts of fuzzy theory

2.3.1 Fuzzy set

Informally, a *fuzzy set* is a class of objects in which there is no sharp boundary between those objects that belong to the class and those that do not. A more precise definition may be stated as follows.

Definition. Let $X = \{x\}$ denote a collection of objects (points) denoted generically by x. Then *fuzzy set A in X* is a set of ordered pairs

$$A = \{(x, \mu_A(x))\}, x \in X$$

where μ_A is termed the *grade of membership* of x in A, and $\mu_A : X \to M$ is a function from X to a space M called the membership space.

When M contains only two points, 0 and 1, A is nonfuzzy and its membership function becomes identical to the characteristic function of a nonfuzzy set. In what follows, we shall assume that M is the interval [0, 1], with 0 and 1 representing, respectively, the lowest and the highest grade of membership. (More generally, M can be a partially ordered set or, more particularly, a lattice.) Thus, our basic assumption is that a fuzzy set A — despite the unsharpness of its boundaries — can be defined *precisely* by associating with each object x a number between 0 and 1 which represents its grade of membership in A.

We turn next to the definition of several basic concepts which we shall need in later sections.

Definition. The *intersection* of fuzzy sets A and B in X is denoted by $A \cap B$ and is defined as the largest fuzzy set contained in both A and B. The membership function of $A \cap B$ is given by

$$\mu_{A \cap B}(x) = Min(\mu_A(x), \mu_B(x)), \; x \in X$$

where $Min(a, b) = a$ if $a \leq b$ and $Min(a, b) = b$ if $a > b$. In infix form, using the conjunction symbol $\wedge$ in place of Min, the previously defined formula can be written more simply as

$$\mu_{A \cap B}(x) = \mu_A(x) \wedge \mu_B(x).$$

The notation of intersection bears a close relation to the notation of the connective "and". Thus, if A is the class of tall men and B is the class of fat men, then $A \cap B$ is the class of men who are both tall *and* fat.

2.3.2 Fuzzy logic

Fuzzy logic is basically a multivalued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false, black/white, etc. Notions like *rather warm* or *pretty cold* can be formulated mathematically and processed with the computer. In this way an attempt is made to apply a more human-like way of thinking in the programming of computers.

2.3.3 Fuzzy decision

By decision-making in a fuzzy environment it is meant a decision process in which the goals and/or the constraints, but not necessarily the system under control, are fuzzy in nature [1]. This means that the goals and/or the constraints constitute classes of alternatives whose boundaries are not sharply defined. An example of a fuzzy constraint is: "The cost of A should not be *substantially* higher than a," where a is a specified constant. Similarly, an example of a fuzzy goal is: x should be in the *vicinity* of x_0, where x_0 is a constant. The italicized words are the sources of fuzziness in these examples. Fuzzy goals and fuzzy constraints can be defined precisely as fuzzy sets in the space of alternatives. A fuzzy decision, then, may be viewed as an intersection of the given goals and constraints. A maximizing decision is defined as a point in the space of alternatives at which the membership function of a fuzzy decision attains its maximum value. The use of these concepts is illustrated by examples involving multistage decision processes in which the system under control is either deterministic or stochastic.

In the conventional approach to decision-making, the principal ingredients of a decision process are (a) a set of alternatives; (b) a set of constraints on the choice between different alternatives; and (c) a performance function which associates with each alternative the gain (or loss) resulting from the choice of that alternative. When we view a decision process from the broader perspective of decision-making in a fuzzy environment, a different and perhaps more natural conceptual framework suggests itself. The most important feature of this framework is its symmetry with respect to goals and constraints — a symmetry which erases the differences between them and makes it possible to relate in a relatively simple way the concept of a decision to those of the goals and constraints of a decision process.

Definition. Assume that we are given a fuzzy goal G and a fuzzy constraint C in a space of alternatives X. Then, G and C combine to form a *decision*, D, which is a fuzzy set resulting from intersection of G and C. In symbols,

$$D = G \cap C$$

and correspondingly $\mu_D = \mu_G \wedge \mu_C$. The relation between G, C and D is a common region in the two-dimensional pictures of them.

