ELSEVIER

Knowledge-Based Systems 14 (2001) 385–396

Knowledge-Based SYSTEMS

www.elsevier.com/locate/knosys

Specifying fault tolerance in mission critical intelligent systems

Tolety Siva Perraju

Verizon Communications, MS35, 40 Sylvan Road, Waltham, MA 02451, USA

Received 23 August 2000; accepted 2 February 2001

Abstract

Real time intelligent systems are being increasingly used in mission critical applications in domains like military, aerospace, process control industry and medicine. Despite this vast potential, the major concern about deploying mission critical intelligent systems is their dependability. Dependability encompasses such notions as reliability, safety, security, maintainability and portability. A major concern about mission critical intelligent systems is their performance in the presence of failures. Intelligent systems are characterized by often non-existent, imprecise or rapidly changing specifications. This makes the task of characterizing an intelligent system's performance in the presence of failures much more difficult. In this paper, we characterize the failures that are likely in a mission critical intelligent system. We propose an extended I/O automata model to capture these failure specifications. We further demonstrate how these specifications can be realized in a real time expert system by structuring the knowledge base. This formalism can also be used to specify the fault tolerant properties of the underlying hardware and software over which the intelligent system resides. Thus we have an unified formalism to specify fault tolerance properties in hardware, system software and the intelligent system. This will enable us to reason about the performance of the entire system inclusive of all its components in an uniform manner. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Fault tolerance; Intelligent systems; I/O automata formalism; Mission critical systems; Real time expert systems

1. Introduction

Real time intelligent systems are being increasingly used for mission critical applications. The proliferation of mission critical applications leads to the challenge of making real time intelligent systems dependable. A dependable system is fault tolerant and provides high assurance to its users by maintaining high guaranteed levels of security, reliability, timing, availability, safety, and other attributes characterizing their operation. High Assurance Systems presume unambiguous specifications [4]. However, the development of intelligent systems is often characterized by non-existent, imprecise or rapidly changing specifications [30]. High Assurance in intelligent systems is yet to gain the attention it deserves. Some of the attempts at improving dependability of intelligent systems include Refs. [1,6–8,12]. The recent success of agent based systems and other AI systems in real world applications necessitates the need to focus on high assurance in intelligent systems. A first step towards developing high assurance intelligent systems is to have unambiguous specifications. A formalism for developing such specifications is in order. Considerable work is carried out in making software systems fault tolerant [2,11,13,28]. Formalisms developed to model the development of dependable systems focus on various design issues like specifying the faults to be handled, how to detect the existence of a fault, identifying the fault recovery methods, what happens during fault recovery, and the time constraints of the recovery process. However, this research is not easily extensible to the design of dependable intelligent systems.

In this paper, we first characterize the failures that are likely to occur in a mission critical intelligent system and then present a formalism, which will enable us to specify these requirements, in an unambiguous manner. To this end, we chose the I/O automata formalism [17]. I/O automata provide an appropriate and powerful model for discrete event systems consisting of concurrently operating components. The basic I/O automata model is extended to capture the notions of fault tolerance in mission critical intelligent systems. Once the specifications are realized in a design, it is necessary to prove that the design meets the requirements. Standard proof techniques reported in I/O automata literature are adaptable to the extended formalism. The extended I/O automata formalism can also be used to specify the fault tolerant properties of the underlying hardware and software over which the intelligent system operates [24]. Thus we have a unified formalism to specify fault tolerance properties in hardware, system software and the intelligent system.

E-mail address: toletys@acm.org (T.S. Perraju).

0950-7051/01/$ - see front matter © 2001 Elsevier Science B.V. All rights reserved.
PII: S0950-7051(01)00151-4

This will enable us to reason about the performance of the entire system inclusive of all its components in a uniform manner.

In Section 2, we characterize faults, failures and the associated recovery methods in high assurance intelligent systems. In Section 3, we present the I/O automata formalism, followed by an extension to specify fault tolerance. Section 4, shows how to specify failure detection and recovery in intelligent systems using the extended I/O automata formalism. In Section 5, we describe, how the specification of fault tolerant properties are realized by structuring the knowledge base of an intelligent system. Section 6 concludes with directions for future work.

