POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

FORMAL METHODS IN ACTION:
using runtime verification to automatically address occurrences of labor exploitation crimes

Laurea Magistrale in Computer Science and Engineering

Author: Federica Suriano

Advisor: Prof. Marcello Maria Bersani

Co-advisor: Prof. Damian Andrew Tamburri

Academic year: 2022-2023

---

1. Introduction

Nowadays, labor exploitation is enormously widespread throughout Europe and beyond. Already in 2012 according to the Global Estimate of Forced Labor presented in a Regional Factsheet European Union by the International Labor Organization (ILO), it was estimated that there were 880,000 people in forced labour, i.e. 1.8 per 1000 inhabitants of the member states of the European Union.
The factors that most push people to submit to such abuses are **poverty**, **lack of knowledge** of labor laws and the **lack of institutions** that effectively monitor the situation of workers in sectors of the economy where labor exploitation occurs [2].
Even where there are sufficient institutions these are not enough efficient and therefore this is perceived as an absence of adequate monitoring of crimes relating to exploitation. Added to this is the **absence of good enough software** systems specifically created to monitor these phenomena.
For this reason, ***SENTINEL*** was born: it is a system that aims to monitor possible cases of labor exploitation in the North Braabant region, in the Netherlands.
Our goal is to make it possible to **automatically verify the occurrence of crimes**, in particular crimes related to labor exploitation. This work was carried out starting from the analysis of SENTINEL system.
Providing the results of this type of analysis to state bodies, governmental and beyond, that operate in contexts of this type and deal with justice, laws and fundamental rights of human beings is of extreme importance and can positively influence an evolution in how, too often, these inequities are managed.

2. Methodologies and tools

The verification of the monitored events by the SENTINEL system occurred using the **runtime verification** (RV), a method used to **verify system properties at runtime**. To carry out the verification, it is necessary to have a trace of events on which to verify, which represent the behavior to be monitored, and the properties to be verified translated into a formal language, for example in temporal logic. The choice of properties to verify varies based on what the goal of

the verification is. Typically, the RV method is used in critical systems to verify properties such as robustness, safety and security.

2.1. SENTINEL extention to save logs

In order to carry out the verification, it was necessary to **extend SENTINEL system** to integrate the **logs** representing system events, used to allow the identification of the crime regarding labor exploitation at runtime. We saved the entire situation reported in a SENTINEL report on a single log entry. This means saving much of data entered by the user of the system regarding a report of possible victimization. The data that we have decided to keep in the log files are data concerning the **position** in which the possible victimization is taking place and the data concerning the **victimization**, i.e. specifically what type of victimization is taking place and what the affected categories are. A log file generated by the extendede system will contain one or more entries of the type:

<Timestamp> ***situation(****int idReport, int idAddress, String city, String province, boolean housingExpl, boolean healthExpl, boolean paymentExpl, boolean laborExpl, boolean employerExpl, boolean householdsExceeded, boolean workingHoursExceeded, boolean paymentExceeded, boolean noContract, boolean noVacation****)****.*

All the parameters contained in the log entry represent a particular category o sub-category of victimization.

2.2. Identification of the properties

In our analysis and based on the context in which SENTINEL operates, we focused on identifying properties relating to the counting of specific criminal events, the time range in which they occur, the security relating to certain events, the compliance of recorded events with respect to Dutch Law and GDPR privacy regulations.
The specification language used to translate properties in logic formulae is **Metric First Order Temporal Logic** (MFOTL), which is particularly suitable for verifying the previously chosen properties.

The tool we used for SENTINEL runtime verification is **MonPoly** [1]: it is a monitoring tool for checking the compliance of systems to properties, which identifies and reports all property violations during program execution. This means it can spot violations in real time, providing valuable insights into the system's behavior while it's running.
Below we report an example of a property analyzed and translated into MFOTL.

Property 1: Number of reports entered in total in the last 3 months > 100
Property 1 formalized in MFOTL:
$\forall$c $[CNT_{idR\ [0,3m]}$ situation(...)$]$ (c) $\rightarrow$ c ≤ 100

The property formalized in MFOTL (in the second row of the example) is the negation of the related property (in the previous row). This is a trick that we needed to be able to use the formulas within MonPoly, where they will then be run negated again.

3. Results

3.1. Simulator

Before the extension we did on the system through which it is now possible to log its events information, we did not have log files available and therefore we **simulated** the creation of the latter to carry out the verification. Obviously the system and consequently the creation of logs also work without a simulator. However, creating a large amount of reports via SENTINEL would have required running the application in the real world for months and would therefore have been a slower process of obtaining the logs to verify. This is the reason why we decided to implement a simulator that could independently creating a series of events recorded by the system.

3.2. Properties verification

The analysis of the results obtained showed that:

- runtime verification is suitable and useful for analyzing properties regarding the **granularity** of the data and the **periodicity** with which certain events occur in a certain context, which can be, for example, a certain geographical area;
- through runtime verification it is possible

to **track the occurrences of labor exploitation crimes at runtime**;
- through runtime verification it is also possible to verify that there are **no errors** in some parts of the event processing, which would therefore lead to the discovery of system bugs. For example, we could create a property that verify that a constraint is actually respected. Specifically in SENTINEL, we could verify that there is no event having a parameter of a subcategory set to true and having at the same time the parameter of the corresponding category set to false. In SENTINEL case, no bugs have been found so far;
- runtime verification can be used to verify the **absence of certain values**, for example we could verify that the logs do not contain personal data, by creating a property that verify that the fields relating to the situation do not contain values other than true and false.

3.3. Time and space data

To evaluate the time and space required to verify a policy thanks to MonPoly we have created - with the simulator presented previously - other simulated logs containing **100, 1'000, 10'000, 100'000, 1'000'000, 10'000'000, 100'000'000 log entries**. We will refer to them respectively with the names **L1, L2, L3, L4, L5, L6, L7**.
In Table 1 and 2 results are presented.

Even on simulations containing a huge number of log entries, memory used and time required to verify a policy are promising.

Log	Space
L1	10532 KB
L2	10532 KB
L3	12484 KB
L4	21812 KB
L5	21812 KB
L6	23424 KB
L7	21994 KB

Table 1: Space required for verification.

Log	Real Time	User Time	Sys Time
L1	0.005s	0.004s	0.000s
L2	0.050s	0.030s	0.021s
L3	0.554s	0.336s	0.218s
L4	5.385s	2.857s	2.516s
L5	55.785s	29.037s	23.666s
L6	9m 34.474s	4m 47.420s	3m 55.997s
L7	92m7.794s	48m51.107s	43m6.226s

Table 2: Time required for verification.

4. Conclusions

This research made it possible to analyze the role that runtime verification can have within complex systems in which the **correctness and coherence of the assimilated data** is fundamental and the possibility of tracking it is extremely useful.
In a fast-paced world, having access to the data extrapolated in real-time thanks to runtime verification could be very beneficial for better managing the actions to be taken in the immediate future to preserve the system and make the best use of it. In the context of SENTINEL, this research has demonstrated how it is possible to apply this verification method to a legal related system not only to verify its basic properties, but also to **evaluate the occurrences of events** that are recorded by it, i.e. the succession of possible **crimes**. This allows, in the specific case of SENTINEL Monitor, to make the necessary considerations on the system and also on the criminal events monitored in real time, to take **strategic decisions** to hopefully drastically **reduce the phenomenon of labor exploitation**.

5. Final considerations

The verification of SENTINEL events is carried out taking into account the **laws of the Netherlands**, but could also be extended to all member states of the European Union in the future. A single system that offers an overall vision would make it possible to identify the differences between the various states in terms of labor exploitation and would allow us to emphasize the differences between neighboring states. Having a general picture of the situation would allow us to understand the phenomena underlying this

problem in the Netherlands, but also in other European Union countries. A big step forward would be to create a free and accessible platform that declares the average working conditions in a given sector and in a given region in order to make people more aware of what they deserve based on their skills and put them in a position to claim it.

6. Acknowledgements

I would like to express special thanks to Prof. Marcello Maria Bersani and Prof. Damian Andrew Tamburri for the constant support, professionalism and precious advice they gave me for the realization of this work. It was a pleasure collaborating with you.

A thanks also goes to Politecnico di Milano and to all the people who work with dedication to make it a place of professional and personal growth.

References

[1] David Basin, Felix Klaedtke, and Eugen Zlinescu. The monpoly monitoring tool. *Kalpa Publications in Computing*, 2017.

[2] Marije Braakman, Saskia Van Bon, Gregor Walz, and Igor Boog. Social fieldwork research (franet) severe forms of labour exploitation supporting victims of severe forms of labour exploitation in having access to justice in eu member states. 2014.