This electronic thesis or dissertation has been downloaded from the King's Research Portal at https://kclpure.kcl.ac.uk/portal/

# Empolying process mining for RBAC analysis

Alrahili, Rawan

*Awarding institution:*
King's College London

**The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without proper acknowledgement.**

END USER LICENCE AGREEMENT

**Unless another licence is stated on the immediately following page** this work is licensed

under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International

licence. https://creativecommons.org/licenses/by-nc-nd/4.0/

You are free to copy, distribute and transmit the work

Under the following conditions:

- Attribution: You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work).
- Non Commercial: You may not use this work for commercial purposes.
- No Derivative Works - You may not alter, transform, or build upon this work.

Any of these conditions can be waived if you receive permission from the author. Your fair dealings and other rights are in no way affected by the above.

**Take down policy**

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 14. May. 2025


# Employing process mining for RBAC analysis

By
Rawan Alrahili

A thesis submitted in fulfilment for the degree of Doctor of Philosophy

in the
Department of Informatics
School of Natural & Mathematical Sciences
King's College London

2023

# Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

# Acknowledgements

This PhD has been a long, and definitely not an easy journey. There have been many obstacles and setbacks, however, stopping these studies has never been an option.

This thesis, would not have been possible without the support and encouragement of those around me. First of all, I would like to express my sincere gratitude to my supervisors *Prof. Luca Viganò* and *Prof. Jose Such*. I am especially grateful to *Prof. Luca Viganò* for his guidance and feedback. I will always appreciate your kindness and support through this long journey. "Thank you".

I would like to thank my father *Mohammed*, and my mother *Maryam* for their generous love and support. You taught me to strive to succeed, and that quitting is not an option. To my lovely brothers *Nader, Mohanned, Ahmed,* and *Abdullah* for their encouragements and believing in me. To my sisters *Noora* and *Sara*, my best friends, for their positivity and endless love. I must be forever grateful for having this amazing family.

My biggest thanks to my husband *Jihad* for all the support he has shown me through this research. I simply could not have done this without you, special thanks. To my two kids *Jodi* and *Omar*, for sharing their positive energy with me, and for their patience whilst I wrote this thesis. I hope that I showed you when life becomes uneasy, we need to push the boundaries and work twice harder to achieve our goals. I love you the most, and I will always be by your side.

# Abstract

Organisations have expectations of how their systems should work, called *modelled behaviour*. However, in reality this is not usually the case. Considering flaws in modelling systems and misbehaviour from the human side result in what is called *actual behaviour*. Organisations typically enforce restrictions on the modelled behaviour. Therefore, checking these restrictions while assuming a system is working as expected, may provide inaccurate results or undetected violations. By relating actual and modelled behaviour, checking unleashes the maximum potential of gaining accurate and precise results.

This thesis focuses on the analysis of security, particularly *role-based access control (RBAC)*, using *process mining*, which allows linking the actual and modelled behaviour by following an interactive methodology to design this research. Thus, the methods used in this research are designed to answer the research questions, which aim to understand the current state of the art, find limitations, uncover challenges, and identify opportunities for improvement.

My thesis consists of four main parts. First, I conduct a systematic literature review in Chapter 3 to better understand the research landscape. Then, in Chapter 4, I provide an exploratory case study to seek new insights for the research, while in Chapter 5, I introduce a new multi-perspective approach to improve the conformance checking of RBAC.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# INTRODUCTION

## 1.1 Problem Statement and Motivation

Organisations such as hospitals, universities and businesses have specific expectations for how their systems should work. Since they now rely heavily on digital systems in almost every aspect of their operations, they use process modelling methods to design the behaviour needed from them. Moreover, they install computer programs to regulate their systems and digital security mechanisms to enforce and control the necessary restrictions. As a result, the functionality of these organisations is two-sided. *Modelled behaviour* is represented by the digital programs that draw the expectations, whereas *actual behaviour* is shown by the human users running these programs. Therefore, inaccurate results may be given when checking the enforced security restrictions under the assumption that the system is following the design and working as expected and in isolation of the actual behaviour.

Security is now one of the hottest research topics for individuals and organisations. Securing access and controlling authorisation is crucial for organisations that use many different systems and resources. Access control is necessary to provide secure access to

the available resources. Role-based access control (RBAC), a policy-neutral mechanism defined according to roles and privileges, has become a de facto standard for access control in today's systems [6]. The basic idea of RBAC is that users are assigned roles, which are in turn given permissions [7]. In this way, users obtain permissions according to their role. Understandably, removing, adding or modifying permissions becomes easier when dealing with roles, rather than individual users. In core RBAC, the user and permission role assignments can be many to many. Hence, the same user can be assigned many roles, while a single role can have many users. Likewise, a single permission can be assigned to many roles, and a single role can be assigned to many permissions. Due to its simplicity and functionality, RBAC has become increasingly popular among organisations [8].

Organisations employ policies to constrain access to their different resources. The implementation of access control manages the decision of granting or denying authorisations. Therefore, it is important that the access control policies are implemented correctly and followed within systems.

There has been a considerable amount of research on analysing access control [9, 10, 11]. However, one major limitation of traditional methods is that they are analysed in isolation from the real execution of the software system. Thus, access control policies are analysed under the assumption that an organisation's systems are working as expected. For instance, by using the model checking method [12], an access control policy is verified and validated without giving consideration to the modelling and execution of the target software system. Moreover, since organisations often use different event logs for their systems and the resources they protect, these traditional methods do not exploit these recorded data in the analysis. Therefore, the relationship between the modelled behaviour (access control policies) and the actual behaviour (real execution of the software system) is missing. Thus, this thesis studies the analysis of security, partic-

ularly *RBAC*, using *process mining*, a method that enables the actual and the modelled behaviour to be linked.

Process mining is an emerging discipline between machine learning and data mining on one side and process modelling and analysis on the other [3]. It provides methods that can extract knowledge from event logs available in information systems and use it to discover, analyse and improve an organisation's current systems of information [3, 13, 14, 15, 16]. More specifically, given an event log and a security constraint, a security analyst can verify whether the constraint holds by extracting from the event log information about what is happening in the system and analysing it to investigate the security state [17]. Research on process mining began in the late 1990s [18]. Over time, it has become a common topic in academia and industry. Figure 1.1 shows the number of articles published in the field of process mining between 2001 and 2022. What stands out is the phenomenal growth in the number of publications.

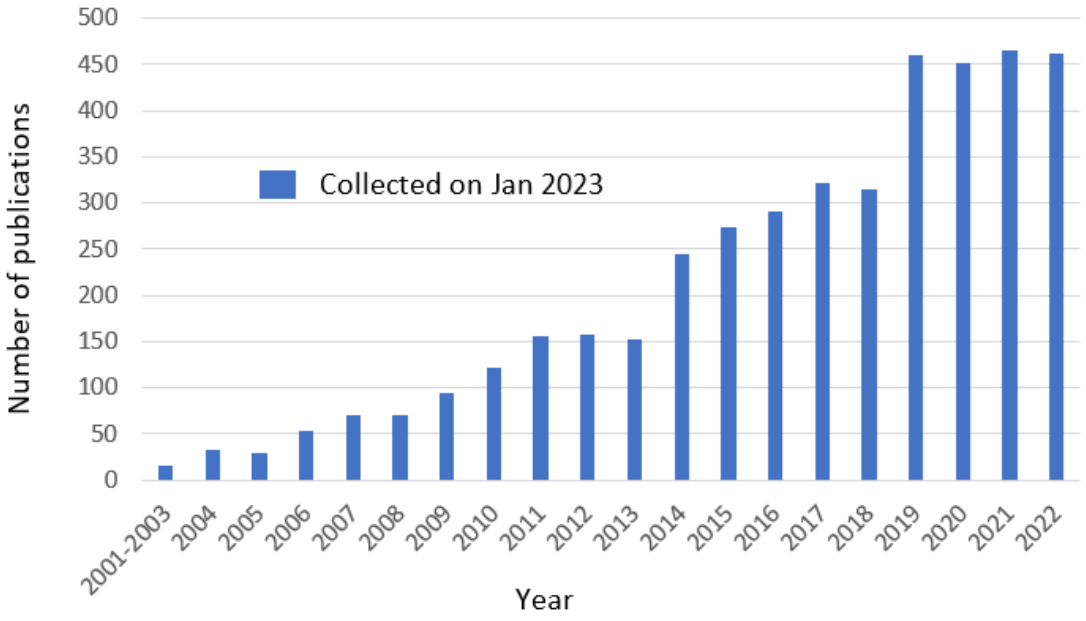



Figure 1.1: The number of article published, using the term (allintitle: "process mining") in Google scholar

Process mining is a hot sector right now. According to recent research by Deloitte

[19], globally, 63% of companies have started using process mining, and plan to pilot process mining soon. Figure 1.2 shows the top process mining software in 2022 [1]. These were Workfellow [20] and Celonis [21], which raised over $2.3 billion in investment funding in August 2022 from the Qatar Wealth Fund; UiPath [22]; Signavio [22], which SAP acquired in 2021; IBM Process Mining [23]; and ProM (Process Mining Framework) [2]. ProM is popular in academia because it is an open-source framework that users and developers can leverage when building process mining capabilities. The first version of ProM was released in 2004 with 29 plug-ins. Over time, the platform has been extended, and it now includes over 1500 plug-ins [24].


Figure 1.2: Top process mining software in 2022 [1].

Aalst [24] stated that the primary strength of process mining is that it is generic and can be applied in any organisation. Therefore, it is interesting to use process mining for RBAC analysis, as it gives an accurate picture of the application of RBAC policies on organisations' systems. However, there has been too little attention given so far to the analysis of access control in organisations based on recorded event data. Because of this, we have limited understanding of the characteristics, capabilities and challenges of

using process mining for RBAC analysis.

