# AUTOMATIC TEST CASE GENERATION OF REAL PROTOCOLS: FRAMEWORK AND METHODOLOGY

[1]Tae-hyong Kim, Ik-soon Hwang, Chan-min Park,
[2]Jai-yong Lee, Sang-bae Lee

Dept. of Electronic Eng., Yonsei Univ.
Seodaemun-gu, Shinchon-dong 134, Seoul, 120-749, Korea
Phone: +82-2-361-2864, Fax: +82-2-312-4584
{[1]thkim,[2]jyl}@nasla.yonsei.ac.kr

**Abstract:** A lot of researches have been made for an effective and efficient test case generation method. However, theoretical researchers and practical researchers follow the different goals and methods each other especially in the test case generation of real protocols. In this paper, we proposed a new framework and methodology for automatic test case generation of real protocols. They bridge between the theoretical and practical researches and try to bring together the best of theory and practice in test case generation. They can cope with the change of test environment and technical state. The key idea is the distributed analysis technique. We used both simulation and model analysis according to the context dependency of the sequence. We defined test coverage selection criteria of a protocol in an EFSM(Extended Finite State Machine) model and in an EEFSM(Expanded EFSM) model. We used three techniques to reduce the complexity of simulation: the depth first search with termination conditions, the partial supplementation technique and the partitioning technique. To show the efficacy of the proposed framework and methodology, we apply them to a real communication protocol, SSCOP(Service Specific Connection Oriented Protocol) for B-ISDN protocol.

## 1 INTRODUCTION

Since whether a communication protocol implementation conforms to its standard specification or not is required to be tested, a great number of researches have been made in this area. A lot of studies among them are interested in test

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35394-4_29

S. Budkowski et al. (eds.), *Formal Description Techniques and Protocol Specification, Testing and Verification*

case generation method for conformance test. To generate more effective and efficient test cases, they try to generate test cases with higher fault coverage and for data flows as well as control flows in protocols. But there are many problems in those studies. One of them is to overcome the difficulties due to the complex structure and many functions of communication protocols. For the purpose, many efforts to reduce the problem size have been made such as modeling or partitioning of a protocol into a simplified structure, test selection and optimization. There is a trade-off relation between test coverage and the cost of test. Theoretical researchers, however, are more interested in the former issue. It is generally known that much time is required for the generation of test cases as well as the testing itself. Automatic test case generation is, therefore, one of the most important issues, because they will save much time.

Several automatic or semi-automatic test case generation tools such as TGV, TVEDA, TestGen and SAMsTAG have been developed and now they are under improvement(Doldi *et al.*, 1996; Grabowsky *et al.*, 1997). They try to reduce manual efforts in the course of test by the power of FDT(Formal Description Techniques) and their related tools. They also naturally focus on generation of test suite in TTCN(Tree and Tabular Combined Notation) that is directly applicable to a real test. Most of them use the simulation technique to find out the sequences corresponding to the selected test purposes. The simulation technique can also give the values to complete the constraints of the test suites. Their developers, however, do not try to apply techniques of theoretical researches to them yet such as techniques for extending fault coverage of test cases. Theoretical researchers also try to develop automatic tools to generate test cases, but their tools are mostly for validation of their researches and for simple protocol with many assumptions. So it is nearly impossible to use them directly to generate test cases for real protocols.

Like the most cases of other areas, in general, there is also a considerable gap between the theoretical approaches and the practical approaches in the area of conformance test. In industrial context, the most important goal is whether it can be applied to implementations and almost all the tests have real constraints such as deadline and the cost. In academic context, on the other hand, most of tests are free of constraints by some assumptions and the effectiveness and validity are the matter of primary concern. Their viewpoints to generation of test cases for real communication protocols show this gap well. Theoretical researchers will not use a real communication protocol as a target protocol, because they are so much complicated and difficult to model or simplify properly. Practical researchers seldom use the theoretical methods in testing real protocols because they think the methods are impractical and irrelevant for real protocols. Therefore, new test case generation framework to be able to use both the advantages of two approaches is required and we proposed a new framework for automatic test case generation and its methodology in this paper.

