Journal of Systems Architecture 68 (2016) 38–50

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Runtime observation of functional safety properties in an automotive control network

Donal Heffernan*, Ciaran MacNamee

University of Limerick, Ireland

ARTICLE INFO

Article history:
Received 28 April 2015
Revised 25 January 2016
Accepted 7 May 2016
Available online 12 May 2016

Keywords:
Control networks
Runtime verification
Functional safety
Automotive
ISO 26262

ABSTRACT

This paper exploits the observability of control messages in a control network to formally monitor safety properties to verify a control application's correct behaviour. A monitor scheme is proposed based on a runtime verification method, which can verify selected properties of an application's behaviour, including the verification of formally specified functional safety properties. A prototype hardware based circuit is developed to provide a monitor function. A case study example for an automotive gearbox control system is presented. The control application is evaluated in the target application environment, which is a controller area network (CAN) based network. The behaviour of the monitor is assessed and the results show that it is feasible to monitor and verify functional safety properties, as defined by the ISO 26262 standard for functional safety in road vehicles, using the proposed method.

1. Introduction

Many control system designs are based on control network or sensor-level networks so that functionality can be distributed in space. Applications are found in industrial automation, transport vehicles, and many other fields. A large number of such small networks are employed in safety-critical applications and there are increasing demands from regulatory agencies and industries for product developers to ensure compliance with functional safety requirements.

Functional safety is an integrated part of the overall safety requirements for a product and is concerned with assuring that a system or equipment item operates correctly; taking into account the safe management of likely operator errors, system failures and environmentally induced problems.

This paper investigates the feasibility of monitoring safety property requirements for an application that runs in a control network based system, in real-time. The aim is to develop a specialised monitor device, which is a programmable hardware based circuit that observes the behaviour of an application in real time and reports on violations. The work suggests that conventional control network architectures lend themselves to the development of runtime monitoring schemes in a useful way where the various control messages are exposed on the network and the set of these messages can be interpreted in formal logic equations to inform an observer about the behaviour of some key properties. It is specifically proposed that such a monitor can be used to observe functional safety properties.

A case study example illustrates how such a monitor can be developed for an automotive application which is based on the controller area network (CAN) bus [1]. The case study example is based on a software controlled automotive gearbox, which can be seen as an important equipment item for safety evaluation.

Two distinct use cases are evaluated for the monitor employment. The first use case is where the development engineer carries out the verification process as part of the product development cycle, using the runtime monitor as a development tool. The second use case is where the runtime monitor is permanently installed in the system and is used to verify the correct behaviour of an application, through the lifetime of the product.

Based on the case study example it is demonstrated that it is feasible to verify functional safety properties for a product using such runtime verification. However, the value of using such a monitor for lifetime monitoring within a product has questionable benefits.

The remainder of this paper is organised as follows: Section 2 describes some background for functional safety and runtime monitoring; Section 3 summarises related research work; Section 4 introduces the gear controller case study and its requirements; Section 5 describes the case study implementation; Section 6 discussed the evaluation and testing for the concept; and Section 7 summarises the conclusions.

* Corresponding author.
E-mail addresses: donal.heffernan@ul.ie (D. Heffernan), ciaran.macnamee@ul.ie (C. MacNamee).

http://dx.doi.org/10.1016/j.sysarc.2016.05.001

2. Background

Runtime monitoring for checking the performance behaviour of embedded systems' applications is an established field. In this study the concept is extended to the monitoring of functional safety properties.

2.1. Functional safety

Functional Safety is achieved by product developers by designing and developing products to ensure all specified safety functions are implemented; and the level of performance required of each safety function is met. This is usually achieved by a development process that involves: identifying the formally required safety functions, based on risk assessment and management; assessment of the risk-reduction required by the safety function; ensuring the safety function performs to the design intent; and verifying that the system meets the assigned safety integrity levels. Then there is a requirement to conduct Functional Safety audits to assess evidence that the appropriate safety lifecycle management techniques are being applied consistently in the relevant lifecycle stages of a product.

2.2. The functional safety standards

Probably the best known functional safety standard in the electronics industry is the IEC 61508 functional safety standard and its automotive adaptation ISO 26262 [2]. The IEC 61508 document defines four general Safety Integrity Levels (SILs), where SIL 4 represents the most stringent safety level. The ISO 26262 document defines four Automotive Safety Integrity Levels (ASILs) where ASIL D is the most stringent safety level. Each level corresponds to the likelihood of failures for a safety function.

