Specification-based Detection of Telecom Service Degradations*

A. M. da Silva Filho
J. T. Saito
Department of Informatics
State University of Maringa
amendes|jts@din.uem.br
87020-900, Maringa, PR, Brazil

Abstract

Detecting software failures is a major requirement today as it is mainly used for real world applications. With the critical role played by software systems in the operation of telecommunications networks, the ability to detect and report failures has become of great importance. This paper presents the specification-based detection telecom service degradations. A real-time supervisor can automatically detect software failures based on boundary signals and the requirement specification of the target software system. An implementation of a real-time supervisor has been shown to be able of detecting software failures in a telecom system.

1 Introduction

The history of telecommunications show a clear trend towards automatic detection of service degradations. When done in real-time, such detection permits corrective action to be taken before the degradations affect or are even noticed by users of telecom services.

This trend has been very evident in hardware. For example, in interoffice trunking, the initial reliance on users to report transmission problems had given way to periodic trunk testing and, with the advent of digital trunks, to continuous monitoring of error rates.

Automatic detection of software failures has lagged behind. At present, telecom service providers rely on a collection of special purpose measures, including watchdog timers, software audits, and test calls [2, 1, 5, 4]. The failure fration of field problems reports are eventually traced back to software causes.

Note that software systems are playing a critical role in the operation of today's telecommunications networks. The steadily growing size and complexity of these systems are making it extremely difficult to exaustively test the software to ensure that it will adequately perform its specified function.

In addition, with this increasing reliance on software for implementation of telecom service functionality, the need for automatic detection of software failures is be-

* This research is supported by the Brazilian Council of Research (CNPq).

coming acute. In addition, the current trends toward multi-supplier networks, in which the localization of a software fault is of paramount importance, requires such a capability.

2 Specification-based Supervision

Herein, this paper deals with one general approach for the automatic detection of software failures, so-called specification-based supervision. Within this context, the software supervisor is an enhanced *clone* of target system being supervised. The supervisor monitors the inputs and outputs of the target system, as illustrated in Figure 1.

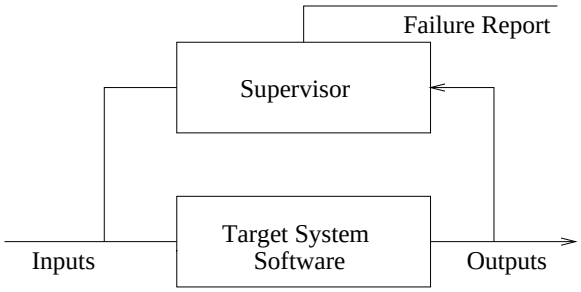

Figure 1: Software system supervision.

It is worth pointing out that the supervisor executes a model formally derived from the target system software specification. The supervisor checks the observed behavior against the specified one and reports discrepancies that occur as failures.

A major issue in specification-based detection of failures is the nondeterminisms present in the specification. Under some input scenarios, these nondeterminisms result in several distinct but legitimate behaviors of the target software system. The supervisor is able to recognize all such behavioral alternatives such that they are not reported as failures.

We have applied specification-based supervision to a small telephone exchange. The supervisor was in charge of monitoring the behavior of the control program implementation for the telephone exchange [3]. The faults contained in the implementation were known. Failure data were collected for the implementation operating under different operational profiles of random traffic, ranging from light loads to heavy loads. Supervisor has been able to detect and report failures.

Despite we have applied specification-based supervision to a small telephone exchange, the results obtained so far provided a clear demonstration of its feasibility.

References

[1] Y. Huang and C. Kintala. Software Implemented Fault Tolerance: Technologies and Experience. *The 23rd International Symposium on Fault-Tolerant Computing (FTCS-23), IEEE Computer Society Press*, pages 2–9, June 1993.

[2] P. A. Lee and Andersen T. Fault Tolerance - Principles and Practice. *Dependable Computing and Fault-Tolerant Systems Series, Springer-Verlag*, 3, 1990.

[3] Silva Filho A. M. Toward More Reliable Telecom Systems. *The 23rd International Computer Software and Applications Conference (COMPSAC9), IEEE Computer Society Press*, pages 151–156, Oct 1999.

[4] M. N. Meyers, W. A. Routt, and K. W. Yoder. No.4 ESS: Maintenance Software. *The Bell System Technical Journal*, 56(7):1139–1167, Sept 1977.

[5] B. Penney. The DMS-100: A Switch that takes care of itself. *Telesis*, 4, 1980.