The paper proceeds as follows: In section 2, we show the proposed test case generation framework. The next 3 sections explain the methodology of test case generation in the framework: test coverage and test selection criteria are explained in section 3, the distributed analysis method in section 4 and reduction techniques of problem size in section 5. Section 6 shows the empirical results in applying the proposed method to B-ISDN protocol SSCOP(Ser-vice Specific Connection Oriented Protocol). We evaluate the method by comparing with related studies in Section 7. Finally conclusions and future works are given in section 8.

## 2 OVERALL FRAMEWORK

The characteristics of the required framework are as follows. They should be able to accept the requirements and viewpoints of the theoretical and practical approaches. They should be also able to use the advantages of the two approaches and deal with newly developed theoretical techniques, various test environment and new complex protocols. Considering the above requirements, we proposed a new framework as shown in Figure 1.

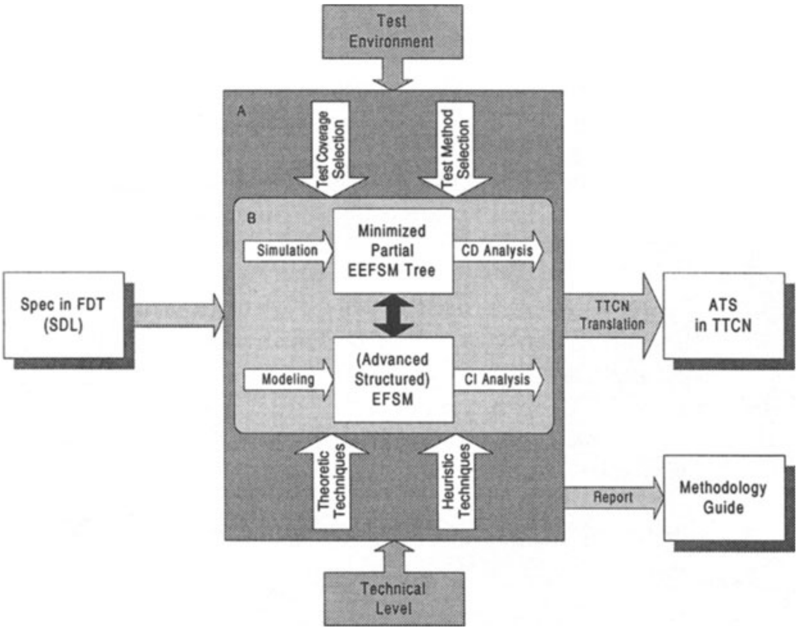


**Figure 1** The proposed test case generation framework

The target protocol specification is given in FDT such as SDL. The test case generation procedure is affected by two external factors: the test environment and the technical level. The test environment factor is for several constraints of practical testing and the technical level factor is for new results of theoretical or practical researches. These two factors decide variable elements of this framework. Test constraints like cost or deadline may decide the test coverage of the test case generation method. Physical conditions of test system may select the test method. Where the PCOs(Points of Control and Observations) of the given IUT(Implementation Under Test) or SUT(Sys-tem Under Test) exist will select the specific test method. By the selected test coverage and the test method, the test purposes of the test suite to be generated are decided. On the other hand, newly developed techniques of test case generation may solve the problems existed in the existing method. Theoretical studies or a lot of test experiences can develop the techniques. The test case generation method is developed by those techniques. Thus, this rather dark region 'A' containing the above two methods can allow manual decisions or manipulations, if needed by external constraints. The test coverage selection is explained in detail in section 3 and among the techniques those for reduction of problem size are explained in section 5.

The core of the proposed framework is the distributed analysis method depicted as the region 'B' in Figure 1. In general, while the practical tools to generate test suite in TTCN use simulation to find test cases reflecting test purposes, theoretical researchers try to derive test cases by modeling the specification and its analysis. The simulation technique may take long time and the model analysis is incomplete now. Real communication protocols have several functions like a buffer management and a timer management that are difficult to model easily. The model analysis for all the protocol functions is incomplete and impractical but the analysis will be able to help to reduce the time of simulation, we think. So we used two techniques partially. The parts that are difficult for the model analysis are context dependent parts having predicate statements. Therefore the model analysis is in charge of context independent parts and the simulation technique is of context dependent parts. Section 4 explains these techniques in detail.

By the distributed analysis technique, the values of constraints in the test suite are decided and those results of the analyses are translated into an abstract test suite in TTCN. The selected test method, the techniques and other information for the generation process are reported as a methodology guide. This information is used for a new heuristic technique.

