Chapter 6
Functional Safety of Distributed Embedded Control Systems

Atef Gharbi
INSAT, Tunisia

Hamza Gharsellaoui
INSAT, Tunisia

Mohamed Khalgui
Xidian University, China

Samir Ben Ahmed
INSAT, Tunisia

ABSTRACT

This chapter deals with the functional safety of distributed embedded control systems following the component-based approach. The authors define a new concept of components called "Control Component" (CC) to cover all of the used technologies in industry. To guarantee the functional safety of distributed control software components, the authors define an agent-based architecture where an intelligent software agent is deployed in a device of the execution environment in order to apply local reconfiguration scenarios, and a coordination agent is used for inter-devices coordination in order to allow coherent reconfigurations.

INTRODUCTION

We model the whole distributed system by Net Condition/Event Systems (which is considered as an extension of Petri Nets) and we verify some properties concerning functional safety using the model checker SESA.

However, verifying some properties on the whole model may lead to combinatory explosion. To avoid this problem, the refinement approach, which permits to specify automatically feasible Control Components checking the correctness of each one of them, is proposed. Finally, to ensure the functional safety of the whole distributed control systems, we define a communication protocol so that when a specific agent applies a new reconfiguration, the other agents must be aware in

DOI: 10.4018/978-1-4666-0294-6.ch006

order to put the whole system in a coherent state. We develop a complete tool "ProtocolReconf" to simulate the communication protocol. Two Benchmark Production Systems are used as a running example to explain our contribution.

The development of distributed embedded control systems is not a simple task to perform by considering their classical functional and temporal constraints, in addition to their time to market that should be shorter than ever. Among the proposed solutions is the use of software component-based approach. In fact, the software component-based approach has become very popular during the last recent years as it is possible to reuse the already developed software components for the generation of new systems. This advantage reduces the time to market and allows minimizations of the design complexity by supporting the system's software modularity. The software components that we assume in this book chapter as recomposed units of algorithms and interfaces that should classically satisfy user constraints. Nowadays, several component-based technologies have been proposed such as JavaBeans (related to Sun society), Component Object Model (related to Microsoft society) and Corba Component Model (provided by the Object Management Group (OMG)) (Artist-Project, 2003). However, there are few technologies (such as Koala, PBO, PECOS . . .) which are currently used for the development of embedded systems. Anyway, each component-based technology has its benefits and its drawbacks. Nowadays, the Functional Safety of Distributed Embedded Control Systems is considered as a crucial point to study because any fault may lead to catastrophic hazard. To outline the importance of the Functional Safety in Distributed Embedded Control Systems, some examples of failures with dramatic consequences are briefly noted below (Baier & Katoen, 2008). The Ariane 5, launched in 1996, was damaged only 36 seconds after the launch due to conversion from 64-bit float to 32-bit integer. The fault in Pentium II of Intel due to floating-point division unit causing a loss of 475 million dollars to replace the faulty processors. The airport of Denver was delayed to open for 9 months due to faults in baggage handling software leading to a loss of 1.1 million dollars per day. To be clear, we define the following terms first:

Functional Safety (Krosigk, 2000): "*In order to achieve functional safety of a machine or a plant the safety related protective or control system must function correctly and, when a failure occurs, must behave in a defined manner so that the plant or machine remains in safe state or is brought into a safe state*."

Embedded System (Colnaric, Verber & Halang, 2000): "*special-purpose computer system designed to control or support the operation of a larger technical system (termed the embedding system) usually having mechanical components and in which the embedded system is encapsulated*."

In fact, the Functional Safety of Distributed Embedded Control Systems concerns the study of two kinds of faults: hardware faults and software faults. Generally, it is known that hardware faults are easier to treat as it is possible to characterize them and to prevent them. Three mechanisms are proposed for hardware faults: fail-safe device, fault-tolerant device or redundancy (Peters & Peters, 2002). A fail-safe device means that a failure may occur and the corresponding device needs to be repaired or replaced. A fault-tolerant device means that a failure may take place and the corresponding device is still running correctly (example of research work on this domain (Ilic & Troubitsyna, 2005)). Redundancy device means the presence of an extra device which may be activated in parallel with another identical device (active redundancy) or just activated whenever a failure affects the other device (passive redundancy). However, software faults are more difficult to treat as it is not possible to characterize them. Besides, the means to treat software fault may range from simple to very sophisticated solution. Software fault may be due to wrong behavior which means that the model (i.e. the real behavior) does not correspond to the requirements (i.e. the desir-