Definition. More generally, suppose that we have n goals $G_1, ..., G_n$ and m constraints $C_1, ..., C_m$. Then, the resultant decision is the intersection of the given goals $G_1, ..., G_n$ and the given constraints $C_1, ..., C_m$. That is,

$$D = G_1 \cap ... \cap G_n \cap C_1 \cap ... \cap C_m$$

and correspondingly

$$\mu_D = \mu_{G1} \wedge ... \wedge \mu_{Cm}.$$

Note that in the above definition of a decision, the goals and the constraints enter into the expression for D in exactly the same way. This is the basis for our earlier statement concerning the identity of the roles of goals and constraints in our formulation of decision processes in a fuzzy environment.

Comment. The definition of a decision and the intersection of the goals and constraints reflects our interpretation of "and" in the "hard" sense. If the interpretation of "and" is left open, we shall say that a decision — viewed as a fuzzy set — is a *confluence* of the goals and the constraints. Thus, "confluence" acquires the meaning of "intersection" when "and" is interpreted in the sense of $\mu_{A\cap B}(x) = \mu_A(x) \wedge \mu_B(x)$, the meaning of "algebraic product" when "and" is interpreted in the sense of $\mu_{AB}(x) = \mu_A(x)\mu_B(x)$, and may be assigned other concrete meanings when a need for a special interpretation of "and" arises. In short, a broad definition of the concept of *decision* may be stated as:

Decision = Confluence of Goals and Constraints.

2.3.4 Fuzzy control

Fuzzy controllers are the most important application of fuzzy theory. They work rather different than conventional controllers; expert knowledge can be expressed in a very natural way using *linguistic variables*, which are described by fuzzy sets.

The employment of fuzzy control is commendable..

- ...for very complex processes, when there is no simple mathematical model,
- ...for highly nonlinear processes,
- ...if the processing of (linguistically formulated) expert knowledge is to be performed.

The employment of fuzzy control is not so good idea if...

- ...conventional control theory yields a satisfying result,
- ...an easily solvable and adequate mathematical model already exists,
- ...the problem is not solvable.

2.4 Evaluation of test suites /fuzzy sets/

The evaluation of the test suites is a basic element of this model. The nature of this grouping has some fuzzy features (grade of importance). Many points of view can be defined. A *good behavior* grouping emphasizes the importance of the minimum good behavior in every communication phase in the test suite. If we test the *timing effects*, those test suites have more importance, that contain timing events. A *crucial behavior* gives more importance to the bad situations in the test suites. The number of the observable errors also may play a valuable role if the goal is to find as many errors as possible. The investigation of the error reasons is a great challenge, but this is the real aim of the testing and naturally correction of the bugs.

2.5 Test suite selection

This function of the model can be a part of control of testing. After defying the goals and constraints we can optimize the selection of the next test suite. It is worth to build an adaptive, intelligent test controller (but may be more expensive solution), which has the memory for the history of the previous testing steps, the knowledge base with the experiences of the testers built in the inference machine and the optimizer/terminator unit. This last unit also may manage the results of the certification. In many cases the definition of the control rules has the fuzzy features and we can use the experiences of the *fuzzy control*.

3 Example

To demonstrate the operability of our approach, we chose only one middle-size real protocol. It was MTP-2 data link protocol in SS7. This protocol is a member of the well-known, effective and well-used data link protocol family. Another reason for its selection is the actuality of SS7, serving as a basement in developing the intelligent networks (IN). Very good motivation was the analysis of the earlier test results made by test labs. In the cooperation with these labs we can prove our results.

3.1 Signaling System #7

The international telephone network was facing the problems of providing new services such as data and video. In many countries, just getting telephone service was a feat in itself. As international bodies began investigating alternative technologies for providing telephone service to the masses (such as cellular), the need for an all-digital network became apparent. Thus, the beginnings of an all-digital network with intelligence arose. The International Telecommunications Union (ITU) commissioned then CCITT to study the possibility of an all-digital intelligent network. The result was a series of standards known now as Signaling System #7 (SS7). These standards have paved the way for the Intelligent Network (IN) and, with it, a variety of services, many yet to be unveiled.