2. Faults, failures and recovery

A failure in a system refers to its inability to perform some of its designated functions or to deliver the specified services. A fault is the adjudged reason of the failure. A fault is noticed when a corresponding failure is detected at the system boundary. Since, failures are the observable symptoms that need to be overcome in a high assurance system, we restrict our attention to failures.

2.1. Failures in intelligent systems

Intelligent systems are increasingly likely to encounter failures, as they are being deployed in mission critical environments. The likely sources of failures are:

- hardware and software faults in the underlying computational platform;
- failures induced by faults in the external environment and
- cognitive failures.

Hardware and software failures and pertinent fault tolerance approaches are dealt with extensively in fault tolerant systems research [14]. We assume the availability of a fault tolerant computing platform, on which the high assurance intelligent system will run[1].

Failures due to *external environment faults* occur when there is a deviation in the environment from its expected behavior[2]. These faults usually manifest as errors in the input data. For example, in a resistive circuit the voltage and current readings fail to follow the Ohm's law. This error could be either due a fault in the observed system itself (in this case the resistive circuit) or in the sensors employed to observe the target system (voltmeters and ammeters). The faults in the external world are inputs to the intelligent system. The system is expected to perform inference using these inputs and arrive at a conclusion.

The faults in the sensors employed to observe the target system are to be reckoned with, in making a mission critical intelligent system fault tolerant. Sensor faults cause failures, termed as *data integrity violations*[3]. Data integrity violations are transient failures. These failures are due to skew in the data. They are removed once the skew is corrected. Research is carried out in designing fault tolerant data fusion algorithms [5,19]. In these algorithms, data inputs from multiple sensors monitoring the same parameter are used to arrive at an estimate of the monitored parameter, with acceptable degree of accuracy. However, in mission critical systems, where space is usually at premium, multiple sensors are a luxury. The alternate strategy is *data corroboration*. Data corroboration is the process of validating an observed parameter value by using the readings of one or more related parameters. For example, in the resistive circuit, if the voltage reading is beyond the expected range, the accuracy of the voltage value can be estimated by taking the current reading and applying the Ohm's law[4].

Cognitive failures occur during the problem solving activity of the intelligent system. Cognitive failures occur either when the intelligent system is unable to meet a deadline or when it is unable to solve a given problem. The former are failures due to *deadline violations*, while the latter are *heuristic failures*.

Deadline violations occur either because the required action is taken after the deadline has passed or no action is taken by the intelligent system. These are usually transient in nature and appropriate recovery strategies can be applied.

A *heuristic failure* occurs when the proposed action is an inadequate or incorrect response to the input stimulus and the current state of the world. These failures also occur when actions taken by the intelligent system fail to have the desired effect on the environment. The former can be classified as heuristic failures due to *inadequacy in heuristics* employed and the latter as failures due to actions having *no desired effects* on the external world.

Heuristic failures can either be transient or permanent faults. Failures due to heuristic inadequacies are permanent failures. These could be either due to inadequate knowledge to solve the problem, or an ill suited inference strategy. Heuristics not having the desired effect on the environment, could be either due to:

- the action proposed by the system being inappropriate to the current state of the world or
- a change in the state of the external world between the time when the system sensed its environment and the time at which the system initiated an action in response.

[1] A computing platform whose behavior is well defined for a given set of fault hypothesis is a fault tolerant platform.

[2] The environment includes the world in which the system functions and any other system that may possibly interact with the intelligent system.

[3] Data integrity violations occur when the sensor readings do not correspond to the observed parameter's actual value but are skewed.

[4] In practice, data corroboration requires multiple parameters. Data corroboration with a single parameter value is a hazardous approach, decreasing the dependability of the system.

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
✓ امکان دانلود نسخه ترجمه شده مقالات
✓ پذیرش سفارش ترجمه تخصصی
✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
✓ دانلود فوری مقاله پس از پرداخت آنلاین
✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات