

# Towards a Decision Support System for Automated Testing of Complex Telecommunication Networks

A. Schuster, R. Sterritt, K. Adamson, E.P. Curran, C.M. Shapcott  
University of Ulster at Jordanstown  
Shore Road, Newtownabbey, BT37 0QB, Northern Ireland

**Abstract** - This paper provides an overview of a research & development project in the area of automated testing of complex telecommunication networks. The paper reports on the current state of the project, which, for example, includes the development of a database for the organisation and the management of the data that is generated in a network test. Along this line the paper also briefly reports on a future direction in the project, a data warehouse for the management and the organisation of data that is gathered in the wider scope of the project. A major focus of the paper is the proposal and description of a decision support system (DSS). The potential of such a system has been identified in previous research. The overall aim of the DSS is to support the decision making of test engineers to further improve the automated testing of complex telecommunication networks.

**Keywords:** telecommunication networks, automated testing, data organisation, and decision support.

## 1 Introduction

This paper describes a sub-part of a large-scale research & development project that is currently carried out between Nortel Networks (from now on simply referred to as Nortel), the University of Ulster, and other partners. The described project relates to the domain of automated testing of complex telecommunication networks and is a successor of projects that have been undertaken by the two parties in the past [1] [2]. The motivation for the project is largely based on suggestions for improvements to the current system that have been identified in meetings frequently held between test engineers at Nortel and computer scientists at the university. In this paper the identified improvements can be separated into two classes:

- Management and organisation of data that is generated in a network.
- Decision support for the test engineers to aid and possibly improve automated network testing.

Potential approaches for improvement proposed by the computer scientists include:

- Design and implementation of a database/data warehouse for data storage and data organisation.
- Design of a DSS to assist the test engineers and possibly improve the process of automated testing.

At the moment the two streams are realised to different degrees. Data storage and data is largely completed via a

database prototype. The DSS to a somewhat larger extent is only conceptually defined. This is primarily due to the fact that the DSS operates on the generated data and therefore the task of data organisation has a higher priority at the moment. The DSS is intended to include a knowledge rich rule-based component, but also a knowledge poor optimisation strategy represented by a genetic algorithm. For a start a DSS prototype has been developed that operates on the bases of a simple statistical analysis (mean values, maximum and minimum values).

The remainder of the paper is organised as follows. Section 2 introduces the automated testing environment. Section 3 describes the current data management. Section 4 proposes an alternative to this approach. The same section also investigates the potential of a DSS to support and possibly improve automated network testing. Section 5 summarises the current state of the project. Section 6 finally ends the paper with a discussion and future work.

## 2 Testing Environment

Testing is a major element in the development cycle of telecommunication equipment, both in terms of cost and time-scale [3]. Historically, at Nortel, testing was performed manually. An engineer would follow a test case that was loosely coupled to the requirements. This testing would be long and repetitive requiring much expensive over-time. It is also a less attractive role for a qualified engineer. Although this was not considered the best way to spend the engineers' time it did offer the advantage that, as an expert in the field, the engineer could spot anomalies and probe further beyond the test case. Yet, generally it was felt that humans poorly did regression testing. Hence, automated testing has been introduced at Nortel, offering an advantage in terms of reduced cost, reduced time to market, and "freeing-up" of engineers for further design and development. Figure 1 (next page) illustrates one of the SDH rigs constructed by the test engineers that facilitates the testing of each release of the multiplexer software.

Typically, a SDH test network comprises many interacting components. For example, the test network illustrated in Figure 1 contains three multiplexers (EUSTON, ENFIELD, and ACTON), connected to each other in a ring topology using fibre optic cables. Each multiplexer has a series of slots which house cards with specific functions. These are Add/Drop multiplexers since they have connections to

tributaries. Once a multiplexer receives data from a tributary, it multiplexes the data into a series of frames. The frames are sent out on the fibre optic cables until it arrives at its destination multiplexer. The frame is then demultiplexed and sent on the appropriate tributary. In the test network, neighbouring tributaries are connected together so that there is a continuous cycle of data being transmitted. Any events that occur on a component (e.g., event type, event time) are detected by that component and sent to the Network Manager in Figure 1 via the Hub. The Network Manager logs these events in a file known as the Event Log.

