Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Department of Measurement and Information Systems

Hierarchical runtime verification for critical cyber-physical systems

Scientific Students' Associations Report

Authors:

László Balogh
Flórián Dée

Supervisors:

dr. István Ráth
dr. Dániel Varró
András Vörös

2015.

Contents

Contents iii

Abstract v

1 Introduction 1

2 Background 3

3 Overview 5

4 Runtime verification of embedded systems 7

5 Complex event processing 9
5.1 Formal Intro of the Event Automatons 9
5.1.1 Current Formalisms 9
5.1.2 Our Formalism 9
5.2 Examples of Event Processing 10
5.2.1 File System 10
5.2.2 Mars Rover Tasking 10
5.3 Implementation 10
5.3.1 Metamodel 10
5.3.2 Executor 10

6 Case study 11
6.1 Overview 11
6.2 Architecture 12
6.2.1 Total view 12
6.2.2 Hardware 12
6.3 Concept 12
6.4 Computer vision as a source of information 12
6.4.1 Hardware 12

6.4.2 Computer vision 13
6.4.3 Introducing to OpenCV 13
6.4.4 Marker design 13
6.4.5 Software 14
6.5 Physical - logical mapping 15
6.5.1 Elements of physical mapping 15
6.5.2 Elements of logical mapping 15
6.5.3 Introducing to EMF 15
6.5.4 Building the EMF model 15
6.5.5 Introducing the IncQuery 15
6.5.6 Building the IncQuery patterns 15

7 Conclusion **17**

References **19**

Összefoglalás Ipari becslések szerint 2020-ra 50 milliárdra nő a különféle okoseszközök száma, amelyek egymással és velünk kommunikálva komplex rendszert alkotnak a világhálón. A szinte korlátlan kapacitású számítási felhőbe azonban az egyszerű szenzorok és mobiltelefonok mellett azok a kritikus beágyazott rendszerek – autók, repülőgépek, gyógyászati berendezések - is bekapcsolódnak, amelyek működésén emberéletek múlnak. A kiberfizikai rendszerek radikálisan új lehetőségeket teremtenek: az egymással kommunikáló autók baleseteket előzhetnek meg, az intelligens épületek energiafogyasztása csökken.

A hagyományos kritikus beágyazott rendszerekben gyakorta alkalmazott módszer a futási idejű ellenőrzés. Ennek célja olyan ellenőrző programok szintézise, melyek segítségével felderíthető egy kritikus komponens hibás, a követelményektől eltérő viselkedése a rendszer működése közben.

Kiberfizikai rendszerekben a rendelkezésre álló számítási felhő adatfeldolgozó kapacitása, illetve a különféle szenzorok és beavatkozók lehetővé teszik, hogy több, egymásra hierarchikusan épülő, különböző megbízhatóságú és felelősségű ellenőrzési kört is megvalósíthassunk. Ennek értelmében a hagyományos, kritikus komponensek nagy megbízhatóságú monitorai lokális felelősségi körben működhetnek. Mindezek fölé (független és globális szenzoradatokra építve) olyan rendszerszintű monitorok is megalkothatók, amelyek ugyan kevésbé megbízhatóak, de a rendszerszintű hibát prediktíven, korábbi fázisban detektálhatják.

A TDK dolgozatban egy ilyen hierarchikus futási idejű ellenőrzést támogató, matematikailag precíz keretrendszert dolgoztunk ki, amely támogatja (1) a kritikus komponensek monitorainak automatikus szintézisét egy magasszintű állapotgép alapú formalizmusból kiindulva, (2) valamint rendszerszintű hierarchikus monitorok létrehozását komplex eseményfeldolgozás segítségével. A dolgozat eredményeit modellvasutak monitorozásának (valós terepasztalon is megvalósított) esettanulmányán keresztül demonstráljuk, amely többszintű ellenőrzés segítségével képes elkerülni a vonatok összeütközését.

Abstract According to industrial estimates, the number of various smart devices - communicating with either us or each other - will raise to 50 billion, forming one of the most complex systems on the world wide web. This network of nearly unlimited computing power will not only consist of simple sensors and mobile phones, but also cars, airplanes, and medical devices on which lives depend upon. Cyber-physical systems open up radically new opportunities: accidents can be avoided by cars communicating with each other, and the energy consumption of smart buildings can be drastically lower, just to name a few.

The traditional critical embedded systems often use runtime verification with the goal of synthesizing monitoring programs to discover faulty components, whose behaviour differ from that of the specification.