The ISO 26262 does not specify any development processes or technologies. Rather it assumes that development processes such as ISO 9001, CMM or similar process are already in place and imposes specific safety related requirements and outcomes on them.

2.3. Runtime monitoring

In this paper it is proposed that an independent monitor can be developed to monitor key safety properties during product runtime to confirm proper behaviour of an application during its execution phase. Such a runtime verification monitor is concerned with the monitoring of program execution behaviour so as to establish compliance with a requirements specification. The properties to be observed can be decided by the developers to gain confidence in the correct operation of the product or system.

The concept of the proposed runtime monitor makes the following key assumptions:

(a) The runtime environment on its own cannot detect violations of an application's safety properties. Traditional means of node diagnostics and network diagnostics cannot have awareness of an application's specifications.

(b) It is not sufficient to verify the behaviour of functional safety properties outside of the actual runtime environment. The verification of properties outside of the runtime context cannot make realistic assumptions on how the implementation environment can impact on the application's behaviour in terms of functionality and timing.

(c) The evaluation of the safety properties will not lead to erroneous conclusions resulting in false negatives. This is assuming the monitor itself does not fail and all runtime parameters that can impact on the verification are accounted for.

(d) The target system will expose sufficient global variables to the network so that the monitor logic can be meaningfully expressed by observing the exported messages.

(e) The assertion check process is sufficiently fast so that verification can be achieved in real-time while the system is executing.

In the proposed scheme the requirement specification of a system is stated as an executable specification that describes the behaviour of the system. That described system model is formally verified using a model-checker that can verify timed behaviour. Program code is generated from the specification model for product implementation. At runtime, selected properties that have already been formally verified in the model checker are evaluated. This evaluation during runtime is a form of verification that is based on assertion testing [3].

3. Related work

A runtime verification monitor is concerned with the monitoring of program execution behaviour to establish compliance with a requirements specification. The concept of monitoring system behaviour during runtime is well established as a means to employ a 'lightweight' formal verification method to assess runtime compliance behaviour in accordance with a product's requirement specifications. Runtime monitoring methods have been proposed by Havelund and Roşu [4], Drusinsky [5], Havelund et al. [6], and Sammapun [7], amongst others. The Monitoring and Checking (MaC) framework, by Lee et al [8] and Kim et al [9], proposes a scheme to automatically link low-level observations of program execution behaviour to the relevant monitored properties. PathExplorer (PAX) is a runtime verification tool by Havelund and Roşu [10] that uses linear temporal logic (LTL). Watterson et al. [11] provide an in-depth review in the context of the requirements for monitoring in embedded systems. Michael et al. [12] describe runtime execution monitoring schemes to assess whether formal assertions correctly capture the intent of some natural language requirements.

Many of the proposed methods to date are focused on solutions that perform on-line monitoring, and employ off-line processing of captured traces from the monitoring exercises. In the work described in this paper, a monitor device is proposed that can be embedded right into the product. The need for trace memory can be eliminated by using state by state evaluations during runtime.

Interest in evaluation for safety conformance in products spans many engineering disciplines as described by Saleh et al [13]. In recent times there is growing emphasis on the development of software related safety cases, as discussed by Hawkins et al. [14]. However, the concept of mapping specific safety requirements from a safety standard to runtime monitored properties is not well explored in the research literature. Clauses 7 and 8 of ISO 26262-6, which refer to software architectural design and software unit design respectively, each point to external monitoring techniques as a means of attaining ISO 26262 compliance, while the test case derivations covered by clauses 9 (Software Unit Testing) and 10 (Software integration and testing) lend themselves to a monitoring approach.

In automotive systems the need for behaviour monitoring is well understood, but solutions to the problems are incomplete. One significant outcome from research on these issues is the development of the AUTOSAR (AUTomotive Open System ARchitecture) [15] standard. AUTOSAR is an open, standardised software architecture for automotive E/E (Electrics/Electronics) systems; providing an infrastructure to assist with the development of in-vehicle software for the entire product lifecycle. Although the AUTOSAR recommendations are open to the inclusion of software monitoring, they do not suggest any formal approach. Lotoczky et al. [16]