## 3 TEST COVERAGE AND TEST SELECTION CRITERIA

Lately as the protocol model used for test case generation of conformance test EFSM(Extended Finite State Machine) model is used in many cases, because the FSM(Finite State Machine) or LTS(Labeled Transition System) model cannot detect any error of data flows, and moreover may be impractical because of not considering executability of a test sequence. Therefore, test cases that can find the faults in data flows as well as control flows are required. In the data flow test of a protocol, Weyuker's data selection criteria based on data flow analysis of software testing are widely used as fault coverage selection criteria(Weyuker and Rapps, 1985). But her software model is different from the general EFSM model of a communication protocol; data elements like *def* or *use* are usually considered to be only in edges in an EFSM model of a protocol.

Modification of the criteria is, therefore, required. The modified the data selection criteria for protocols with the test coverage criteria of control flow test are shown in Figure 2.

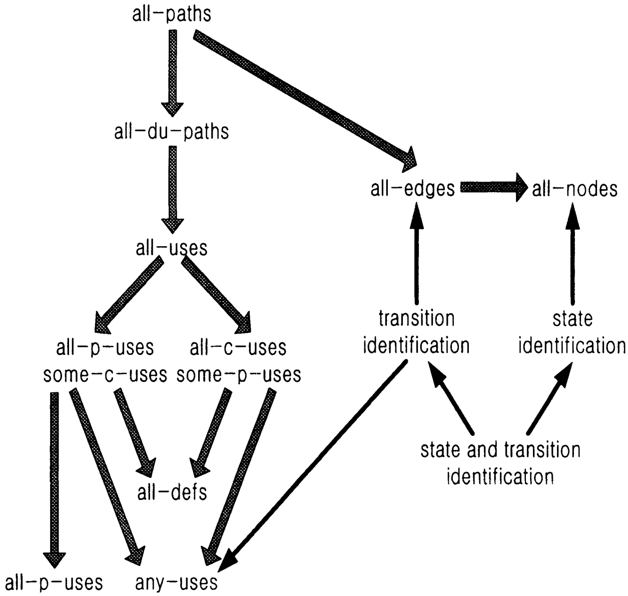


**Figure 2** The test data selection criteria of EFSM

The major difference from the Weyuker's is that the test cases satisfying *all-p-uses* or *all-uses* criterion cannot cover the test cases satisfying *all-edges* criterion. There can exist edges without data element such as def or use in an EFSM model of a protocol when we do not consider major states of FSM a kind of variables as usual cases in protocol testing. All the test cases satisfying *all-uses* criterion, therfore, do not have all the edges. We added one criterion in the relations, *any-uses* criterion. The definitions of those criteria are as follows.

**Definition 1** The test suite satisfies **any-uses** criterion iff for all *uses* of all the variable $x$ whose edge number is $m$, there is at least one *def-clear-path* with respect to $x$ from $n$ to $m$, where $n$ is the edge number of the *def* of the *def-clear-path.*

*Any-uses* criterion can be used in some practical environments. The test coverage selection criteria of a control flow test depicted together show the relations with data selection criteria. Naturally transition identification criterion covers *all-edges* criterion and *state identification* criterion covers *all-nodes* criterion. Test cases satisfying *transition identification* criterion come to have all the *uses* of the variables and their prior *defs* in natural, in other words they have at least one *def-clear paths* for each *use*. Therefore, *transition identification* criterion covers *any-uses* criterion.

As we mentioned in section 2, we use simulation for context dependent analysis. By simulation, the specification is transformed to a reachabiliy tree. The tree has normally many *homogeneous states* that were originally same states. To reduce the problem size, we use minimized EEFSM(Expanded EFSM) tree instead of the reachability graph as shown in Figure 1(Henniger *et al.*, 1995). The minimized EEFSM tree is generated by merging the homogeneous states having same data values in the reachability tree. In the minimized EFSM tree, the relations between test data selection criteria and test coverage selection criteria are changed as shown in Figure 3.

