

Failure Detection in Telephone Switching Systems

A. M. Silva Filho, M. K. I. Doi, A. M. P. Lima
Department of Informatics
State University of Maringa
amendes | mkd | amplima@din.uem.br

1 Introduction

Today software systems are playing a critical role in the operation of telecommunication networks. The steadily growing size and complexity of these systems are making them extremely difficult to exhaustively test the software to ensure that it will adequately perform its specified function. With the increasing reliance on software systems for implementation of telecom service functionality, the need for automatic detection of software failures is becoming of paramount importance. Furthermore, the current trends toward multi-supplier networks, in which the localization of a software failure is critical issue, requires such a capability. This paper presents an approach for failure detection in telephone switching systems. A software supervisor can automatically detect software failures based on boundary signals and the target software system specification.

2 Software Reliability

Several approaches have been proposed to improve the system's software reliability as well as to provide a fault-tolerance feature. Two of the major techniques are N-version programming [1] and recovery blocks [2]. Although both approaches have been shown to achieve a reduction in software failures, they have not proved to be cost-effective [4]. Other approach for indirect software reliability improvement is called software supervision [6, 7, 8]. We have used A(ctual) hypotheses to reduce the computational cost of supervision, discussed in Section 3. Other study addressing supervision uses conditional-belief approach as reported in [5].

Software supervision includes the following features: behavioral monitoring, failure detection and reporting. Supervisor consists of an enhanced executable specification of the target system derived from the target system specification [6, 7]. Note that software supervision differs from other approaches in that failures are detected based

on a system's specification [8]. The advantage is that several versions of identical software need not to be produced and therefore the cost of developing and maintaining additional software is less for software supervision compared to others.

3 Actual Hypotheses

To choose actions to be taken, the supervisor must be able to make hypotheses and subsequently revise their assumptions when discoveries contradict these hypotheses. These discoveries results in A-hypotheses (or actual hypotheses) which allow reduce computational cost. Within this context, at any point in a target system specification where nondeterminism causes multiple valid paths to exist, a hypothesis is created to represent each possible path. These hypotheses evolve concurrently and are represented as threads. Each thread can be thought of as a process which is associated with a hypothesis. When an output arrives which invalidates the hypothesis represented by that thread, it is terminated. If all the threads in a hypothesis set are terminated, a failure has occurred, since no hypothesis remains to explain the behavior. We have used Statecharts [3] convey this notion of a-hypothesis being held by the supervisor at a time σ_i is given. Consider the sequence of time steps $\{\sigma_i\}_{i \geq 0}$, corresponding to the sampling rate of the System Under Description (SUD). The time intervals are defined by $I_i = [\sigma_i, \sigma_{i+1})$. At σ_{i+1} , the SUD reacts to external stimuli occurring in the interval I_i . Three external stimuli are associated with the interval I_i is a triple (Π, Θ, ξ) where:

- Π - set of external events occurring in I_i ,
- Θ - set of external conditions whose values are true at $[\sigma_i, \sigma_{i+1})$ for some $\sigma \geq \sigma_i$, and
- ξ - a function determined by the external environment, such that for a variable v , $\xi(v) = x$ if v 's value is x in $[\sigma_i, \sigma_{i+1})$ for some $\sigma \geq \sigma_i$.

A system configuration associated with an instant σ_{i+1} is a tuple $(\mathcal{S}, \Pi, \Theta, \xi)$ where \mathcal{S} is the maximal state configuration of the root state and (Π, Θ, ξ) is the external stimulus associated with I_i . A configuration is a maximal set of states that the system can be in simultaneously. The set of a-hypotheses held by a system at a time σ_i is given by the tuple $(\mathcal{S}, \Pi, \Theta, \xi)$.

4 Case Study

A telephone switching system (TS) is responsible for establishing and maintaining connections among telephone lines. The system example is a small TS which provides the basic functionalities to its subscribers. Operational diagnostics attempt to detect failures that, being reported earlier enough to an operator, can minimize their effects.

The specification of our telephone switching system (TS) consists of two blocks, named *CallHandler* and *ResourceManager*. CallHandler (CH) performs the main TS operations, since it supports the placing of a telephone call by detecting the lines going on and off, and the numbers dialed, providing dial tones, and connecting to the appropriate lines. ResourceManager (RM) checks for the availability of resources to process a requested call. For instance, consider that the subscriber of phone 1 picks up his phone. As a result, a signal OffHook1 is sent off to the TS passing through a channel, C1, and arrives at the process CH1. The same would take place for phone 2, and so forth. It motivates multiple, but legitimate behaviors likely to occur. The nondeterministic channel delay is represented by abstracting the concept of *signal in-transit*. Only legitimate behaviors are taken into consideration. After a signal is sent off through a channel, two hypotheses are created: one, that the signal is consumed and another that the signal is still *in-transit*. For instance, consider the case where both signals *OH1* and *OH2* are sent at practically the same time, i.e., *OH2* is sent off at a time δ (fraction of a second) immediately after *OH1* has been sent off as shown in Figure 1a. Note that the signals *OH1* and *OH2* are sent through different channels. In such an example, depending on which signal *OffHook* gets to the TS first, it will be processed provided that the TS has resource available to process a call. Consider Figure 1a, the indeterminate delay of channels gives rise to five combinations of signals arrivals shown in Figure 1b.

In the current version, TS is capable of servicing up to 60 phones and capable of carrying up to 15 simultaneous calls. The scenario used for the purpose of simulation consists of the TS emulator, the load generator and the supervisor. The TS emulator reacts to the stimuli received from the load generator. These stimuli and responses are

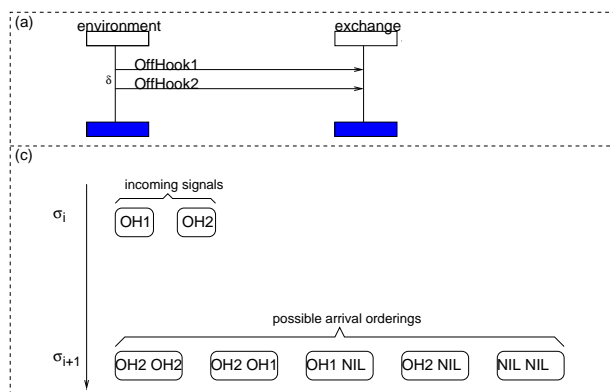


Figure 1: Notion of Signal in-Transit.

monitored by the supervisor and if any behavior which is not legitimate occurs, it is reported as a failure. The load generator generates telephone actions, such as OffHook, OnHook, and communicates them to the TE emulator. As well, the load generator is able to allow different load levels, as well as place simultaneous calls. Hence, simulations are carried out by using traffic loads mimicking the actual environment which a TE faces with. The major indicator used to assess the supervisor's capability is its ability to monitor and detect failures in real-time [7].

References

- [1] A. Avizenis. "The N-Version Approach to Fault-Tolerant Software". *IEEE Trans. on Software Engineering*, pages 1491–1501, Dec. 1985.
- [2] J. H. et al. "A Program Structure for Error Detection and Recovery". *Lecture Notes in Computer Science 16*, Springer-Verlag, pages 171–187, 1974.
- [3] D. Harel. "Statecharts: A Visual Approach to Complex Systems". *Sci. Comput. Program.* 8, pages 231–274, 1987.
- [4] J. C. Knight and N. G. Leveson. "An Experimental Evaluation of the Assumption of Independence in Multiversion Programming". *IEEE Trans. on Software Engineering*, pages 96–109, Jan. 1986.
- [5] J. Li and R. Seviara. "Automatic Failure Detection with Conditional-Belief Supervisors". *Proc. of ISSRE'97*, pages 4–13, 1997.
- [6] A. M. SilvaFilho. "On Deriving Statecharts Supervision Models from SDL Specifications using SSM". *Proc. of 10th Brazilian Symposium on Software Engineering*, pages 73–83, 1996.
- [7] A. M. SilvaFilho. "Toward More Reliable Telecom Systems". *Proc. of COMPSAC'99*, pages 151–156, 1999.
- [8] A. M. SilvaFilho. "Specification-based Detection of Telecom Service Degradations". *Proc. of ISSRE'2001 (Fast Abstract)*, 2001.