A series of UNIX test scripts have been developed to automate the testing. They apply commands, or stimuli to the multiplexers in the network. These stimuli will result in various cause and effect relationships including the generation of faults on certain components. For each fault, the affected components will raise one or more alarms. These alarms are special types of events, and as such are cascaded back to the Network Manager, which logs them in the Event Log. During testing the Event Log can rapidly grow since it logs all events. For instance, the simulation of a fibre optic break generated 6Mb of data. Although the automation of the testing provides substantial cost and time savings some of these may be negated by having the engineer manually analyse the Event Log. In effect, the means by which the testing is carried out changes from real-time to batch losing the advantage of additional real-time probing into the network's state. To overcome this inherent disadvantage, the test engineer when coding the test script includes feedback comments, which are output in a file, referred to as MUX LOG in Figure 1. Since these logs will indicate specifically and clearly the results that the script was designed to test, this allows the engineer to quickly identify if the test was a 'pass' or a 'fail'.

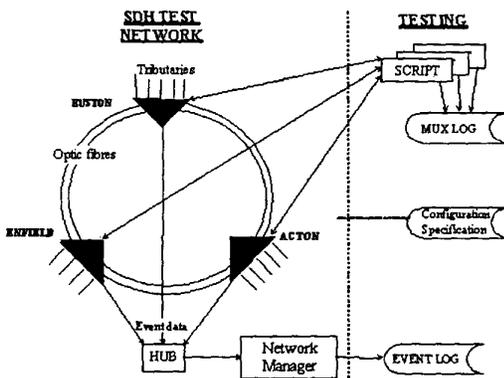


Figure 1: Automated testing set-up.

Historically, test engineers at Nortel organise and store data that is generated in a network test in a hierarchy of folders and sub-folders on PCs running Windows NT. Figure 2 illustrates a section of this hierarchy, and also uses the folder 04-05-1999\_EC13-1bb\_EEA37\_001 for explanation purposes.

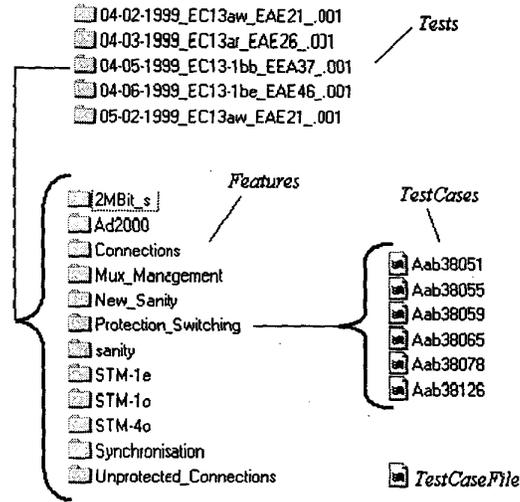


Figure 2: Folder hierarchy.

The first folder level indicated as Tests in Figure 2 identifies a network test in terms of test date (04-05-1999), element controller load (EC13-1bb), network element load (EEA37), and a counter for the number of tests carried out on a particular day (001). The second level contains the possible Features or areas (e.g. 2Mbit/s, Ad2000, Connections) that can be tested in a network test. The level referred to as TestCases in Figure 2 holds the test cases that have been tested in a particular area. (Note: a test case is a script/program that imposes certain actions in the network. In total there are about 800 individual test cases.) For example, Figure 2 illustrates that the test cases Aab38051, Aab38055, etc., have been used for certain tests in the area Protection\_Switching. Additionally to the data that is captured in the folder hierarchy the engineers store and use a file, in Figure 2 simply referred to as TestCaseFile, in which further test information is recorded (e.g., whether the outcome of a test case was a pass or a fail). Further, the intranet and the WWW are used to present test results for interested groups inside Nortel. Figure 3 illustrates the wider view of data organisation and data presentation. Figure 3 also indicates the use of the Perl programming language for different tasks.

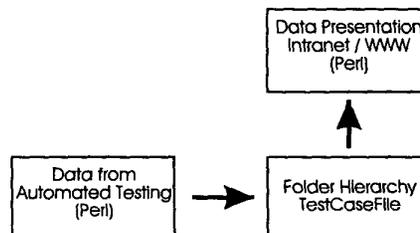


Figure 3: Data organisation and data presentation

## 2.1 Shortcomings

In simple terms the folder hierarchy shows all the disadvantages such a file structure has over a, for example, relational database structure. For instance, it does not allow the quick formulation and execution of complex or even simple queries. It is also time expensive to perform a comprehensive statistical analysis on a test, since this can only be done via specifically and manually written programs. The same accounts for the generation of reports, charts, etc., and so the engineers may be restricted in the evaluation and analysis of a test.