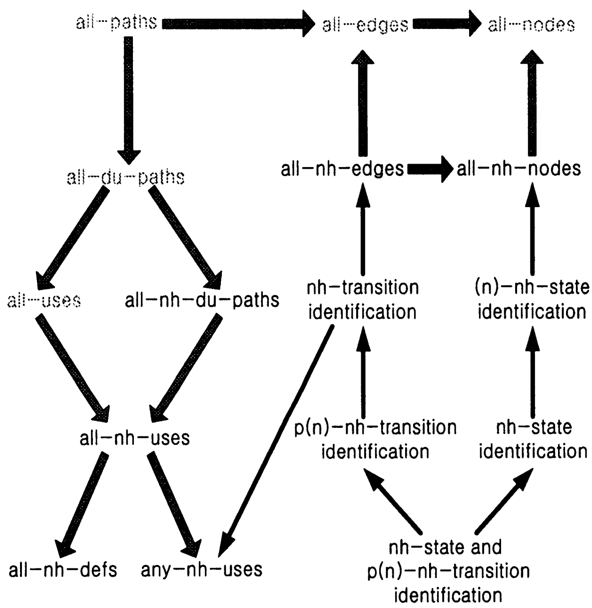


**Figure 3** The test data selection criteria of expanded EFSM

In the minimized EFSM tree, there are still not a few homogeneous states, edges and data elements. Consequently, the total number of the states, edges and data elements is so large that too many test cases are required to satisfy the existing criteria. We, therefore, defined *all-nh-du-paths*, *all-nh-uses*, *all-nh-defs*, *any-nh-uses*, *all-nh-edges*, *all-nh-node*, *nh-transition identification* and *nh-state identification* criteria, where the word 'nh' means *non-homogeneous*. The defined criteria are equivalent to the criteria without 'nh' in the case of EFSM.

In test coverage selection criteria for control flow test, we added *p(n)-nh-transition identification* criterion and *(n)-nh-state identification* criterion. A series of input parameter values in a test case that make the test case executable may not be unique. Finite or infinite number of values can be valid for the test case. Which values to select in generating the test case has been an important issue in software testing or protocol testing. How to find out the exact set of values that makes the test case executable is a theoretical interest for many researchers but the problem turned out to be undecidable in normal cases. In case of simulation, it is difficult to find out the set of executable values. We get to know, by simulation, whether a series of specific values makes the test case executable or not only. But by using more than one test cases having different values of input parameters, we can generate a test suite with more test coverage. The *p(n)-nh-transition identification* criterion is satisfied by the test suite which has $n$ test cases having different values for each input parameter. The *(n)-nh-state identification* criterion is for reduction of test case explosion by the state identification parts for all the states. The *(n)-nh-state identification* criterion is satisfied by the test suite that has the state identification parts for the specific $n$ states.

Most test case generation tools generate a test suite satisfying the *nh-transition identification* criterion while theoretic methods use ones having more test coverage such as the *all-nh-du-paths* criterion or the *all-nh-uses* criterion. The *nh-transition identification* criterion has high test coverage to cost ratio but if a test suite having more test coverage is required, the procedure of the next section will generate them effectively.

## 4 DISTRIBUTED ANALYSIS METHOD

All the constraint parts of a test suite in TTCN can be completed automatically by the distributed analysis method. This method uses two well-known techniques, the simulation and the modeling, as shown in Figure 1. Why is this method needed? Using only one of two techniques, which is usual, did not produce complete results. But two techniques are complementary each other. The modeling technique can reduce the size and time of the simulation technique. On the other hand, the simulation technique can give the executable values of input parameters. Which technique to use to generate test cases for a certain sequence is decided by whether the sequence is context dependent or independent. A sequence is said to be *context independent* if the predicate of every transition in the sequence is independent of any control variable of the specification. The sequence that is not context independent is called a *context dependent* sequence. To know the executability of the sequence, simulation is required in most cases.