The computing and data-processing capabilities of cyber-physical systems, coupled with their sensors and actuators make it possible to create a hierarchical, layered structure of high-reliability monitoring components with various responsibilities. The traditional critical components' high-reliability monitors' responsibilities can be limited to a local scope. This allows the creation of system-level monitors based on independent and global sensory data. These monitors are less reliable, but can predict errors in earlier stages.

This paper describes a hierarchical, mathematically precise, runtime verification framework which supports (1) the critical components' monitors automatic synthetisation from a high-level statechart formalism, (2) as well as the creation of hierarchical, system-level monitors based on complex event-processing. The results are presented as a case study of the monitoring system of a model railway track, where collisions are avoided by using multi-level runtime verification.

Chapter 1

Introduction

Chapter

Chapter 2

Background

Chapter

Chapter 3

Overview

Chapter

Chapter 4

Runtime verification of embedded systems

Chapter

Chapter 5

Complex event processing

Chapter

5.1 Formal Intro of the Event Automatons

5.1.1 Current Formalisms

Quantified Event Automaton

Calendar Event Automaton

5.1.2 Our Formalism

Ezeknek nyilvan fancybb nev kell, csak errol fog szolni.

Things which are the same as in one of the previous formalisms

Things which are not the same (changed?)

Nondeterminism -> Deterministic behaviour

Timing

5.2 Examples of Event Processing

5.2.1 File System

5.2.2 Mars Rover Tasking

5.3 Implementation

5.3.1 Metamodel

5.3.2 Executor

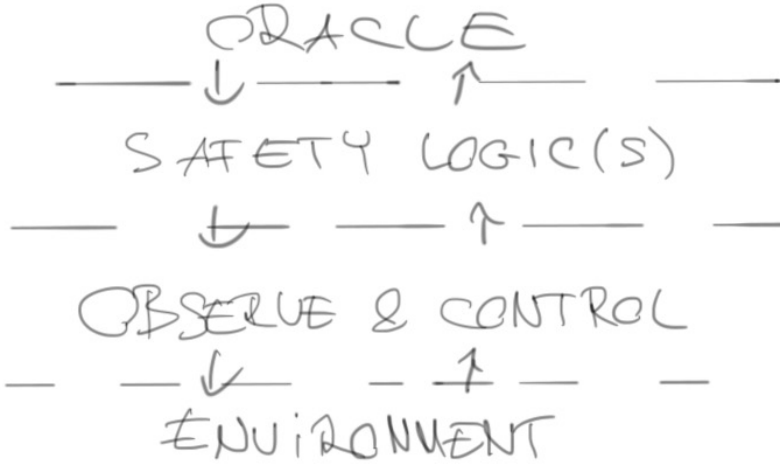
Chapter 6

Case study

6.1 Overview

The goal of this chapter is to present a case study of a hierarchical runtime verification technology based on the previous chapters. The motive of this study is the related report from 2014 [1], which goal was a distributed, model based security logic. Their case study called the *Model Railway Project*, and our study integrates their work, and finds ways to integrate it with other sensor sources for a hierarchical, more reliable security logic.

Because of this close relation to this railroad project, our focus will stay on railroad technologies and standards, it's important to notice our solution is a general approach for any critical system. This approach is based on controllability, observability, and hierarchy to increase a system general reliability, even when some of it's components are failing.



6.2 Architecture

6.2.1 Total view

Itt átvesszük a hardware alapjait, hogy egyáltalán miről van szó, hogy néz ki, mit tud.

6.2.2 Hardware

Itt az arduino alapú vezérlést emelném ki röviden, az ezzel felmerült problémákat, illetve a jövőbeli fejlesztések rövid bemutatását.

6.3 Concept

A koncepció magyarázata, szép összefoglaló ábrával, és annak az összefoglalása, hogy mit tudunk ezzel elérni.

6.4 Computer vision as a source of information

6.4.1 Hardware

In case of a computer vision (CV) based approach, it is critical to choose the appropriate hardware. We had two parameters in the selection of the camera: height above the board, and FOV.

Definition 6.1 Field of View (FOV) is the extent of the observable world that is seen at any given moment. See figure Section 6.4.1 for visual explanation.

