ABSTRACT

To familiarize the reader with the field of fault tolerance, this report discusses the most important terms and paradigms used in that field. After establishing a basic terminology, the fundamental techniques to achieve fault tolerance, i.e., the basic ways to employ redundancy, are identified. In particular, the rôle that fault hypotheses play in the design of a fault tolerant system is illustrated.

To enable the development of formal methods for fault tolerance, the interaction between fault hypotheses and design decisions is analyzed in detail for two fault tolerant systems. The first is a stable virtual disk which is implemented using a number of unreliable physical disks. The second concerns a reliable broadcast protocol.

Keywords  Fault tolerance, fault hypothesis, redundancy.

1.1 OF FAULTS AND FAILURES

According to Laprie (cf. [13]) fault tolerance is the property of a system "to
provide, by redundancy, service complying with the specification in spite of faults having occurred or occurring”. This report introduces the paradigms and associated terminology commonly used in the field of fault tolerance.

A system consists of components which interact as described by a design. There is no conceptual difference between the notions ‘system’ and ‘component’: the system is simply the component under discussion. The major difference between ‘system’ and ‘environment’ is that you can control the system but not the environment.

The service delivered by a component is the visible abstraction of the component’s behaviour at the point of interaction — the interface. The behaviour of a system can easily be separated into two distinct categories: behaviour in accordance with the specification, and behaviour not in accordance with the specification. We refer to these kinds of behaviour as normal and abnormal, respectively. A failure occurs when the behaviour of a system deviates from that required by its specification [19]. The failure of a component appears to the system as a fault. Notice that there is no basic difference between ‘fault’ and ‘failure’: they are merely used to distinguish the cause from the consequence. In this report we do not consider system failures that are caused by design faults.

Faults are usually classified according to the specific aspects of the specification they violate. Timing faults, for instance, can be divided into late behaviour, which can lead to omission, and early behaviour, which can lead to overrun. Another example is the occurrence of a range fault the moment a value does not conform to the specified range.

If it is possible to deduce from assertions about a component’s behaviour that some fault has occurred, we call that fault detectable. Different fault models arise from the assumptions about the correctness of the behaviour with respect to the various specification aspects, and, in case that behaviour is not assumed to be correct, the detectability of such faults.

When discussing hardware defects, the notions ‘transient’ and ‘permanent’ are well established [1]. A transient defect is present for only a limited period of time (no longer than some threshold) after which it spontaneously disappears. Any defect which is present for longer than that threshold period is said to be permanent. Analogous to this, a system failure transient or permanent.

The remainder of this report is organized as follows: in Section 2 the various stages of the procedure of tolerating faults are mentioned and it is discussed