The test case generation procedure using the distributed analysis method is shown in Figure 4. First, the protocol specification is modeled in an EFSM for model analysis. In this paper, we confined the analysis model to a single module EFSM without local signals. In case of the protocol specification that has communicating multiple modules, each module is modeled an EFSM individually without considering concurrency between modules. Test case generation for a concurrent system is still an open problem. When the target EFSM model is built, several fundamental test components are generated by its analysis. The test components are the *preambles* to each state, the *postambles* from each state and *def-clear path candidate sets*. We first search for the preambles and

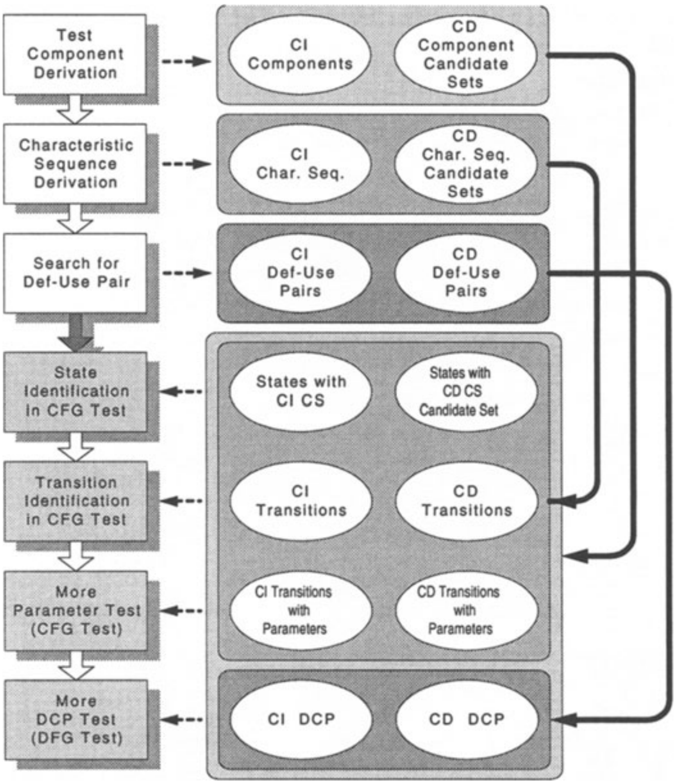


**Figure 4** The test case generation procedure using the distributed analysis method

the postambles that are context independent. Only one context independent preamble or postamble for each state is found out and used. If failing to find out any context independent component for a state, we search for all the context dependent components for the state and build the context dependent component candidate set for the state. Each element in the set is a candidate for a preamble or a postamble. Then, CS(Characteristic Sequences) of states in the model are derived in the same way as test component derivation. The context independent CS are generated in the first place and the context dependent CS candidate sets are generated for the state having no context independent CS. Then, we search for *def-use pairs* for data flow test. A def-use pair is context independent if the edge of the *def* has no predicate. Otherwise, the def-use pair is context dependent.

After those basic test components are generated, main test cases for control and data flow test are derived by direct assembling or simulation. In normal cases, generation of test cases for state identification is optional while test cases for transition identification is necessary. State identification verifies the uniqueness of CS, so it increases the fault coverage of a test suite(Anido and

Cavalli, 1995). For state identification, input sequences of all CS are fed to each state to check if generated output sequences of each CS are unique. To reduce the problem size, we can select *(n)-state identification* criterion. If there are some states having context dependent component candidate sets or CS candidate sets, we can pick up the elements fit for the other given components by simulation. Then, test cases satisfying transition identification criterion are generated. As we know, these test cases are for detection of transfer errors and output errors of transitions. The test case of a transition is composed of the preamble, the transition, CS and the postamble. If you can not find out a context independent element for any element, we can pick up the proper element in the relevant candidate set by simulation. You can increase the test coverage of the test suite by generating test cases satisfying *p(n)-nh-transition identification* criterion additionally. As shown in Figure 2 and 3, *transition identification* criterion covers *any-uses* criterion. Test coverage of any uses criterion is almost lowest among data selection criteria. Test cases for data flow test has been disregarded to a test suite designer for its big size. More test coverage for data flows, however, may be needed for more complete test. Technical advance may be able to pay the expense of data flow test. In any case, if additional test cases for data flow test are required, we can select one of the data selection criteria, find out the proper def-clear paths and generate the test cases by assembling or simulation. *All-def* criterion or *all-uses* criterion will be a good selection for its efficiency, we think.

We used the DFS(Depth-First Search) method as the search method in simulation. The search method is explained in next section with other reduction techniques of problem size. The efficacy of this distributed analysis method is discussed in section 6 with the empirical results.

## 5 PROBLEM SIZE REDUCTION TECHNIQUES

We used three problem size reduction techniques in the proposed methodology: *the DFS with termination conditions*, *the partial supplementation technique* and *the partitioning technique* to reduce the complexity of simulation.