There are also issues from the viewpoint of the DSS that require a more versatile data storage facility. For example, it would be helpful for the engineers to have a means for the identification of pass/fail patterns, given that such patterns occur in the network. The testing process would also benefit from optimisation strategies for network tests that are constrained by certain requirements. To maximise the number of test cases in a given time slot could be such a requirement. Another demand might be to identify test cases that over time did show a 'high', 'somewhat high', or a 'low' pass/fail rate. These tasks all could benefit from a data source that is organised and structured and hence comfortably and quickly accessible.

## 3 Database, DSS and Data Presentation

A solution proposed to the problems discussed in the previous section is illustrated in Figure 4.

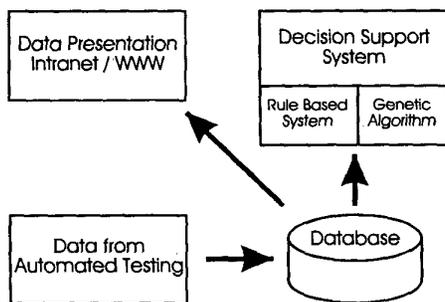


Figure 4: Database, DSS and data presentation.

Figure 4 illustrates that the test data is going to be stored in a database. The database is the main source for the DSS, but also for the data presentation on the intranet/WWW. The activities in the project are characterised by the motivation to identify, explore, and test solutions for the database, the DSS, and data presentation.

### 3.1 Database

A database design has been defined, and a prototype has been built using Access97. For example, the design consists

of three tables and simple one-to-many relations. The test engineers currently use this prototype to explore whether it captures the data that is generated and processed in the current Figure 3 set-up. They are also encouraged to specify further database requirements in terms of a comprehensive statistical analysis of a network test, queries, reports, charts, etc. Access97 is used as a test case. The data generated in the long run may require a larger database. The database efforts therefore include the investigation of alternative database packages, for example, PostgreSQL, Microsoft SQL Server, and Informix.

At this stage a potentially wider scope of the project has to be mentioned. The automated testing domain is a very large area containing more than the described data source. One of the future directions in the project therefore is the design of a data warehouse to capture and manage the data of these different sources [4]. The database packages mentioned before provide the capability to include the proposed database into a data warehouse and therefore it is worthwhile to investigate their potential.

### 3.2 DSS

The basic design steps and components of a DSS consist of three parts: knowledge acquisition, knowledge representation, and inference [5] [6]. The same accounts for the DSS in this project. The knowledge acquisition process in this project basically includes:

- To get familiar with the automated testing domain.
- The extraction of the domain expert's (test engineers) domain knowledge by the knowledge engineers.
- To understand and identify problems that potentially can be addressed by a DSS.

Knowledge presentation means to select a problem-solving strategy that represents the problem domain adequately for problem solving. For example, some of the identified problems have been classified as optimisation tasks. It is frequently the case that there is only a certain time slot available for a network test. In such a case test engineers may intend to optimise a network test based on certain requirements (e.g., include as many cases as possible in the time slot, or, include mainly test cases that did show a high, or somewhat higher fail rate in the past). To meet these requirements the DSS in Figure 4 contains a rule base component as well as a genetic algorithm component. The two components are intended to work complementary [7]. To capture the uncertainty prevailing in statements like 'a somewhat higher fail rate' the rule base is likely to include a fuzzy component [8]. Inference finally describes the reasoning methods and mechanisms employed in the decision making process of the system. The methods can be knowledge poor as in the case of a genetic algorithm, but also knowledge rich as in the case of (fuzzy) rule based systems [9] [10]. In previous work researchers involved in the project have been developing and testing techniques and

methods that might be applicable and useful in the current project [11] [12].

The previous section mentioned the wider scope of the project in which the database is very likely to be integrated into a data warehouse. In the same wider scope it is certainly appropriate to refer to future development stages of the DSS as a hybrid system.

### 3.3 Data Presentation

The data presentation in Figure 3 uses an intranet and the WWW. The folder hierarchy and the TestCaseFile (Figure 2) are accessed via programs written in Perl. The extracted information is put forward on a web site making it more accessible. The approach presented here intends to use the database and active server pages for the same but also for more complex tasks. The more complex tasks include the definition of interactive web-based queries on the database for example. For the time being the test environment is an Access97 database on a PC running Windows NT 4.0. Web access is realised using Microsoft Internet Information Server and VB Script for the design of active server pages.

## 4 Current State

A number of prototypes have been developed and demonstrated. The demonstrations indicated the potential of the proposed approaches. The current state of the project therefore can be summarised by the following activities:

- Implementation and integration of the database design using PostgreSQL.
- Implementation of various scripts facilitating data preprocessing and data import into the PostgreSQL database for example.
- Implementation and testing of a DSS prototype.
- A strategy for statistical analysis and data mining has been completed. However, more data is required before this strategy can be applied.

After these tasks are completed we should be in a position to test the value of the DSS. The statistical analysis and data mining activities may also provide useful information in the context of network testing.

## 5 Discussion and Future Work

The goal of the paper was to describe the current state and future directions of a project that is currently carried out by the University of Ulster, Nortel, and other partners. The paper discussed a data management approach conducted at Nortel. Proposed improvements to this approach have been implemented in the form of a database prototype, a DSS prototype and a data presentation approach via active server

pages. The paper mentioned the wider scope of the project, leading to a data warehouse and a hybrid DSS. The potential of these techniques for some of the problems in the area of automated testing has been discussed throughout the paper.

Future work is along these lines. Currently researchers are involved in the task of generating active server pages for database querying and data presentation. Another short-term objective is a thorough statistical analysis of the data captured in the database. An important aim of the project however is the development of the DSS. Work is currently underway to implement a DSS prototype in the automated testing environment and to test its performance under real circumstances. The feedback obtained in these studies might be useful for the design and testing of other strategies in the domain, for example genetic algorithms or swarm intelligence.

### Acknowledgements

We are very grateful to Nortel for their support, in particular to the people that are more closely involved in the project for the efforts they make in this collaboration.

### Abbreviations

DSS = decision support system, Nortel = Nortel Networks, SDH = synchronous digital hierarchy, WWW = World Wide Web.

### References

- [1] Sterritt R., Adamson K., Shapcott CM., Bell DA., 1998. "An architecture for knowledge discovery in complex telecommunication systems". Editors: Adey R.A., Rzevski G., Nolan P., Artificial Intelligence in Engineering XIII, Computational Mechanics Publications, Southampton, pp 627-640.
- [2] Sterritt R., Curran EP., Adamson K., Shapcott CM., 1998. "Application of AI for automated testing in complex telecommunication systems". Proceedings of 10<sup>th</sup> International Conference on Artificial Intelligent Applications, EXPERSYS 98, pp 97-102.
- [3] Bouloutas AT., Calo S., and Finkel A., 1994. "Alarm correlation and fault identification in communication networks". IEEE Transactions on Communication, Vol. 42, No 2/3/4, Feb/Mar/Apr, 1994.
- [4] Mattison R., 1997. *Data Warehousing And Data Mining For Telecommunications*. Artech House, Boston, London.
- [5] Luger G.F. and Stubblefield W.A., 1998. *Artificial Intelligence: Structures And Strategies For Complex Problem Solving*. Addison Wesley.
- [6] Graham I. and Jones PL., 1988. *Expert Systems, Knowledge, Uncertainty And Decision*. Chapman and Hall, London, New York.
- [7] Burattini A., 1995. *Diagnostic Reasoning In Rule-based Neural Systems*. Chapman & Hall.
- [8] Ross TJ., 1995. *Fuzzy Logic With Engineering Applications*. McGraw-Hill, New York, London.
- [9] Mitchell M., 1996. *An Introduction To Genetic Algorithms*. MIT Press, Cambridge, Massachusetts, London.
- [10] Cox ED., 1995. *Fuzzy Logic For Business And Industry*. Charles River Media, Rockland, Massachusetts.

- [11] Schuster A., Adamson K., and Bell DA., 1999. "An application of a genetic algorithm for rule weight optimisation in a fuzzy expert system". IASTED International Conference on Artificial Intelligence and Soft Computing, pp 572-576, Honolulu, Hawaii, USA.
- [12] Schuster A., Dubitzky W., Lopes P., Adamson K., Bell DA., Hughes JG., and White JA., 1997. "Aggregating features and matching cases on vague linguistic expressions". Proceedings of the 15th International Joint Conference on Artificial Intelligence IJCAI'97, pp 252-257, Nagoya, Japan.