Chapter 14
Fault Detection and Recovery Mechanisms and Techniques for Service Oriented Infrastructures

Andreas Menychtas
National Technical University of Athens, Greece

Kleopatra G. Konstanteli
National Technical University of Athens, Greece

ABSTRACT
The need for guaranteed QoS and efficient management in Service Oriented Infrastructures is an essential requirement for the deployment, execution, and management of modern business applications. In that frame, the capabilities for fault detection and recovery in all layers of a Service Oriented Infrastructure are essential for the smooth operation of the business applications and the wide adoption of these solutions in the global market. In this chapter, we present the concepts of fault detection and recovery, including terminology, classification of faults, and analysis of the key processes taking place in a system in order to diagnose and recover from failures. The state of the art mechanisms and techniques for fault detection and recovery are also analyzed, while recommendations for applying them in Service Oriented Infrastructure are presented.

INTRODUCTION
Service Oriented Infrastructures, and generally the Distributed Systems, are increasingly considered to be ideal candidates for implementing platforms capable to support business applications. This is tightly coupled with the need for high quality of service (QoS) provisioning for the applications not only to guarantee their smooth operation and management but also to support the primary capabilities of such systems for scalability, service orchestration, autonomous management and abstraction of the systems’ complexity. In this frame, fault tolerance functionality is necessary to achieve high QoS provisioning for both the system and the applications. The advantage of Service Oriented Infrastructures from the fault tolerance perspective is significant. These systems can be easily made redundant, which is the corner-stone for all fault tolerance techniques. Unfortunately,
distribution also means that the imperfect and fault prone physical world cannot be ignored, so that as much as they help in supporting fault tolerance, distributed systems may also be the source of many failures.

On the other hand, the building blocks and components of the SOIs are independent of each other and therefore are independent points of failure. Even though this may be an advantage for the end users, it is also a complex problem for the developers and administrators in terms of synchronization and management. In fault tolerant distributed systems, a component failure means that the other components and services have to detect and handle that failure to keep the system running and maintain the QoS level for the applications and end users to an acceptable level. This involves redistributing the functionality from the failed component to other, or it may mean switching to some kind of emergency mode for the operation of the systems and the execution of the application.

Providing fault tolerant design for every component is normally not an option. In such cases the following criteria may be used to determine which components should be fault tolerant:

- How critical is the component? In a car, the radio is not critical, so this component has less need for fault tolerance.
- How likely is the component to fail? Some components, like the drive shaft in a car, are not likely to fail, so no fault tolerance is needed.
- How expensive is it to make the component fault tolerant? Requiring a redundant car engine, for example, would likely be too expensive both economically and in terms of weight and space, to be considered.

In the next sections of introduction we present the terminology related with fault detection and fault tolerance systems since many of these terms are confused. In addition we have categorized the faults that may occur in SOIs based on their duration, cause and effect and we analyze their most important drawbacks. In section two, the key processes that take place in such systems to provide the fault tolerance functionality are described. Section three presents the state of the art mechanisms and techniques for the fault tolerance while section four concludes our work and highlights the future trends.

**Terminology**

Nowadays mechanisms and techniques that maintain the desired QoS levels and smooth operation of systems, services and applications are widely adopted and used, while in terms of implementation and design, they span in various different domains following different approaches. Often the terms related with the technologies for fault tolerance, detection and recovery are confused and hard to distinguish, for instance the terms “failure” and “error” or the techniques of “redundancy” and “replication”. In this section we introduce the key terms for the fault tolerant systems based on Randell, B., Lee, P., and Trelleven, P. C., (1978), Selic B., (2004), IFIP 10.4 Working Group on Dependable Computing and Fault Tolerance, (2010) and Ding, S.X., (2008).

- **Fault Tolerance**: Fault tolerance is the ability of a system to perform its function correctly even in the presence of internal faults. The purpose of fault tolerance is to increase the dependability of a system.
- **Fault Prevention**: Fault Prevention refers to techniques, such as inspection, whose intent is to eliminate the circumstances by which faults arise.
- **Fault Detection**: is the detection of occurrence of faults in functional units of the process, which lead to undesired or intolerable behavior of the whole system.
- **Fault Isolation**: describes the process of localization (classification) of different faults.
Related Content

Evaluating the Java Native Interface (JNI): Leveraging Existing Native Code, Libraries and Threads to a Running Java Virtual Machine
[www.igi-global.com/article/evaluating-the-java-native-interface-jni/202382?camid=4v1a](www.igi-global.com/article/evaluating-the-java-native-interface-jni/202382?camid=4v1a)

A Comparative Study and Algorithmic Analysis of Workflow Decomposition in Distributed Systems

Keys for Administration of Reconfigurable NoC: Self-Adaptive Network Interface Case Study
[www.igi-global.com/chapter/keys-administration-reconfigurable-noc/44221?camid=4v1a](www.igi-global.com/chapter/keys-administration-reconfigurable-noc/44221?camid=4v1a)

Collaboration Support for Activity Management in a Personal Cloud Environment
[www.igi-global.com/chapter/collaboration-support-activity-management-personal/72254?camid=4v1a](www.igi-global.com/chapter/collaboration-support-activity-management-personal/72254?camid=4v1a)