Normally in reachability analysis, the BFS(Breadth First Search) method is used, because the reachability graph is infinite. In this paper, however, we used the DFS method because we need the reachability graph of a specific part rather than that of the whole specification. Because we have some information about the way to the target sequences, we can find out the executable path to the target sequences faster. We used termination conditions of simulation to prevent infinite simulation. Two conditions were used for termination of simulation: the characteristics of the minimized EEFSM and the satisfaction of the selected test coverage. The minimized EEFSM is generated by merging the homogeneous states having same data values. So if the homogeneous state having same data set as the already generated states are generated by simulation, the simulation is terminated. If all the sequences satisfying the selected criterion are found out, the simulation is also terminated.

We used *the partial supplementation technique* to reinforce the fault coverage of only a specific part of the specification. We can select some states and generate test cases satisfying *(n)-state identification* criterion for the states only. We can also select some input parameterized transitions and generate test cases satisfying *p(n)-transition identification* criterion for the transitions only. In the same way, we can select some variables and generate test cases satisfying higher level of criterion among data selection criteria for the variables only.

The specification has several control variables. But some control variables may lie in a specific part in a functional group. *The partitioning technique* makes use of this feature and reduce the number of control variables partially by the functional partitioning. Each partition has, therefore, smaller control variable than the whole specification, but has full information about its control variables. If the values of the control variable of the partition change, a new state having the changed data of its control variables is created. For the control variables of other partitions but not of the present partition, each partition has the data set of the variables. If the values of one of the variables change which are not the control variables in the present partition, instead of creating a new state, only the changed value is added to the data set of the variable. Only after the execution path may go out to other partitions where the variables is the control variable, according to the elements of the data set, new states are created. This technique with the above termination conditions reduces the size of the generated EEFSM by simulation considerably.

We did not use any optimization technique such as the RCP method additionally to reduce the number of test cases. Because optimization techniques will blur the test purposes and decrease the test coverage of the generated test cases in most cases(Anido and Cavalli, 1995).

## 6 EMPIRICAL RESULTS

To show the efficacy of the proposed framework and methodology, we applied the methodology to the SSCOP (Service Specific Connection Oriented Protocol) used in the B-ISDN AAL(ATM Adaptation Layer). This protocol is often chosen for conformance test because its SDL specification was standardized in ITU-T and its abstract test suite in TTCN was released in ATM FORUM. Besides, several test case generation tools such as TVEDA or SAMSTAG also generated abstract test suites of the protocol.

For generation of test cases, we adopted DS(Distributed Single layer) method, because an upper interface to the IUT exists in the method, which make it possible to generate desirable test cases with high test coverage. As the test coverage of a test suite, we selected both *transition identification* criterion and *(6)-state identification* criterion for control flow test and *all-defs* criterion mainly and *all-uses* criterion partially for data flow test. The partially selected criterion increases fault coverage of the specific part. For addition of test cases satisfying *(6)-state identification* criterion, we selected the state set, { S0(Idle), S2(Outgoing Connection Pending), S10(Data Transfer Ready), S7(Outgoing

Recovery Pending), S8(Recovery Response Pending), S4(Outgoing Disconnection Pending) }. For partial addition of test cases satisfying *all-uses* criterion, we selected the data element set, { VT(S), VR(R) }.

SSCOP has 10 states and more than two hundred edges. The state, S10 is for data transfer and the edges from/to the state are rather complex. The edges have cascaded predicates and inside recursions. We cut the edges into each segment subedges. Thus we generated an EFSM model for the specification that has 10 states, 150 edges and 98 subedges. By model analysis, we generated the preamble of each state that is context independent and the context independent postamble of each state. We derived each context independent CS for 7 states but for the states, S8, S9(Incoming Recovery Pending), and S10, we generated context dependent CS candidate sets whose elements are 2, 3 and 26 each. The SDL specification of SSCOP has 12 variables, 8 PDU parameters, 5 SSCOP parameters, 7 signal parameters, 5 timers, 15 PDUs and 23 signals. We considered only variables and timers for data flow test. The numbers of *defs* to constants/*defs* of the variables, VT(PS), VT(A), VT(PA), VT(MS), VT(PD), VT(CC), VT(SQ), VR(H), VR(MR) and VR(SQ) are 17(10), 12(12), 11/11, 21/21, 17/20, 20/24, 1/17, 18/18 and 19/19 each. The numbers of the pairs *use* to predicate or output/*def-use pairs* of the variables, VT(S) and VR(R) are (10,32)/(11, 33) and (11,9)/(13,12) each.

