Available online at www.sciencedirect.com
SCIENCE DIRECT®
Reliability Engineering and System Safety 82 (2003) 195–206
RELIABILITY ENGINEERING & SYSTEM SAFETY
www.elsevier.com/locate/ress

Analysis of fault tolerance and reliability in distributed real-time system architectures

Stephan Philippi

Department of Computer Science, University of Koblens, P.O. Box 201 602, 56016 Koblenz, Germany

Received 27 January 2003; accepted 26 June 2003

Abstract

Safety critical real-time systems are becoming ubiquitous in many areas of our everyday life. Failures of such systems potentially have catastrophic consequences on different scales, in the worst case even the loss of human life. Therefore, safety critical systems have to meet maximum fault tolerance and reliability requirements. As the design of such systems is far from being trivial, this article focuses on concepts to specifically support the early architectural design. In detail, a simulation based approach for the analysis of fault tolerance and reliability in distributed real-time system architectures is presented. With this approach, safety related features can be evaluated in the early development stages and thus prevent costly redesigns in later ones.

Keywords: Systems modeling; Simulation; Fault injection; Petri-Nets; Information horizon; x-by-wire

1. Introduction

Distributed real-time systems are increasingly becoming more common in areas, which can be attributed as critical for different reasons. Systems from such diverse sectors as aviation, process control, telecommunications, electronic commerce and others have in common that not only functional, but also timely failures may have severe impacts on monetary and environmental scales, in the worst case even including the loss of human life. Clearly, such systems have to be designed as *fault tolerant*, i.e. in case of faults potentially catastrophic consequences have to be prevented by design. In this context, the development of systems with *redundancy* has a long tradition to achieve fault tolerance and thus reliability. The basic idea of redundant systems design is that specific components have redundant counterparts. In case of a fault, such a design ensures the overall correct timely and functional behaviour of a system. A non-trivial task in this context is to develop an architecture for a system which meets specific safety requirements. It is commonly agreed upon that fault tolerance measures should be considered as early as possible in the course of systems development. Otherwise, it is likely that integrating redundancy in later development stages is more expensive and, even worse, not as effective in comparison to earlier introduced measures [1,2]. The reason for this is that (partially) existing systems usually impose tight restrictions on the introduction of redundancy into the overall architecture in later development stages. In earlier development stages, on the other hand, it is a non-trivial task to design and comparatively evaluate alternative architectures for safety critical distributed real-time systems. Since potential problems and their consequences within a complex system cannot be easily predicted at early development stages, it is particularly difficult to decide which components of a system have to have redundant counterparts and which do not. The cause for this situation is that the increasing complexity of computer controlled systems and their dependability requirements in fact have exposed the limits of validation techniques traditionally used for safety and reliability analysis, like fault trees or Failure Mode and Effects Analysis [3]. Furthermore, there is barely any tool support for the evaluation of architectures in early design stages. Both leads to the following observation in Ref. [4]:

> "While manufacturers of [safety] critical systems are prepared and have a considerable experience on the validation of their *products*, there are major difficulties and much less experience in the early validation of system *design*."

E-mail address: stephan.philippi@uni-bielefeld.de (S. Philippi).

0951-8320/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0951-8320(03)00169-8

In order to help this situation, the high level analysis of real-time system architectures with respect to fault tolerance and reliability in the very early design stages is the subject of discourse in the remainder of this article. Even if functional aspects cannot be taken into account in early development stages, nevertheless simulation models suited for evaluation purposes can be built. Such models are based on the distributed topology of a system and provide the possibility to examine its architecture with respect to the consequences in case of single and multiple failures.

Starting from this perspective the structure of the article is as follows. Section 2 introduces the concept of *information horizon* as a basis for the analysis of fault tolerance and reliability in distributed real-time system architectures. In Section 3, a *brake-by-wire* system from the automotive industry is described, modeled and finally analyzed with a fault injection enabled simulation model. Section 4 concludes the presentation and describes further perspectives of the presented approach.

2. The information horizon in distributed systems

From a high level point of view, distributed systems consist of components and communication systems connecting them (Fig. 1). Each component of such a system sends and/or receives messages via one or more communication channels. Active components in distributed systems decide how to act on the basis of a combination of locally available data, e.g. from sensors, and data received from remote components, like supervisor nodes. The actual availability of relevant information at the right time for each component of a distributed real-time system is therefore an important necessary condition for the overall correct functionality. The main control unit of a brake-by-wire system in a car, for example, updates its internal model of the current vehicle and driving conditions with sensor data such as wheel turn angle, wheel rotation speeds, and others. As functions like Antilock Braking System (ABS) and Electronic Stability Program (ESP) are nowadays realized in software, it is obvious that the main control unit in a car, which includes these functions, operates most reliably, if all necessary information needed for the update of the internal model is available completely and in time.

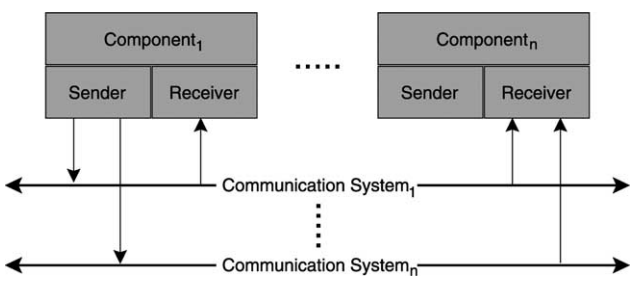

Fig. 1. A high level view on distributed systems.

In order to be able to measure the availability of information for each component in a distributed real-time system architecture, we introduce the concept of *information horizon* in the following. In a system without any faults the information horizon of each component is 100%, i.e. the data needed for optimal decision making in local components is available completely and in time. In case of loose cables, defective plug connections or partial failure of sensors, the information horizon of particular components is lower than this optimum. In fault tolerant systems, different message types may carry redundant information via different connections to the same component. Such message types form *redundancy groups*, which are equivalence classes from the point of view of the receiving component. This way, the loss of single messages can be compensated by redundantly transmitted ones. If a redundancy group as a whole fails to deliver messages within a given time frame, the information horizon of the distributed component is affected. In a brake-by-wire system, for instance, there are usually several redundant sensors which measure pedal braking force. The measured sensor data is distributed to specific components of such a system by means of different connections. The information horizon of a component which receives multiple sensor data in this scenario is affected only if braking force data is completely unavailable in a given time frame, i.e. the amount of received redundant messages via different communication systems is not important, as long as there is at least a single active connection left.

Since different kinds of information needed in subsystems are of different importance to the local decision making process, the equivalence classes transmitting these information types are weighted. Such a weight reflects the importance of a particular type of information for a specific component. The failure of a temperature sensor in a brake-by-wire system, for instance, may not be as significant for optimal decision making of the main control unit than the failure of one of the sensors for measuring wheel rotation speeds. If the information received by a local component from a redundancy group is less than was expected, the information horizon of this component is lowered by the weighted loss of information. The information horizon of each component in a distributed real-time system can therefore be determined with respect to a given time frame on the basis of weighted equivalence classes. The information horizon is formally defined as follows:

Definition 1. (Real-time system architecture): Let $A = (N, C, f_c, M)$ be a *distributed real-time system architecture* with:

- N the set of nodes,
- C the set of communication systems,