3.1.1 Overview of the SS7 Protocol Stack

The SS7 architecture differs somewhat from the OSI model. While the OSI model consists of seven different layers, the SS7 standard uses only four levels. The term “level” is used in the same context as “layer”. The functions carried out by these four levels correspond with the OSI model's seven layers. Some of the functions called for in the OSI model have no purpose in the SS7 network and are, therefore, undefined. It should also be noted that the functions in SS7 have been refined over the years and tailored for the specific requirements of the SS7 network. For this reason, there are many discrepancies between the two groups of protocols and their corresponding functions. Regardless of the differences, SS7 has proven to define a highly reliable packet-switching protocol, providing all of the services and functions required by the telephone service providers. This protocol continues to evolve as the network grows and the services provided by the telephone companies change. The levels are: level 1 - physical level, level 2 - data link level, level 3 - network level, 1-3 together - message transfer part, level 4 - user parts. Level 4 in the SS7 network consists of several different protocols called user parts and application parts. For basic telephone call connection and disconnect, the Telephone User Part (TUP) or ISDN User Part (ISUP) protocols are used. TUP is used in Europe and other companies following ITU-TS standards, while ISUP is used primarily in North America. To access network databases, the Transaction Capabilities Application Part (TCAP) protocol is used. TCAP supports the functions required to connect to an external database, perform a query of the database, and retrieve information. The information or data retrieved is then sent back in the form of a TCAP message to the signaling point that requested it.

3.1.2 Message Transfer Part (MTP)

The Message Transfer Part (MTP) acts as the carrier for all SS7 messages, providing reliable transfer of messages from one signaling point to another. This function includes level 1, 2, and 3. In addition to providing signaling point communications, the MTP also provides error detection and correction. The methodologies of MTP are very similar to those used in other bit-oriented protocols (BOPs), such as X.25. Sequence-numbering and error-checking mechanisms are very similar. The signal unit structure used in all SS7 messages provides all the information required by MTP level 2 and level 3. Flow control is provided through the use of a special signal unit called the Link Status Signal Unit (LSSU).

3.1.3 MTP-2

The backward sequence number and the forward sequence number are used for sequencing packets and are used by level 2 to ensure that all transmitted packets are received. They are also used for positive and negative acknowledgments. The indicator bits are used to request a retransmission. The length indicator allows level 2 to determine the type of signal unit being sent, and the cyclic redundancy check

(CRC) field is used to detect data errors in the signal unit.

3.2 MTP-2 Test Specification

This recommendation [6] contains a set of tests of signaling system No. 7 MTP level 2 protocol. These tests intend to validate the protocol specified in Recommendation Q.703. This recommendation conforms to Recommendation Q.780 which describes the basic rules of Test Specification. The level 2 tests aim at testing the level 2 protocol conformance in a given implementation. These tests as a whole aim at a complete validation of the level 2 protocol without redundancies. Each test is described as simply as possible to check precisely each elementary function of the protocol, which is referred in the columns "reference", "title" and "sub-title" of each test description. This list is presented in the form of a succession of tests. The presentation order is essentially functional. However, the operator performing these tests may change this order, taking into account some other practical criteria such as: use pre-test conditions to order the list, the end of a given test may be the pre-test condition of another test.

3.3 Supporting of selection

CERMOD relational model is composed of four sets:
- tests for error detection,
- observable errors,
- goals,
- constraints.

Fuzzy membership values are associated with the elements of the sets, expressing the grade of the belonging to the classes defined for grouping.