We used *the partitioning technique* to reduce the complexity of simulation. We separated the complex data transfer part around the state S10 and partitioned it into five parts as the input. The number of the control variables for the whole specification are twelve or all the variables, but the numbers of control variables for six partitions are 2, 3, 3, 2, 2 and 5 each. This partitioning technique, therefore, could reduce the size of generated EEFSM effectively.

We generated 72 test cases satisfying *(6)-state identification* criterion for the six states and 205 test cases satisfying *transition identification* criterion for all the edges and subedges for control flow test. For data flow test we generated 159 test cases satisfying *all-def* criterion for the ten variables and 376 test cases satisfying *all-uses* criterion for the two variables. Because of not using any optimization technique, the number of the test cases for data flow test is rather large.

## 7 RELATED WORKS AND EVALUATION

A lot of researches has been made for automatic generation of test cases for both control and data flows(Chanson and Zhu, 1993; Kim, 1995; Ramalingom *et al.*, 1995; Bourhfir *et al.*, 1997). Those researches modeled a single module of a protocol to a single module EFSM and generated test cases satisfying transition identification criterion for control flow test and *all-du-paths* or *all-uses* criterion for data flow test. They used the techniques to give the executable values of input parameters like linear programming or CIUS(Context Independent Unique Sequences) for not considering executability of the sequences. However, the techniques are valid only in the assumptions that are too strong for the real protocols. The CIUS method is not always applicable because there are many protocols that do not have the CIUS for all states like SSCOP. In addition, the protocols used for their experiments are too simple compared with real protocols and an EFSM model is too weak to model big and complex protocols such as SSCOP.

TGV, TVEDA and SAMsTAG are automatic generation tools of test suite in TTCN(Doldi *et al.*, 1996; Grabowsky *et al.*, 1997). TGV, TVEDA V3 and SAMsTAG use simulation to search for test cases relevant to given test purposes. They generate a test suite satisfying only *all-edges* criterion. TGV uses feeds and filters to reduce the problem size. But because they exclude some messages and edges, they may not be able to find out test cases relevant to a certain test purpose. TVEDA V2 uses the syntactic techniques with some constraints and generates test purposes automatically by heuristic methods. But constraint parts of a test suite generated by TVEDA V2 are incomplete and require manual completions. TVEDA V3 overcame this weakness by using simulation but it still ignores test suite having more test coverage. SAMsTAG needs test purposes in form of MSCs, from which test cases are generated by simulation. But it fails to find out test cases for a considerable amount of test purposes. Developers of these tools are now interested in completing the tools rather than in generating more powerful test suites.

ATM Forum released an abstract test suite in September 1996(ATM FORUM, 1996). The test suite uses the verifying sequence for each state and attaches it to the transition relevant to each test purpose. The test suite, therefore, satisfies almost *transition identification* criterion. But in the test suite, a considerable amount of test cases are omitted including the ones for transitions starting from four states. And the test suite uses the same verifying sequence for each state, which may make some test cases not executable.

The proposed framework and methodology consider those weakness mentioned above. They are interested in test cases having higher test coverage as well as the generation of executable test cases for real protocols by using both model analysis and simulation.

## 8 CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed new framework and methodology for automatic test case generation for a real protocol. The distributed analysis technique, the core of the proposed framework, makes use of the advantages of model analysis used in theoretical methods and simulation used in practical methods. We can select one of the redefined test coverage criteria for generating a test suite having higher test coverage. State space explosion by simulation can be reduced by the depth first search with termination conditions and the partitioning technique. We generated the test cases of SSCOP satisfying *(6)-state identification* criterion and *transition identification* criterion for control flow test and satisfying *all-defs* criterion and *all-uses* criterion for data flow test. The test cases were generated by a simple tool with a simple simulator partially and by manual efforts partially.

We are planning to develop a complete tool where the proposed framework and methodology are implemented. Besides, we are studying a method to cope with communicating multiple module protocols and time-relevant protocols for multimedia communication.