## 1.2 Goals, Research Questions, and Objectives

The goals, research questions and corresponding objectives defined for this investigation are shown in Table 1.1.

| **Goals** | **Research questions** | **Objectives** |
|---|---|---|
| Goal1. To investigate the analysis of RBAC properties on organisations by using process mining methods. | RQ1. How is RBAC analysed using process mining methods? | 1. To identify suitable process mining methods for RBAC analysis.<br>2. To identify the RBAC constraints that are discussed in the literature by using process mining.<br>3. To clarify the process of RBAC analysis by using different process mining methods. |
| | RQ2. To what extent can existing process mining approaches be used in RBAC analysis? | 4. To determine the challenges when using process mining on complex RBAC constraints.<br>5. To uncover the limitations of using process mining for RBAC analysis. |
| Goal2. To propose improvements to the way that process mining methods are used for analysing RBAC. | RQ3. If applied, will the proposed improvement enhance the effectiveness of RBAC analysis by using process mining? | 6. To propose improvements to the way that RBAC is analysed using process mining.<br>7. To implement the proposed improvements and test their effectiveness. |

Table 1.1: Goals, research questions and objectives

RQ1 and RQ2 aim to provide fundamental knowledge about the current state of the art of using process mining to analyse RBAC. RQ3 is an endeavour to create opportunities to use process mining for RBAC analysis. To achieve these two goals and answer

these questions, I have chosen multiple approaches, each suitable for a specific target. The success of this research will be measured by the degree to which objectives 1–7 have been met.

## 1.3 Research Methodology

Research methodology is understood as 'a way to systematically solve the research problem' [25]. It is a logical plan with a set of methods that guide the way to solving the research problem. In designing this thesis, I used as the theoretical framework the interactive research design model proposed by Maxwell [26]. This consists of five design elements, namely the goals, the conceptual framework, the research questions, the research methods and validation. These elements interact during research, with the research questions at the model's core. The approaches and methods used in this thesis were chosen based on the research questions.

In contrast to the traditional research design models that present the process linearly and systematically, Maxwell's interactive model [26] describes research as an iterative process that continuously changes and cannot be completely defined before the study has taken place. Thus, I chose to follow this interactive model for this thesis, as it would allow more flexibility in designing and conducting the research.

### 1.3.1 Overview

#### Phase I

The first phase of this research is to achieve Goal1 and subsequently answer RQ1 and RQ2. Thus, to answer RQ1, I conducted an SLR to identify, evaluate and summarise the state of the art of using process mining for RBAC analysis (reported in Chapter 3). This SLR identified the following six key areas: 1) the challenges of applying process

mining; 2) the organisational background and its role in the process; 3) the essential requirements for conducting the analysis; 4) the most suitable checking approaches; 5) the role of contextual information in the analysis; and 6) tailoring the analysis of RBAC by using process mining. These key areas have not been previously studied in this context; therefore, the knowledge acquired through this thesis can help improve the shape of the current research landscape. These six key areas identify unexplored points and uncover opportunities for improvement in the use of process mining for RBAC analysis.

To answer RQ2, a practical experiment integrated into the exploratory endeavours is conducted to gain knowledge of these unexplored points while identifying indicators of limitations to ensure that the proposed solution delivers what is needed (reported in Chapter 4). The SLR and the experiment were conducted synchronously, since the knowledge from one benefits the other. Thus, the research questions and objectives and the evaluation criteria designed for the SLR were refined and modified from the knowledge and insights gained from the experiment. Likewise, the shape and objectives of the experiment went through several modifications based on the identified methods and the results from the SLR.

The knowledge and results from Phase I provide a solid base for the proposed improvements. The key areas identified from the SLR provide several options for methods, directions and tools. Besides the fact that the experiment identifies new opportunities for improvements, it represents the environment in which the proposed method takes place.

### Phase II

Phase II aims to provide improvements in the analysis of RBAC by employing process mining. Therefore, in the final stage of this research, a multi-perspective approach to

improve the analysis of RBAC is introduced in Chapter 5 by using conformance checking. An experiment is conducted to validate the approach and to test and measure how well it works when applied in practice (reported in Chapter 5).

The outcome of this methods is compared to the one from the case study for evaluation. Furthermore, the results from the experiment are linked to knowledge from the SLR to draw further directions for future work.

In this interactive design, the different parts of this research are linked dynamically so that each one learns and eventually benefits from the other, leading the research to its conclusion. Figure 1.3 provides a visual presentation of how the research parts are integrated in this thesis. Thus, decisions about the design and methodology of these parts influence each other, and the hypotheses are developed based on the information from preceding step.

## 1.4 Thesis Contributions

The contributions of this thesis aim to answer the research questions and achieve the main objectives presented in Table 1.1. Figure 1.4 illustrates the research goals, questions and objectives linked to the actual work.

**RQ1: How is RBAC analysed using process mining methods?**

In Chapter 3, I report on an SLR based on 27 case studies published in the field. This is to better understand the research by analysing the related works. Thus, I provide detailed results of the current process mining approaches that address RBAC (Objective 1). Moreover, I provide condensed results of the current RBAC constraints and models discussed by process mining approaches (Objective 2). Finally, I clarify the process of RBAC analysis by using different process mining methods and discuss their strengths and limitations (Objective 3).

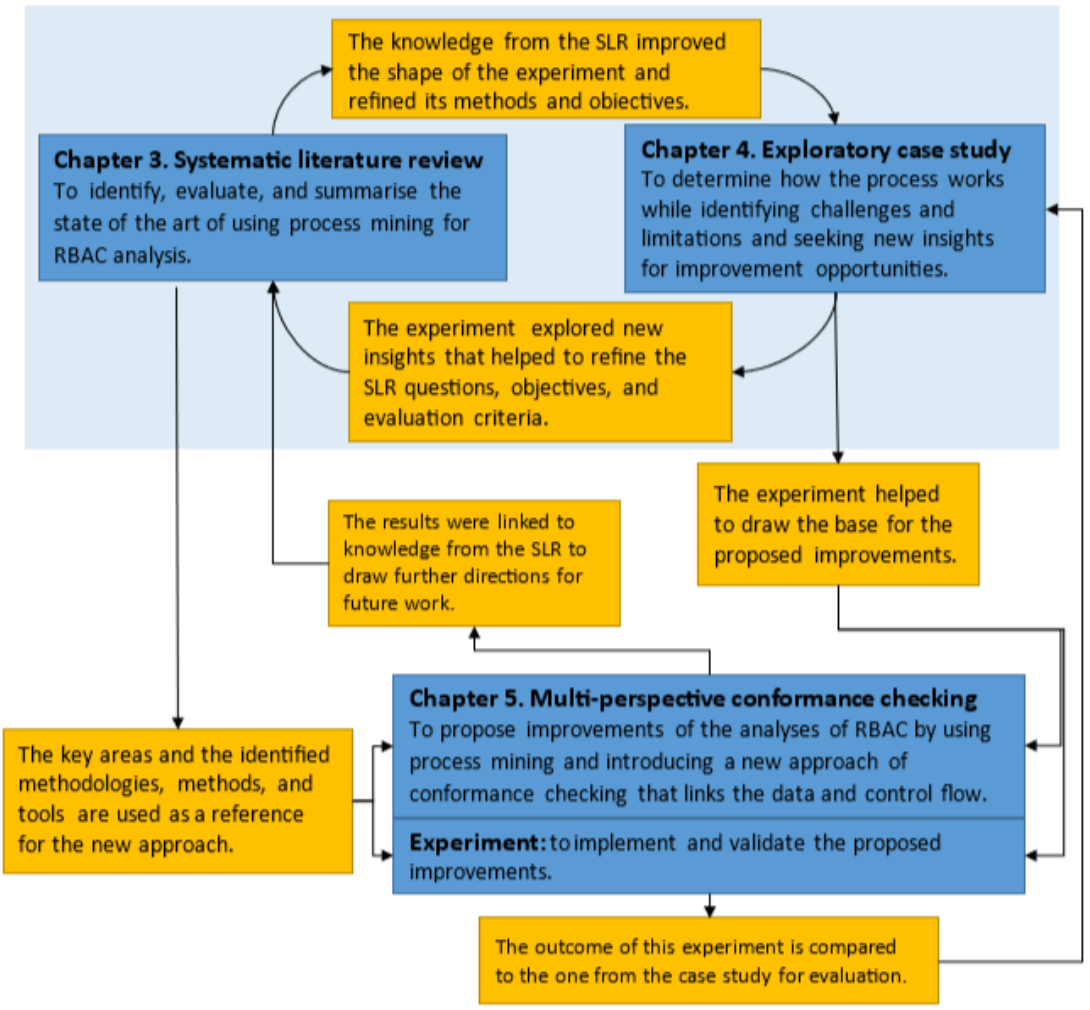



Figure 1.3: A visual presentation of how the research parts are integrated in this thesis.