The test list of MTP. level 2 test specification has *96 test suites*. The original grouping follows the testing of the communication functions: link state control with expected units/orders, link state control with unexpected signal units/orders, transmission failure, processor outage control, SU delimitation, alignment error detection and correction, SUERM check, AERM check, transmission and reception control, and congestion control. These 96 test suites contain *248 observable errors*, from these 200 are different. *8 test suites are related with timers.*

3.3.1 Evaluation

We prepared the fuzzy sets according to the importance of the good behavior (minimal amount of functional testing, testing not continued, when basic functions work), the timer effects (the timing is the main function of the operability), the number of the observable errors in the test suites.

Very simple timer fuzzy set is given in Table 1. There are 4 timers and 2 groups.

Table 1: Importance of timers

	T2	*T1 and T4*	*T1 in A state*
very important	0.8	1	0.2
less important	0.5	0.1	1

3.3.2 Selection

Supporting the selection of the next test suite we described the goals and constraints of the testing. If they are very simple (it means they can derive from previously mentioned evaluation tables in a simple way), than the new fuzzy sets for the evaluation of the goals and constraints are not necessary. Two simple statements are:
Goal: very important timers should be tested.
Constraint: only those with the membership value larger than 0.7 should be tested.

If they are more difficult, we prepare more evaluation tables and use the intersection function (see section 2.3.3). For emphasizing some viewpoints we can introduce the weight system to give the weights to each evaluation. With these values we can select the next test suite (the next member of the fuzzy-optimal subset). This is done automatically if we have adaptive control.

4 Summary and conclusions

It easily follows from previous sections that the support of the certification is possible and more intelligent control is planned from the results of this approach. *The fuzzy logic is an applicable theory in the supporting of the certification process.* The adequate representation of the knowledge of the test experts in the evaluation of the test suites, the description of the testing goals and constraints is crucial for the optimal decision.

From this early stage of this research and from the complexity of the certification several interesting research topics for next future derive. Some of them are: the evaluation of the testing parameters, the analysis of the correlation between the error reasons and the observable errors, the refinement of the process control, the integration of the successfully supported fields of the certification and the aim for supporting the certification of the whole system (stacks of protocols).

Acknowledgements

The authors are grateful to the audience of the university seminars for their very active attention, to the colleagues of the test labs for sharing of their experiences derived from their certification work and to the editors for the review.

References

[1] R. E. Bellman, L. A. Zadeh, "Decision-Making in a Fuzzy Environment,", Selected Paper, *Fuzzy Sets and Applications,* Edited R. R. Yager & al., John Wiley & Sons, 1987.

[2] M. Bohus, "Analyzing of Protocol Conformance Test Results for Certification," *Proc. of PhD. Conference, TTT, TU-Budapest*, Vol. 2, pp. 23–24, 1996.

[3] A. Cavalli, "Test generation methods based on I/O FSMS," *EUNICE'95 Summer School on Protocol Engineering.*

[4] M. Hajnal, L. T. Kóczy, "Classification of Textures by Vectorial Fuzzy Sets,", *Approximate Reasoning in Decision Analysis,* Edited M. M. Gupta and E. Sanchez, North-Holland Publishing Company, 1982.

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

[6] ITU-TSS, *MTP Level 2 Test Specification,* ITU-TSS, Q-781, 1994.

[7] G. J. Klir, T. A. Folger, *Fuzzy Sets, Uncertainty and Information,* Prentice-Hall International, 1988.

[8] R. Lai, W. Leung, "Industrial and academic protocol testing: the gap and the means of convergence," *Computer Networks and ISDN Systems,* Vol. 27, pp. 537-547, 1995.

[9] J. Miskolczi, "Protokollok alkalmasság vizsgálata," *Hiradástechnika,* Vol. 46, pp. 24–30, 1995.

[10] T. Russel, *Signaling System #7,* Mc Graw-Hill, Inc., 1995.

[11] K. Tarnay, *Protocol Specification and Testing,* Akadémiai Kiadó, Budapest, 1991.

[12] M. Törő, T. M. Dung, "Specification of abstract test suites in MSC," *SDL forum,* 1995.

[13] C. H. West, A. Tosi, "Experiences with a random test driver," *Computer Networks and ISDN Systems,* Vol. 27, pp. 1163–1174, 1995.