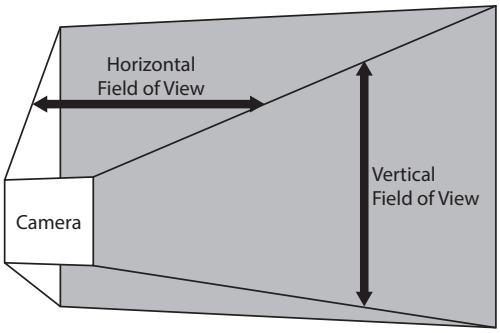


Figure 6.1 Visual explanation of FOV

These parameters are coupled, the higher the camera the less FOV we need. Most of the cameras on the market have horizontal FOV values approximately 60°.

Example 6.1 The board is 2.8 m wide. So if we assume we have a camera with a 60° FOV, using the result of eq. (6.1), we need to place the camera 242 centimeters high. This would be is impossible to realize with average ceiling heights.

$$h = \frac{140}{tg(30)} \tag{6.1}$$

Our chosen FOV became 120°, because the cameras with these FOV values are inexpensive, and easy to find. With a placement height of 120 cm, we can fully assemble the project virtually in any room.

6.4.2 Computer vision

One key point of this study from technological viewpoint is computer vision. It is a non intrusive add-on to the existing hardware, which allows us to monitor the board with fairly big precision and reliability, if the correct techniques and materials are used.

6.4.3 Introducing to OpenCV

We needed a fast, reliable, efficient library to use with the camera, and develop the detection algorithm. Our choose was the OpenCV[1] library, which is an industry leading, open source computer vision library. It implements various algorithms with effective implementation in mind e.g., using the latest streaming vector instruction sets. The main programming language – and what we used – is C++, but it has many binding to other popular languages like Java, and Python.

6.4.4 Marker design

One of the steps of the CV implementation was the design of the markers, which should provide an easy detection, and identification of the marked objects.

The first step was to consider the usage of an external library, named ArUco[2]. This library provides the generation (see Section 6.4.4), and detection library of markers.

The problem with the library was the lack of tolerance in quality, and motion blur. Other limiting factor is the shape of the marker. A square marker scaled up to provide good visibility for the camera 120 cm above the board extends greatly over the width of a

[1] http://opencv.org/
[2] http://www.uco.es/investiga/grupos/ava/node/26

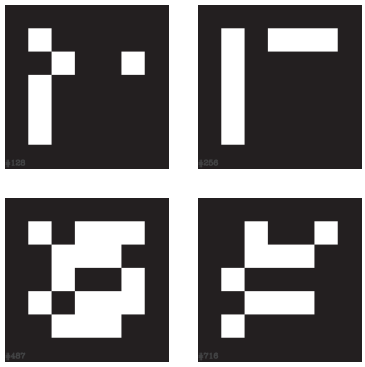


Figure 6.2 Example markers generated by ArUco

railway car. Because these negative properties of the existing libraries, we implemented a marker detection algorithm for our needs.

After the implementation was in our hands, we could make markers which suits our needs. The optimal marker covers the railroad car. This means the marker is narrow and long to fill the top dimensions of the car, but not exceed it.

As explained in Section 6.4.5, circular patterns are well suited for these applications. The final design consists two detection circle, and a color circle for identification between the detection circles.

Figure 6.3 The final marker design

6.4.5 Software

Mathematical solution

A matematikai alap bemutatása, itt főleg a frekvenciatartomány beli mintakeresést kiemelve.

CPU implementation

Az első implementáció gyors bemutatása.

CUDA implementation

A GPU által gyorsított verzió bemutatása, mit tapasztaltunk ennek során, hogyan segített a fejlesztésben.

6.5 Physical - logical mapping

6.5.1 Elements of physical mapping

Az általunk használható architektúra részletes bemutatása.

6.5.2 Elements of logical mapping

Azoknak az elemeknek, elképzeléseknek az áttekintése, amik szerepelhetnek a modellünkben, még teljesen független módon.

6.5.3 Introducing to EMF

Az EMF bemutatása röviden.

6.5.4 Building the EMF model

Az elkészült EMF metamodell bemutatása, és összevetése az elképzelésekkel.

6.5.5 Introducing the IncQuery

Az IncQuery bemutatása.

6.5.6 Building the IncQuery patterns

A biztonsági lokikai patternek bemutatása.

Chapter 7

Conclusion

Chapter

References

[1] Mázló Zsolt Horváth Benedek Konnerth Raimund-Andreas. *Elosztott biztonságkritikus rendszerek modellvezérelt fejlesztése*. Tech. rep. Budapest University of Technology et al., 2014.