## References

Weyuker, E. J. and Rapps, S. (1985). Selecting software test data using data flow information. *IEEE trans. on Software Engineering.*, **SE-11(4)**, 367–375.

Aho, A. V., Dahbura, A. T., Lee, David and Uyar, M. U. (1988). An optimization technique for protocol conformance test generation based on UIO sequences and Rural Chinese Postman Tours. *Protocol Specification, Testing and Verification '88.*, Atlantic City, USA, 75–86.

Lee, Do-young and Lee, Jai-yong (1991). A well-defined Estelle specification for the automatic test generation. *IEEE trans. on Computer.*, **COM-40(4)**, 526–542.

Ural, H. and Yang, B. (1991). A test sequence selection method for protocol testing. *IEEE trans. on Communications.*, **COMM-39(4)**, 514–523.

Chanson, S. T. and Zhu, Jinsong (1993). A unified approach to protocol test sequence generation. *IEEE INFOCOM '93.*, San Francisco, USA, 106–114.

Li, Xiangdong, Higashino, T., Higuchi, M. and Taniguchi, K. (1994). Automatic generation of extended UIO sequences for communication protocols in an EFSM model. *Int. Workshop on Protocol Test Systems '94.*, Tokyo, Japan, 213–228.

Anido, R. and Cavalli, A. (1995). Guaranteeing full fault coverage for the UIO-based testing methods. *Int. Workshop on Protocol Test Systems '95.*, Evry, France, 221–236.

Henniger, O., Ulrich, A. and König, H. (1995). Transformation of Estelle modules aiming at test case generation. *Protocol Test Systems 95.*, Evry, France, 45–60.

Kim, Tae-hyong (1995). Automatic generation of observation-based and length-optimized test cases for EFSM model in conformance testing. *M.S. thesis.*, Dept. of Electronic Eng., Yonsei University.

Park, J. H., Hong, J. P. and Lee, J. Y. (1995). A conformance testing framework for applying test purposes. *Protocol Test Systems 95.*, Chapman and Hall, 1995.

Phalippou, M., Guerrieri, S. and Stokar, D. (1995). INTOOL/CATG - D2: User requirements. *Document INTOOL/CATG/EC/4.*, Rev 5.

Ramalingom, T., Das, A. and Thulasiraman K. (1995). A unified test case generation method for the EFSM model using context independent unique sequences. *Protocol Test Systems 95.*, Evry, France, 289–305.

The ATM Forum Technical Committee (1996). Conformance Abstract Test Suite for the SSCOP for UNI 3.1. *af-test-0067.000.*, Sep. 1996.

Doldi, L., Encontre, V., Fernandez, J., Jeron, T., Bricquir, S., Texier, N. and Phalippou, M. (1996). Assessment of automatic generation methods of conformance test suites in an industrial context. *Testing of Communicating Systems 96.*, Chapman and Hall, 347–361.

Lai, R. (1996). How could research on testing of communicating systems become more industrially relevant?. *Testing of Communicating Systems 96.*, Chapman and Hall, 3–13.

Bourhfir, C., Dssouli, R., Aboulhamid, E. and Rico, N. (1997). Automatic executable test case genration for extended finite state machine protocols. *Testing of Communicating Systems 97.*, Chapman and Hall, 75–90.

Chin, B. M., Kim, T. H., Jang, M. S., Hwang, I. S., Lee, J. Y. and Lee, S. B. (1997). Generation of Reliable and Optimized Test Cases for Data Flow Test with a Formal Approach. *ICOIN-11.*, Taipei, Taiwan.

Grabowsky, J., Scheurer, R., Dai, Z. R. and Hogrefe, D. (1997). Applying SAMSTAG to the B-ISDN protocol SSCOP. *Testing of Communicating Systems 97.*, Chapman and Hall, 397–415.

Rayner, D. (1997). Future directions for protocol testing, learning the lessons from the pass. *Testing of Communicating Systems 97.*, Chapman and Hall, 3–17.

Kim, T. H., Hwang, I. S., Jang, M. S., Kang, S. W., Lee, J. Y. and Lee, S. B. (1998). Test Case Generation of a Protocol by a Fault Coverage Analysis. *ICOIN-12.*, Tokyo, Japan.