**RQ2: To what extent can existing process mining approaches be used in RBAC analysis?**

In Chapter 4, I introduce an exploratory case study using process mining for RBAC checking. I apply a conformance checking approach using event logs and RBAC constraints as inputs. The checking is performed using linear temporal logic (LTL) formulas to detect any violation. I discuss the main challenges and opportunities for using such methods (Objectives 4 and 5). I carry on two case studies, one being a synthesised log based on real hospital scenarios by interviewing doctors and the second a real-life

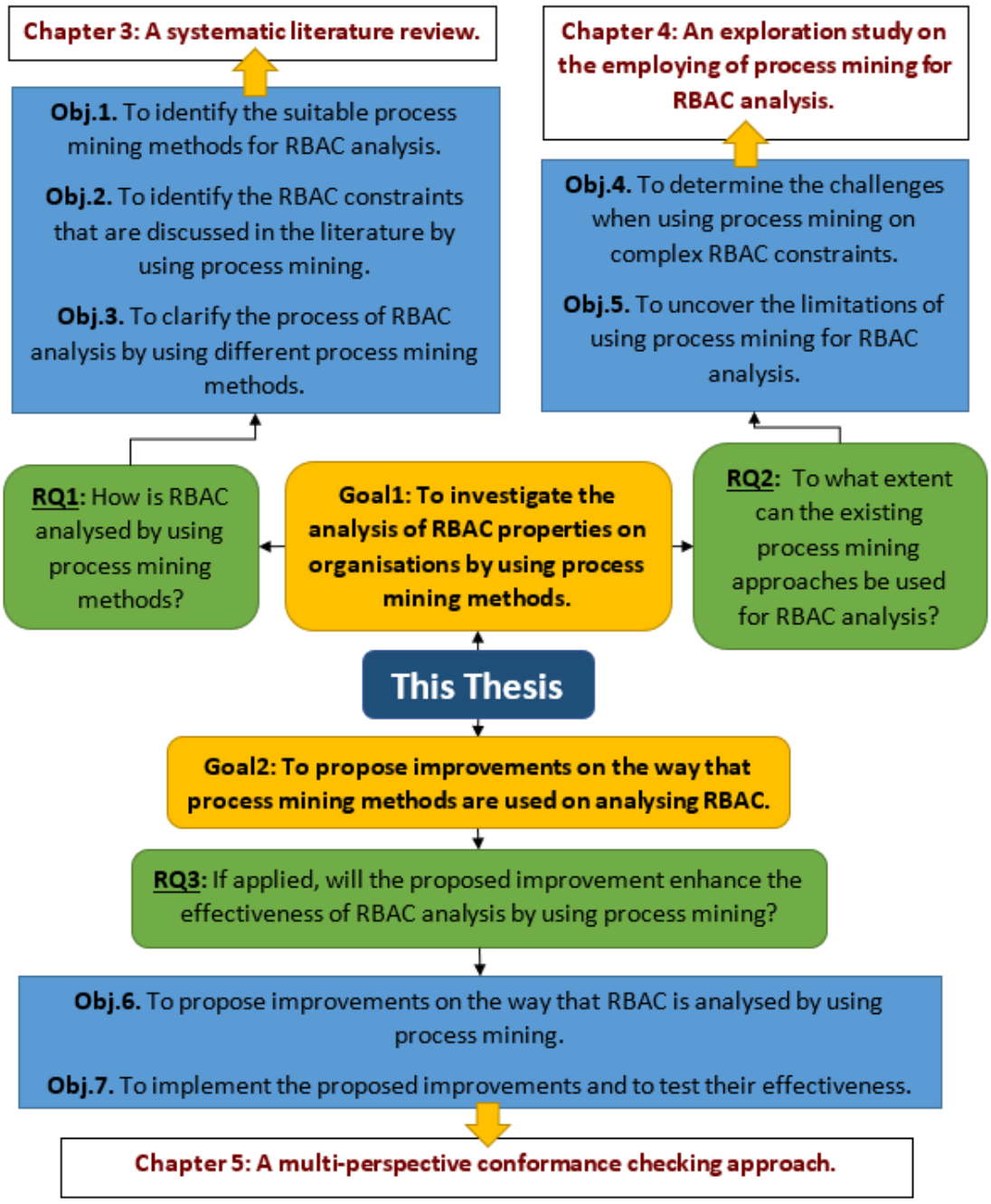



Figure 1.4: Goals, research questions and objectives linked to the actual work carried out.

log from incident and problem management processes supported by Volvo IT's VINST system (available in [27]).

**RQ3: If applied, will the proposed improvement enhance the effectiveness of**

**RBAC analysis by using process mining?**

In Chapter 5, I introduce a novel approach that enables multi-perspective conformance checking of RBAC's constraints. The approach considers data and control flow perspectives to detect RBAC violations while considering their contexts. In this way, I combine the multiple perspectives to link the purpose (data perspective) with the context (control flow perspective) to enable more robust detection of RBAC violations while considering the contexts of the occurrence (Objective 6). For validation, I test the proposed approach through a case study to demonstrate the strengths and limitations of the approach. This is followed by a discussion of possible scenarios and a behaviour analysis (Objective 7).

## 1.5 Publications

Some of the material presented in this thesis has previously published, and one paper is due to be submitted.

1. Alrahili R. Towards employing process mining for role-based access control analysis: a systematic literature review. In *Proceedings of the Future Technologies Conference*, pages 904–927. Springer, 2021

2. Alrahili R. On the usage of process mining for access control analysis. The 2019 NMS PGR Poster Competition, Poster presented at King's College London, London, UK,

3. Alrahili R. A multi perspective conformance checking approach of role-based access control. (to be submitted).

## 1.6 Thesis Outline

The remainder of this thesis is structured as follows.

- **Chapter 2** presents the essential concepts and background used in the remainder of this thesis.
- **Chapter 3** addresses the first research question and provides an SLR to better understand the research by analysing related and existing works.
- **Chapter 4** discusses the second research question and provides an exploratory case study of employing process mining for RBAC analysis. This is conducted on the process of a hospital's accident and emergency department (A&E) for handling patients as well as on incident and problem management processes.
- **Chapter 5** discusses the third research question and proposes a new approach of conformance checking that links the data and control flow perspectives to produce multi-perspective alignments while identifying their costs. Subsequently, the approach is applied for evaluation.
- **Chapter 6** concludes the key findings of this thesis and provides possible future research directions.

# Chapter 2

# BACKGROUND

The two main concepts of this thesis are RBAC and process mining. In this chapter, I introduce the essential concepts, methods and work related to these concepts to enable better understanding of the results produced in the following chapters.

## 2.1 Process Mining

Process mining is an emerging research subject between machine learning and data mining on the one hand and process modelling and analysis on the other [3]. End-to-end process models have not been thoroughly considered in data mining and machine learning techniques. Moreover, event data is not the focus of process science approaches. Hence, process mining bridges this gap by considering both process models and event logs. The idea of process mining is to study and analyse the processes *(modelled behaviour)* and the knowledge extracted from the event logs of the *observed or actual behaviour*, from which, one can determine what people and organisations do. For instance, their compliance can be checked using event logs and process models. Likewise, checking can be applied with security policies to compare how the restrictions are modelled and met in the log. Hence, process mining can be used to identify and understand

bottlenecks, inefficiencies, deviations, flaws and risks.

Aalst [24] stated that a primary strength of process mining is that it is generic and can be applied in any organisation. One reason for this is the applicability of process mining to a wide range of systems, such as pure information systems (e.g. ERP systems) and systems where the hardware plays a more prominent role (e.g. embedded systems). All that is needed is for the system to produce event logs, thereby recording the actual behaviour [31].

The process-aware information system (PAIS) is a well-known type that controls and executes operational processes that involve people, applications and information sources, as stated by process models [32], and is currently widely used to produce event logs. PAIS is a large umbrella that includes various systems that provide detailed information about the activities that have been executed [31]. Examples of PAIS systems are classical workflow management, ERP, PDM, CRM, middleware and hospital information systems.

### 2.1.1 The Fundamentals

#### Process models

Nowadays, most organisations use formal models designed to be analysed and used to present operational processes. The goal of a process model is to decide which activities need to be executed and in which order. It shows that some activities can be optional or mandatory. Moreover, some activities can occur sequentially or concurrently, while others can occur repeatedly. There are two types of process models that are widely used in process mining: Petri nets and business process modelling notation (BPMN). In this section, I introduce Petri nets, since it is the main type that I use in conducting this research. First, I present a simple system of the delivery procedure. The system starts by

registering an order by an office clerk. Then, a message about the delivery information is sent to the customer. Meanwhile, based on the weight and size of the item, a truck or a car delivery is chosen for the item to be delivered to the customer's location. The process ends if the item is delivered; if not, the item is sent back for another delivery and the case must be reported. I define the delivery system as simple and easy to understand yet complex enough to handle different notations. In the following sections, I use this process as an example to introduce the concepts relevant to this thesis.

### Petri net

Petri net is the oldest process modelling language and provides the formal foundation for modelling concurrency [3]. Figure 2.1 shows the Petri net presentation of the delivery system. The graphical notations of Petri nets consist of places and transitions, where transitions represent activities. This process contains eight places, starting at p0 and ending at p7. Tokens can flow through the network by the firing rule, where they move over places. The state of a Petri net is determined by the distribution of tokens over places and is referred to as its marking. Figure 2.1 has only one token in the first place.

Here, I introduce the formal definition of a Petri net by Aalst [3].

**Definition 1.** ***(Petri Net).*** *A Petri net is a triplet $N = (P, T, F)$ where:*

- *$P$ is a finite set of places and $T$ is a finite set of transitions such that $P \cap T = \phi$.*
- *$F$ is a finite set of arcs, such that $F \subseteq (P \times T) \cup (T \times P)$ is a set of direct arcs, called a flow relation.*

In this thesis, I use Petri nets to demonstrate the used process and simulate event logs using special tools, as described in the upcoming chapters.

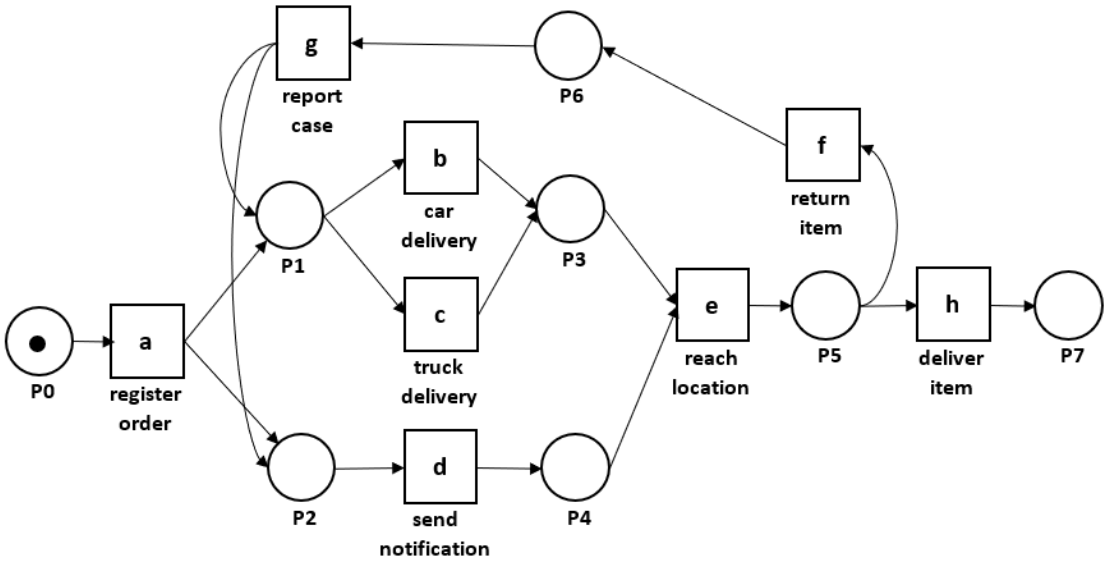



Figure 2.1: Petri net presentation of a delivery system.

### Event logs

In this section, I use the event log shown in Figure 2.2 to clarify some of the foundational concepts. The figure shows a fragment of a possible log corresponding to the delivery system. Each line presents one event, and events are grouped into cases with unique IDs to distinguish them, along with the events. Figure 2.2 shows Activity, Timestamp, and Resource as attributes for each event. In this log, *Activity* presents the activity's name, *Timestamp* shows the date and time in hours and minutes for when the event was completed and *Resource* shows the name of the person who executed the activity. For example, Case 1 has five associated events. The first event has the value 45811752 as an event ID. This event describes the execution of the activity Register order by Bob on 02-11-2022 at 09:14. In other logs, there may be more elaborate attributes, such as the start time, the end time, the cost and the resource's role.

The information required for an event log depends on the process mining techniques and type of questions at hand. However, the minimum requirements for process mining are that any event can be related to a case and activity and that the events within a case

| Case ID | Event ID | Attributes | | | |
|---|---|---|---|---|---|
| | | Timestamp | Activity | Resource | ... |
| 1 | 45811752 | 02-11-2022:09.14 | Register order | Bob | |
| | 45811753 | 02-11-2022:12.33 | Send a notification | Sara | |
| | 45811754 | 02-11-2022:12.35 | Car delivery | Mike | |
| | 45811755 | 02-11-2022:15.11 | Reach location | Mike | |
| | 45811756 | 02-11-2022:15.16 | Deliver item | Mike | |
| 2 | 45811787 | 03-11-2022:11.05 | Register order | Sara | |
| | 45811788 | 03-11-2022:14.41 | Send a notification | Bob | |
| | 45811789 | 03-11-2022:14.55 | Truck delivery | Mike | |
| | 45811790 | 03-11-2022:17.21 | Reach location | Mike | |
| | 45811791 | 03-11-2022:17.25 | Deliver item | Mike | |
| 3 | 45811881 | 05-11-2022:08.10 | Register order | Sara | |
| | 45811882 | 05-11-2022:10.04 | Send a notification | Bob | |
| | 45811883 | 05-11-2022:12.21 | Car delivery | Emma | |
| | 45811884 | 05-11-2022:14.07 | Reach location | Emma | |
| | 45811885 | 05-11-2022:14.56 | Return item | Emma | |
| | 45811886 | 05-11-2022:16.13 | Report a case | Sara | |
| | 45811911 | 06-11-2022:09.35 | Send a notification | Bob | |
| | 45811912 | 06-11-2022:10.03 | Car delivery | Emma | |
| | 45811913 | 06-11-2022:10.49 | Reach location | Emma | |
| | 45811914 | 06-11-2022:10.53 | Deliver item | Emma | |
| ... | ... | ... | ... | ... | ... |

Figure 2.2: An example event log of the delivery system.

are ordered [3]. Therefore, the caseID and Activity columns in Figure 2.2 represent the essential information for process mining. By using only these two columns, we obtain the following sequence of activities (traces) for Case 1, Case 2 and Case 3: <Register order, Send a notification, Car delivery, Reach location, Deliver item>; <Register order, Send a notification, Truck delivery, Reach location, Deliver item>; and <Register order, Send a notification, Car delivery, Reach location, Return item, Report a case, Send a notification, Car delivery, Reach location, Deliver item>, respectively. Figure 2.3 shows the tree structure of an event log. From this figure, we can list the following:

- A process consists of cases;
- A case consists of events;

- Each event relates to one case;
- Events can have attributes.

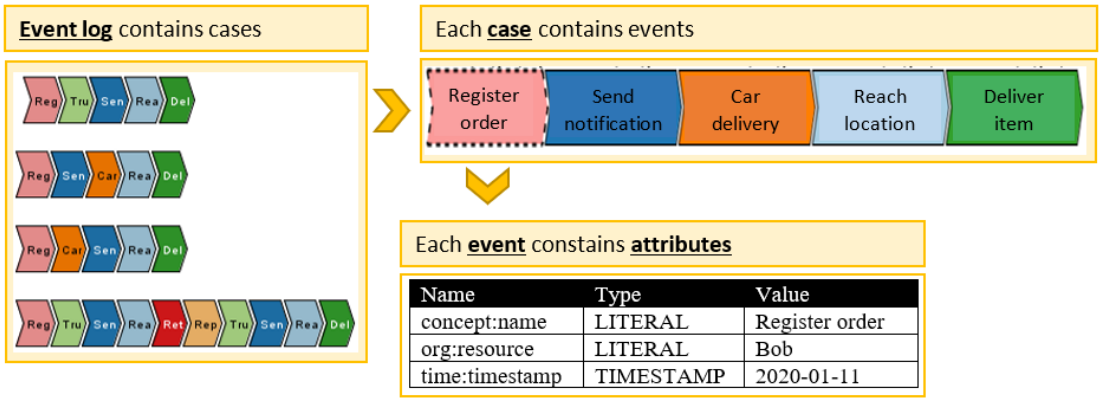



Figure 2.3: Event log structure showing cases, events and attributes. The ProM tool is used to show the event log [2].

Following, I introduce the formal definitions of various notions of event logs as described by Aalst [3]:

**Definition 2.** ***(Event, attribute).*** *Let $E$ be the event universe. Events may have different attributes. Let $AN$ be a set of attribute names. For any event $e \in E$ and name $n \in AN$, $\#_n(e)$ is the value of attribute $n$ for event $e$. If an event $e$ does not have an attribute named $n$, then $\#_n(e) = \perp$. The standard attributes (activity, timestamps, resource) of activity $e$ can be presented as $\#_{activity}(e)$, $\#_{time}(e)$, and $\#_{resource}(e)$, respectively.*

The event log in Figure 2.2 shows three attributes: timestamps, activity and resource; hence, $\#_{activity}(e_4 5811752) =$ Register order, and $\#_{resource}(e_4 5811755) =$ Mike.

**Definition 3.** ***(Classifier).*** *For an event $e \in E$, $\underline{e}$ is the name of the event. A classifier simply decides how to identify events. If events are identified by their names, then $\underline{e} = \#_{activity}(e)$.*

In this thesis, I use activity as the identifier. Therefore, I used activities to write traces in the previous section.

**Definition 4.** ***(Case, trace, event log).*** *Let CC be the case universe. Like events, cases have attributes. For any case $c \in C$ and name $n \in AN$, $\#_n(c)$ is the value of attribute n for case c. If a case c does not have an attribute named n, then $\#_n(c) = \perp$. Each case has a mandatory attribute trace.*

*A trace is a finite sequence of events $\sigma \in E^*$, such that each event appears only once; that is, for $1 \leq i < j \leq |\sigma| : \sigma(i) \neq \sigma(j)$.*

*An event log is a set of cases $L \subseteq C$ such that each event appears at most once in the entire log; that is, for any $c_1, c_2 \in L$ such that $c_1 \neq c_2 : \delta_{set}(\hat{c1}) \cap \delta_{set}(\hat{c2}) = \phi$.*

#### Extensible Event Stream (XES)

The extensible event stream (XES) is a new type of event log that aims to simplify the storage and the reading of event logs in a standardised way [33]. In [34], Christian et al. provided the standard definition of XES. In 2016, XES became the official IEEE standard for storing event data [35]. The structure of XES shows that any XES document contains one log. A log consists of a number of traces, each describing a sequential list of events associated with a specific case. Logs, traces and events can have many attributes. Data stored in event logs have five types: string, date, int, float and Boolean. A primary advantage of XES is that it enables adding attributes, where an event can have any number of attributes through extensions, giving semantics to particular attributes. In this thesis, I use event logs in XES format.

### 2.1.2 Process Mining Types

In his book, Aalst [3] distinguished three main types of process mining, namely process discovery, conformance checking and enhancement. Figure 2.4 shows the three types

of process mining and the link between event logs and process models.

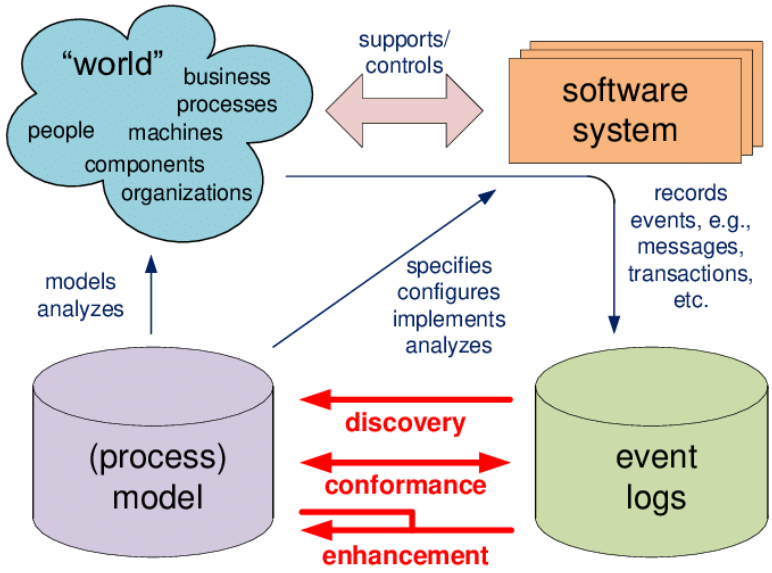



Figure 2.4: Positioning of the three main types of process mining: discovery, conformance and enhancement [3].

### Process discovery

Process discovery is a type of process mining that extracts process models from event logs. Event logs record information about the execution of a process, such as activity names, performers and timestamps. Rinderle-Ma et al. [36] distinguished process mining discovery using three main perspectives: control flow, data and organisational perspectives. The control flow perspective focuses on the control flow, including selecting and ordering activities in models such as Petri nets; the data perspective focuses on information related to the system, such as activities and cases; and the organisational perspective focuses on the resources executing the activities. A resource is the performer of an activity, either human or non-human. A human resource is typically a member of a group or an organisation [37].

### Conformance checking

Conformance checking is a well-known type of process mining that can be applied on event logs and process models. The relationship between observed and modelled behaviour can be indicated using event logs and process models as input for checking [24]. Conformance checking aligning event data and process models provides accurate results that reflect reality, increasing the value of applying process mining techniques in organisations. For instance, Carmona et al. [38] showed this type of checking to be the reason behind growing industrial interest in conformance checking, and it is expected to be the fastest-growing subject in process mining in future years.

There are three main approaches for conformance checking, namely rule checking, token replay and alignments, as shown in 2.5. Although all three approaches use models and event logs as input, they differ in the way they use them, as follows:

- Rule-based checking checks a set of rules (constraints) on a process model or event log.
- Token replay starts with an event log, attempting to replay traces in a process model.
- Alignments use both event logs and process models synchronously.

### Enhancement

Process enhancement aims to extend or improve an existing process model using information about the actual behaviour. The input to an enhancement method is an existing process model and an event log. Process models often exist as part of process documentation, or the basic control flow of process models is discovered by a process discovery

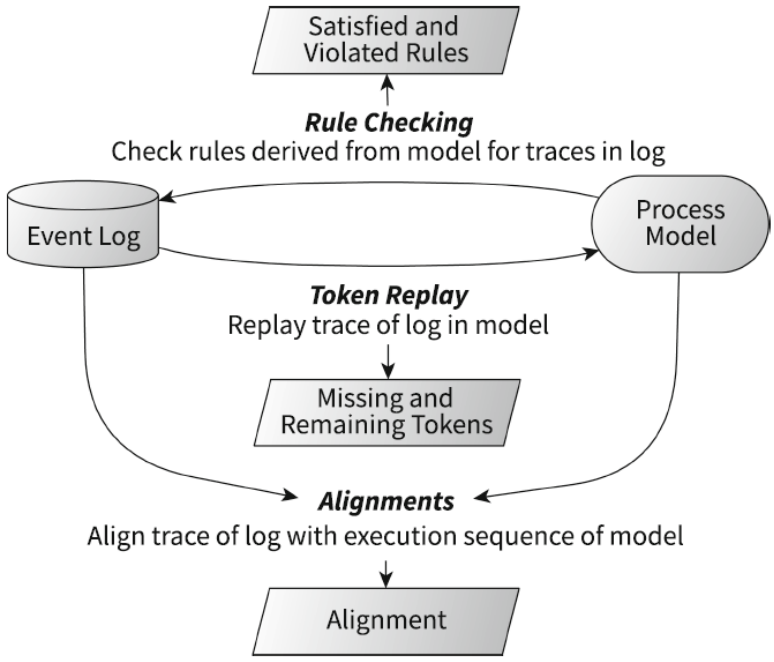



Figure 2.5: General approaches to conformance checking: rule-based checking, token replay and alignments, from [4].

method. These models can be enhanced with extra information, such as the results of conformance checking, to better reflect the real process execution.

### 2.1.3 Process Mining Tools

#### ProM: A process mining platform

ProM [2] is an open-source process mining platform widely used in academic research. It can be freely downloaded from `www.promtools.org`. ProM has several plug-ins for different purposes, such as discovery and checking. Moreover, one can develop a plug-in and use it within the framework. There are different versions of ProM that can be used in many operating systems, including Windows and Linux.

Figure 2.6 shows a screenshot of ProM tool 6.9 after uploading an event log and selecting the log filtering plug-in as an example. I marked four sections, as follows:

- **A** shows the required inputs, one or more depending on the selected plug-in.
- **B** shows the list of available plug-ins highlighted in either green or yellow. Green means all the required inputs are available, while yellow means that at least one required input is missing.
- **C** shows the format of the expected output. For example, in Figure 2.6, the expected output is a filtered log.
- **D** shows information about the plug-in and its author.

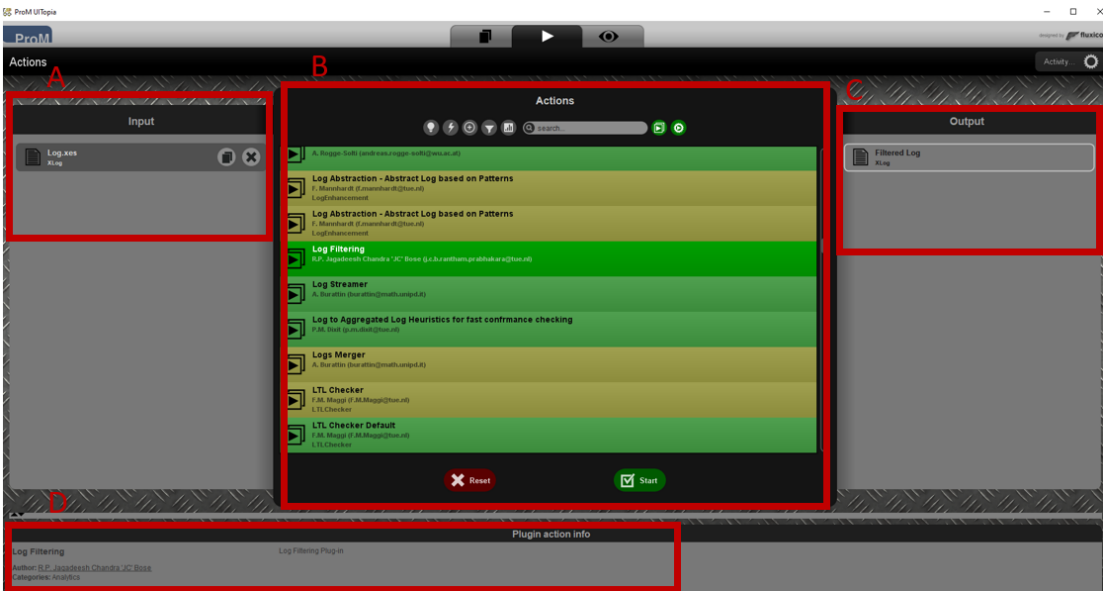



Figure 2.6: A screenshot of ProM tool 6.9 after uploading an event log.

In this thesis, I use different versions of ProM as the main process mining tool.

### CPN: A tool for simulating petri nets

CPN [39] is a tool for editing and simulating coloured Petri nets. The tool features incremental syntax checking and code generation, which occur while a net is being constructed. A CPN tool can create timed and untimed Petri nets. It can also simulate

event logs, although one must have a solid knowledge of constructing Petri nets. I use a CPN tool in this thesis to simulate and analyse event logs. Figure 2.7 shows a screenshot of the CPN tool after simulating the delivery system.

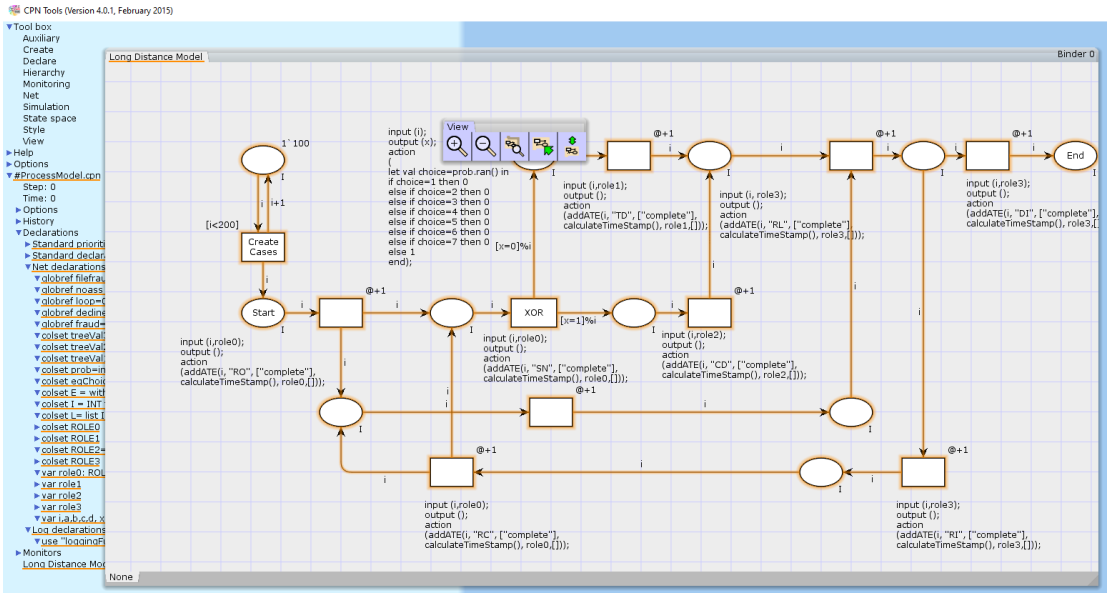



Figure 2.7: A screenshot of the CPN tool.

## 2.2 Role-Based Access Control

Access control systems are among the most critical components of security. They control which users have access to specific resources in a system according to an access control policy. Access control policy refers to the high-level requirements that specify how access is managed, who may access information and under what circumstances. There are several well-known access control policies, such as RBAC, discretionary access control and mandatory access control. This thesis focuses on RBAC policies in particular.

In recent years, RBAC [6, 40] developed into a de facto standard for access control in research and industry. The concept of RBAC has been widely studied in the literature

[7, 41, 6, 42, 43]. Its basic idea is that users are assigned to roles, and permissions are assigned to roles, so users acquire permissions by being members of roles. Using roles makes adding, removing and adjusting permissions easier than assigning permissions to users individually.

This section introduces a family of four conceptual models defined by Sandhu et al. [42], which are base, hierarchical RBAC, constraints and consolidated. The base model is the minimum requirement for an RBAC system, while the advanced models, hierarchical RBAC and constraints, include the base model. Hierarchical RBAC adds role hierarchies (i.e. situations where roles can inherit permissions from other roles), while the constraints model adds constraints (i.e. imposing restrictions on acceptable configurations of the different components of RBAC). The hierarchical RBAC and the constraints model are incomparable to one another. The consolidated model includes hierarchical RBAC and the constraints model, and thus, by transitivity, the base model.

In addition to these four RBAC models, Bertino et al. [44] introduced the temporal RBAC (TRBAC) model, an RBAC model extension that provides temporal dependencies among roles. This was followed by Joshi et al. [45], who identified RBAC constraints that can be applied in a specific set of intervals.

### 2.2.1 Base Model

The base model [42] consists of four entities: users (U), roles (R), permissions (P) and sessions (S). These can be seen in Figure 2.8. A user is an agent (in this case, a human being), a role is a job title within an organisation that is defined and associated with responsibilities, a permission allows access to particular resources and sessions are established by users within roles.

The following are the four main relations in an RBAC model:

- **Permission assignment (PA):** This assigns the necessary permissions to roles to

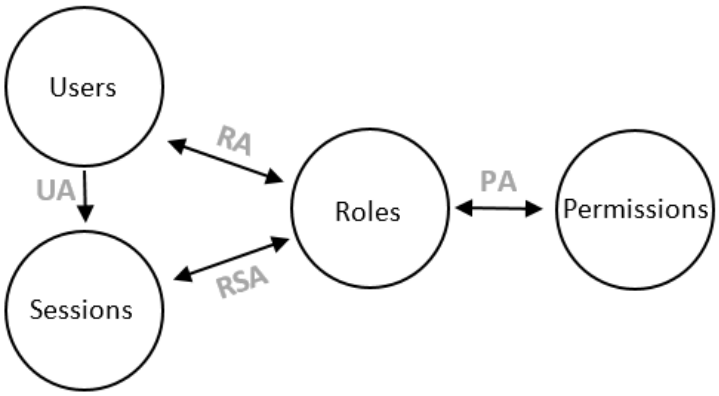



Figure 2.8: Elements of RBAC.

perform their jobs. A role can have many permissions, and the same permission can be assigned to many roles.

- **Role assignment (RA):** This specifies the roles a user is enabled to play. A user can belong to many roles, and a role can have many users.
- **User assignment (UA):** This assigns users to a session.
- **Role session assignment (RSA):** This assigns roles to a session.

Authorisation means that only authorised users are allowed to execute certain activities. The following are Crampton's formal definitions of authorisation in RBAC [46].

**Definition 5.** ***(RBAC configuration).*** *Let a set of roles R, a user-assignment (relation) $UA \subseteq U \times R$, and a permission-assignment (relation) $PA \subseteq R \times T$ be given. A tuple $(UA, PA)$ is called an RBAC configuration. A user u is authorised to execute a task t if there is a role $r \in R$ such that $(u, r) \in UA$ and $(r, t) \in PA$. The user u acts in role r if $(u, r) \in UA$.*

**Definition 6.** ***(Authorisation).*** *Let $(UA, PA)$ be an RBAC configuration. The RBAC process $RBAC(UA, PA)$ models the enforcement of role-based authorisations and the*

*administration of UA. RBAC(UA, PA) engages in a business event b.t.u if u is authorised to execute t with respect to (UA, PA).*

### 2.2.2 Hierarchical RBAC

Here, I introduce the hierarchical RBAC by Sandhu et al. [7, 42]. The idea behind this type is to group objects into object classes to make managing permissions easier and clearer. Hierarchical RBAC supports role hierarchies (RHs), where roles can inherit permissions from other roles. By enabling the hierarchies feature, roles can be structured to reflect an organisation's line of authority and responsibility. By convention, senior roles inherit permissions from junior roles. Using the delivery system as an example, drivers deliver items using a car or a truck. Hence, car drivers have permission to perform the delivery tasks that can be done using cars while truck drivers have permission to perform tasks that can be done using trucks. Whereas driving trucks implies the ability to drive cars, driving cars does not imply the ability to drive trucks. Thus, truck drivers may inherit all the permissions of car drivers.

### 2.2.3 Constraints Model

The constraints model introduces constraints to the RBAC model. Constraints are an important aspect of RBAC; as Sandhu et al. [42] argued, this type of RBAC was the principal motivation behind RBAC. There is a considerable amount of research on identifying and defining RBAC constraints [47, 41, 42]. Regarding the base model, constraints can be applied using the UA and PA relations and the U, R and S entities. The outcome of the constraints will allow or deny access. Here, I introduce the relevant constraints to this work.

**Separation of duty.** The concept of separation of duty (SoD) has been widely studied

in the literature [48, 49, 46, 50, 51, 52, 53, 54, 55, 56, 57, 58]. For example, Simon et al. [55] defined SoD as a security principle used to manage multi-person control policies, requiring different people to complete certain tasks. This principle or constraint aims to prevent fraud by involving multiple people in completing certain tasks. Thus, an individual does not have full authority to perform such tasks. Many authors have discussed and categorised different forms of SoD [55, 57, 58]. The work in [55] defined two main types of SoD, namely static separation of duty (SSoD) and dynamic separation of duty (DSoD). SSoD is the simplest variation of SoD and can be applied when there is no case of a person being a member of two roles that have permission for the restricted tasks. Thus, a person authorised to perform any of these tasks should not be in two different roles that have permission for these tasks. In contrast, DSoD has richer semantics and tends to generalise, providing support for many SoD policies [56]. In the simplest variation of DSoD, restricted roles may have common members but users may not assume both roles simultaneously [43].

**Binding of duty.** Several studies of binding of duty (BoD) exist in the literature [59, 60, 61, 62, 63]. In contrast to SoD, BoD constrains task authorisations by defining that the same individual or role must perform two or more tasks. Like SoD, there are two types of BoD, namely dynamic binding of duty (DBoD) and static binding of duty (SBoD). However, DBoD constraints define that the same individual must perform bound tasks. In contrast, SBoD constraints define that bound tasks must be performed by members of the same role. DBoD and SBoD are sometimes referred to as subject-based and role-based, respectively [60].

**Cardinality.** Cardinality is a constraint determining the maximum number of performers involved in a set of tasks. In [40], Sandhu et al. showed that cardinality de-

termines the maximum number of members in a role, number of roles a user can belong to and number of permissions assigned to a role. For example, in the delivery service, no more than two different drivers can be involved in one delivery. This constraint has more value on sensitive data, such as in healthcare.

## 2.3 Access Control Analysis

Faulty policies, misconfiguration and flaws in software implementations can result in serious vulnerabilities. As software systems become more complex, by managing a large number of objects and users and containing sensitive information, the process of designing and implementing the required access control policy becomes increasingly challenging. Therefore, it is crucial to ensure conformance between policy specifications and their intended function. Analysing access control has been studied mainly from the technical side, in particular for verification and validation. Thus, access control models are written to bridge the gap between policy specification and its implementation. Access control models are the formal presentations of the policies enforced by access control systems.

It is crucial to identify any inconsistency between access control policy, model and implementation, since the correct enforcement of policies depends on the models and implementations being correct. Therefore, a considerable amount of research has been published on verification and validation methods for access control [64, 11, 65].

In this section, I discuss the verification principles (Section 2.3.1) and methods used (Section 2.3.2) for access control verification in general. This is because these aspects are widely studied for access control and also applied particularly for RBAC. In Section 2.3.3, I discuss these methods in the context of RBAC, focusing on their limitations and new directions.

### 2.3.1 Analysis Principles

The main goal of access control policy and implementation verification is to detect conflicting or missing rules by verifying the model and testing its implementation [11]. Faults in access control systems can been seen in cases of leaking privileges or authorised access being blocked. These can result in incorrect granting or denying decisions.

**Model verification.** This principle represents the first building block the verification, as it forms the link between the policy and its implementations. From the user's side, it can be seen as a set of clear rules on how the system should work, whereas developers may see it is a guideline for implementation. Thus, the correct implementation and enforcement of policies is based on correct and precise models. Therefore, access control models must undergo rigorous verification and validation through systematic testing to ensure that there are no faults within the policy that leak or block access privileges.

**Implementation test.** After ensuring the correctness of an access control model, checking for correct implementation takes a place. The implementation test ensures that the correct policy is enforced and functions as intended by testing all possible scenarios. This is usually achieved by applying a test oracle.

To a considerable degree, access control model implementation depends on the organisation's system and is therefore modified more often to reflect the dynamics of organisational changes. Therefore, after introducing changes, consistency between a model and its implementation cannot be guaranteed. For instance, in the case of an organisation establishing a new job function within its structure, it will require a new role to which the new user is assigned. Thus, the new implementation should conform to the model. Moreover, checking must take a place to ensure that the new role is not

in conflict with the roles that are already implemented. Therefore, the access control model may also be modified to solve any conflict.

### 2.3.2 Analysis Methods

Several recent studies have categorised and discussed technologies and methods for verifying access control [64, 11, 65]. In this section, I discuss four main methods that are widely used for verification, namely model checker [12], data structure [66], system simulation [67], and machine learning [65].

#### Model checker

The model checker method is used widely for access control verification. The idea behind it is to convert the access control policy into a formal model, usually a finite state machine (FSM). Test cases are then verified against the model to detect any policy fault. The model can be checked by temporal logics, such as computation tree logic (CTL) and LTL extended with past operators [11].

For instance, NIST developed the Access Control Policy Tool (ACPT) [12], which can be used for RBAC modelling, implementation and verification. The tool generates an FSM model and verifies it by applying the symbolic model verification NuSMV [68] tool. This generates test cases that are used to ensure the following:

1. Safety: The FSM model does not violate any rule of the policy, and it always ensures the desired state;

2. Liveness: The FSM contains neither a deadlock, whereby the system waits forever for system events, nor a livelock, whereby the model repeatedly executes the same operations forever.

### Data structure

The data structure method has been used widely for access control policy verification [66, 69, 70]. The idea behind this method is that it uses structured data that represents underlying policy rule logics to identify any conflict or problem within the policy.

For instance, the Margrave tool [66] for verifying access control policies, which was written in extensible access control markup language (XACML) [71] against test cases, represents policies as binary decision diagrams [72] that assign binary values to the policy attributes that traverse from the root to a terminal of the rule's permission [73]. Thus, the idea behind the Margrave tool is to verify test cases in the form of queries, which should hold by the binary tree. Otherwise, counter examples are produced, indicating violations of the tree structure [74].

### System simulation

The model checker and data structure methods require completed access control policies as the input for the verification. Moreover, they do not provide information about the policy rule that causes conflict or faults [11]. In contrast, the system simulation method builds a simulated system of the access control policy rules for verification. Thus, each policy rule is introduced to the simulation, which enables the detection of the source of faults.

For instance, the access control rule logic circuit simulation (ACRLCS) system [67] detects a fault when a rule that causes it is added to the existing model. Thus, fixing errors can be performed in real time before adding more rules to the model that might complicate the detecting and fixing processes. The idea behind ACRLCS is to detect only the newly added rule to a model that contains only correct rules, instead of checking the whole model at once. ACRLCS [67] was added to the ACPT tool [12] as an optional

step.

#### Machine learning

The machine learning method is used for access control verification by checking the logic of policy rules [65, 75, 76]. Data mining allows the extraction of data patterns from large data and converts them into the required format. Data mining techniques have been used for access control policy verification [77, 78]. Recently, a classification technique was introduced for access control verification [65] whereby the algorithm uses access control policy rules as samples (training data) and the permission assigned to the rules as a classification target. Then, the algorithms build a classification model of the policy to check the accuracy and detect faults in the policy rules. The author showed that machine learning methods can be used to enhance the existing verification methods.

### 2.3.3 Limitations and Opportunities for Improvements

Although there has been a considerable amount of research on access control verification, there still remain a number of unexplored aspects and unanswered questions that could enhance the analysis of RBAC.

**Exploiting the actual execution of a software system.** None of the above methods exploit the actual execution of processes that are recorded in event logs. Nowadays, organisations tends to record such information in event logs and use it for analysing and enhancing their systems [19]. Thus, new formats and extensions of event logs have been proposed to provide flexibility and enable the recording of rich and complex information [34]. Exploiting the actual execution provides knowledge about the control flow of the software system, as in the order of the

executed activities, not only the ones related to the RBAC model, such as in the traditional methods. Moreover, it provides information about different attributes, such as the time of execution, the location and information about the performer, such as their name, role and groups. To my knowledge, using the actual execution of a software system for RBAC analysis has not been studied thoroughly.

**Policy with numerical attributes.** Policy with numerical attributes cannot be verified by traditional methods, as it requires a test oracle with an infinite number of attribute values. Thus, [79] proposed a model-based approach for access control verification that uses Petri nets, since they can capture both control flows and data flows. Also, it has been shown that machine learning techniques can be used to solve this problem, as they work effeminately with non-binary attributes. However, the link between the model-checker and machine learning methods for access control analysis is still in its initial stages.

**RBAC as a socio-technical problem.** The traditional access control methods confined analysis to their technical side. However, modern access control policies, and RBAC in particular, extend beyond that, as they are based on users and roles, rather than objects alone. Nyame et al. [80] emphasised that RBAC is a socio-technical concept and proposed a conceptual framework for RBAC design in knowledge management systems. Analysis of RBAC lacks the link between its technical and social aspects.

## 2.4 Discussion

Process mining benefits from extracting knowledge from event logs as well as analysing and discovering sophisticated process models, such as Petri nets. Process mining can be seen as the link between machine learning and process modelling, since it enables

the usage of the rich syntax of process execution (for modelling) and rich information of data (for machine learning techniques). Moreover, exploiting event logs can be used to introduce a social aspect to the analysis, as the logs provide the information of the actual users. Thus, process mining can be seen as a strong candidate for access control analysis.

In this thesis, I investigate the analysis of RBAC by exploiting recorded data reflecting users' actual behaviours in an organisation. Thus, I study the employment of process mining for RBAC analysis.

# Chapter 3

# A SYSTEMATIC LITERATURE REVIEW OF THE EMPLOYING OF PROCESS MINING FOR RBAC ANALYSIS

## 3.1 Introduction

This chapter introduces an SLR of the existing works on using process mining for RBAC analysis. This work has been published as:

Alrahili R. Towards employing process mining for role-based access control analysis: a systematic literature review. In *Proceedings of the Future Technologies Conference*, pages 904–927. Springer, 2021 [28].

Chapter 3 has been designed to address Goal1, RQ1 and objectives 1–3 of this thesis (Table 3.1).

| Goal | Research questions 1 | Objectives |
|---|---|---|
| To investigate the analysis of RBAC properties on organisations by using process mining methods. | How is RBAC analysed using process mining methods? | 1. To identify suitable process mining methods for RBAC analysis.<br>2. To identify RBAC constraints using process mining that are discussed in the literature.<br>3. To clarify the process of RBAC analysis by using different process mining methods. |

Table 3.1: The goal, research question and objectives discussed in Chapter 3.

### 3.1.1 Motivation

In recent years, process mining has been extensively used in security analysis. However, too little attention has been paid to discovering and verifying the RBAC constraints of business processes based on recorded event data. Likewise, although various access control policy validation mechanisms, such as data mining [77], model checking [12] and formal methods [81], have been proposed by researchers, mechanisms based on the actual execution of the process have been neglected. Thus, our understanding of the use of process mining in RBAC analysis is limited.

The purpose of this chapter is twofold. First, I expand understanding of process mining approaches and their role in analysing RBAC, and second, I seek to answer the first research question of this thesis and meet its objectives. Therefore, I consider the intersection of two principles, namely RBAC and process mining.

**RBAC:** I consider the RBAC model by Sandhu et al. [42], the RBAC components defined by NIST [7] and the introduced temporal extension of RBAC by Bertino et al. [44]. Four model components define the NIST RBAC model, namely core RBAC, hierarchical RBAC, SSoD and DSoD. Core RBAC is required in any RBAC system, but the other principles are independent and may be implemented

separately [7].

**Process mining:** I consider the three types of process mining, namely discovery, conformance and enhancement. Process discovery takes an event log and produces a model. Conformance checking compares defined policies with event logs or models. Enhancement extends or improves an existing process model using information about the actual process recorded in an event log. Furthermore, I consider the methods and technologies used for the analysis.

To effectively identify, evaluate and summarise the state of the art, I provide a systematic review of the growing literature on methodologies tailored to using process mining for RBAC analysis.

### 3.1.2 Contributions

The main contribution of this chapter is an SLR undertaken according to the guidelines provided by Kitchenham [82]. I analysed 27 publications discussing 40 approaches. Specifically, I provide four concrete contributions:

- **Process mining approaches:** I provide detailed and condensed results in the context of current process mining approaches used to analyse RBAC.
- **RBAC aspects:** I provide detailed and condensed results in the context of current RBAC constraints and models discussed by process mining approaches.
- **Understanding the process:** To gain a deep understanding of the process, I discuss the practical side of each approach by identifying the requirements and types of outputs. I also present the tools used in the analysis.

- **Strengths and weaknesses:** I uncover the strengths and benefits of employing process mining for RBAC analysis while also revealing the main challenges and directions for future work.

This analysis shows the growing interest in process mining and RBAC, leading to several insights while highlighting some of the most important contributions and accomplishments. The results show that using process mining for RBAC analysis is highly promising but still in its early stages, so more effort is required.

### 3.1.3 Outline

The remainder of this chapter is structured as follows. Section 3.2 describes the methodology of collecting data to conduct this SLR. Section 3.3 presents the main results of this SLR. Section 3.4 answers the research questions and presents data analyses from the findings. Section 3.5 provides a data quality assessment by discussing construction, internal and external threats to validity and how they have been mitigated. Section 3.6 presents related work. The conclusions are discussed in Section 3.7.

## 3.2 Methodology

This chapter aims to explore and provide a better understanding of the field. Fundamentally, knowledge advancement must be built on prior work. Thus, understanding the current state of the art and providing a literature review are essential elements of academic research because knowing and understanding the existing work helps identify gaps to explore and opportunities to improve. SLRs are used in academia to identify, evaluate and summarise a specific theme's state of the art relative to a particular research question or topic area [82],[83].

### 3.2.1 Overview

I begin by presenting some of the features that distinguish a systematic review from other types of literature review, as defined in [82].

- **Review protocol:** Systematic reviews use rigorous methods that, from the start, define the research questions and methods used to perform the review. Other literature reviews can be broad and descriptive and use various methods to perform the review.
- **Data selection:** Systematic reviews define explicit inclusion and exclusion criteria to assess each potential study to select the most relevant research on the topic.
- **Assessing search results:** Systematic reviews specify exactly the required information from each study.
- **Data quality assessment:** Systematic reviews assess the quality of the evidence by using a risk of bias assessment.

A systematic review fairly synthesises existing work undertaken through a predefined search strategy. Due to the thoroughness and transparency features of systematic reviews, their results tend to be of higher quality and with lower bias than other types of literature review. A primary step in performing a systematic review is to follow a well-defined protocol to increase the reproducibility of the study. Many guides have been proposed to help reviewers design systematic reviews [82, 83, 84]. Likewise, many evaluation methods have been proposed to check the quality of an SLR, such as the preferred reporting items for systematic reviews and meta-analyses (PRISMA) statement [85]. PRISMA defines a set of items to help researchers improve their SLRs.

This SLR has been undertaken based on the guidelines provided by Kitchenham [82]. This chapter presents the process, stages and results of the systematic review of

research on using process mining for RBAC analysis that I conducted to build the first part of this thesis. To evaluate the quality of my SLR, I followed the PRISMA statement by applying the 27-item checklist and the four-phase flow diagram. The 27-item checklist covers all aspects of a manuscript, including its title, abstract, introduction, methods, results, discussion and funding. The idea behind it is to improve the transparency of systematic reviews. The four-phase flow diagram illustrates the flow of information through the different phases of a systematic review to map out the number of records identified, included and excluded, including the reasons for exclusions. The 27-item checklist appears in Appendix A.1.

### 3.2.2 Research questions

The main goal of this SLR is to answer the first research question of this thesis:

| **RQ1: How is RBAC analysed using process mining methods?** |
|---|

To that end, I identified the following SLR questions based on the three corresponding objectives (1–3) in Table 3.1.

- **Q1: What are the process mining approaches used for RBAC analysis?** This is to identify suitable process mining methods for RBAC analysis (objective 1).
- **Q2: To what extent do existing approaches cover the analysis of RBAC?** This is to identify RBAC models and constraints using process mining discussed in the literature (objective 2).
- **Q3: How do the existing approaches address the use of process mining for RBAC analysis?** This is to clarify the practical side of RBAC analysis using different process mining methods, such as input, output and techniques (objective 3).

### 3.2.3 Source and primary works selection

I carried out an automated search of the following digital libraries and databases: ACM Digital Library, IEEE Xplore, SpringerLink, ScienceDirect, Scopus and Google Scholar. To make sure that the most relevant works are retrieved, search strings and keywords should be well-chosen. Therefore, based on the research questions, I used the following search strings with adaptations for some digital libraries: 'RBAC' AND 'process mining'; 'role-based access control' AND 'process mining'; and 'access control' AND 'process mining'.

Process mining is an emerging topic in academia, and the number of articles published, according to the term (allintitle: 'process mining'), are increasing, as can be seen in Figure 1.1. To ensure the coverage of process mining, I directly used the keyword 'process mining', as it is general enough to cover all process mining methods and techniques. However, to be more inclusive, I searched for the keyword across the entirety of the papers, not just their titles.

To include papers on RBAC models and constraints such as SoD, I used the keywords 'RBAC', 'role-based access control' and 'access control'. The first two were to cover RBAC-related works and the third was to include works on access control constraints. This is because several works studied the usage of process mining for security analysis, which includes access control checking without a major focus on RBAC. This allowed me to include methods to check general access control constraints, which provide wider options that can be applied to RBAC constraints.

### 3.2.4 Inclusion and exclusion criteria

I only considered works that presented approaches for using process mining for RBAC analysis in peer-reviewed papers published after 2000 and written in English. This is

because the term 'process mining' was first introduced in 2000 by Aalst [3]. When a publication had multiple versions, only the most recent one was included. Short papers with fewer than four pages and Master's theses were excluded, along with papers proposing RBAC analysis without considering process mining methods and those proposing process mining approaches without considering access control analysis.

### 3.2.5 Conducting the review

The data collection process is shown in Figure 3.1. To select eligible works, the search process was conducted in three phases. First, 426 works were identified after applying the search strings. Then, 93 works were selected based on titles and abstracts. Finally, the full text of each work was reviewed for further refinement. After applying the inclusion and exclusion criteria, the 27 works shown in Table 3.2 were selected.

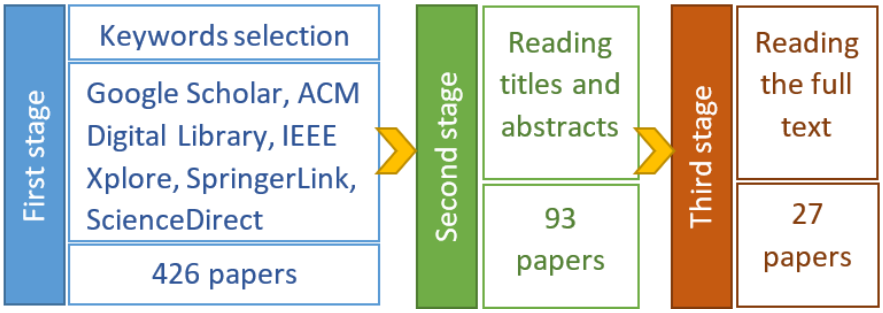



Figure 3.1: Data collection process.

| Ref | Year | Title |
|---|---|---|
| [13] | 2005 | Discovering social networks from event logs |
| [17] | 2005 | Process mining and verification of properties: An approach based on temporal logic |
| [14] | 2006 | Mining Staff Assignment Rules from Event-Based Data |

Table 3.2: The works included in the SLR.

| Ref | Year | Title |
|---|---|---|
| [15] | 2007 | Organizational modeling from event logs |
| [36] | 2007 | Life-cycle support for staff assignment rules in process-aware information system |
| [31] | 2008 | Towards comprehensive support for organizational mining |
| [86] | 2011 | Conformance checking of RBAC policies in process-aware information systems |
| [87] | 2011 | A business process mining application for internal transaction fraud mitigation |
| [88] | 2012 | An ontology for workflow organizational model mining |
| [89] | 2012 | Performative-based mining of workflow organizational structures |
| [90] | 2012 | On the exploitation of process mining for security audits: the conformance checking case |
| [91] | 2012 | Deriving Process-Related RBAC Models from Process Execution Histories |
| [92] | 2013 | A Case Study on the Suitability of Process Mining to Produce Current-State RBAC Models |
| [93] | 2013 | Delta analysis of role-based access control models |
| [94] | 2013 | Business models enhancement through discovery of roles |
| [95] | 2013 | On the exploitation of process mining for security audits: the process discovery case |
| [96] | 2014 | Process mining from the organizational perspective |
| [97] | 2016 | Mining resource assignments and teamwork compositions from process logs |

Table 3.2: The works included in the SLR.

| Ref | Year | Title |
|---|---|---|
| [98] | 2016 | Process Analysis as First Step Towards Automated Business Security |
| [99] | 2018 | From Security-by-Design to the Identification of Security-Critical Deviations in Process Executions |
| [100] | 2018 | Mining Resource Community and Resource Role Network from Event Logs |
| [101] | 2018 | Mining Expressive and Executable Resource-Aware Imperative Process Models |
| [102] | 2018 | Mining team compositions for collaborative work in business processes |
| [103] | 2018 | Linking data and process perspectives for conformance analysis |
| [104] | 2019 | History-Aware Dynamic Process Fragmentation for Risk-Aware Resource Allocation |
| [105] | 2020 | The RALph miner for automated discovery and verification of resource-aware process models |
| [106] | 2020 | Conformance Checking: Workflow of Hospitals and Workflow of Open-Source EMRs |

Table 3.2: The works included in the SLR

### 3.2.6 Evaluation criteria

To assess the identified methodologies, I formulated three criteria that directly address the research questions in Section 3.2.2 and reviewed the relevant literature according to these criteria.

To examine the extent to which the approaches address the process mining methods used for the analysis (Q1), I formulated the first criterion:

**CR1: Process mining principles.** This criterion discusses three types of process mining and the methods that were used in the works.

To examine the extent to which the approaches address RBAC (Q2), I formulated the second criterion:

**CR2: RBAC principles.** This criterion discusses the RBAC models and constraints that were included in the works, and whether the works considered hierarchical and TRBAC approaches.

To characterise the extent to which existing approaches address the process of the analysis (Q3), I formulated the third criterion: beginitemize

**CR3: Practical side.** This criterion considers the input, output, techniques and procedure of the analysis.

## 3.3 Results

This section presents an analysis of the body of literature identified using the process presented in Section 3.2.

### 3.3.1 Overview

This analysis of the literature shows that the current research landscape is fragmented. A variety of methodologies have been proposed to tackle the use of process mining in RBAC analysis, each discussing it with respect to different types. In particular:

**RBAC analysis.** Works under this methodology used process mining to specifically analysed RBAC models, such as using process mining for deriving RBAC models [91, 92]. Baumgrass et al. [91] proposed an approach to identify process-related RBAC models with enforced access control policies. Leitner et al. [92] studied the suitability of process mining to produce RBAC models by using three different methods. Conformance checking approaches can be used to verify RBAC models and constraints [86, 91, 93, 106]. For instance, RBAC constraints were checked on event logs by using LTL statements in [86, 106]. Leitner [93] proposed delta analysis of RBAC models, which compares a prescriptive RBAC model (how users are expected to work) with the actual RBAC model (how users are actually working) derived from event logs. As process mining approaches can be used to enhance process models, Havur and Cabanillas [104] used an event log, process model and RBAC model to enhance the resource allocation of a certain process activity at run time.

**RBAC context.** The focus of the works under this methodology was not on analysing RBAC models but either to analyse systems with enforced RBAC models or to analyse enforced access control constraints. Thus, the authors showed how the approaches can be designed or applied for RBAC analysis. Organisational mining can help in the discovery of RBAC configurations, and it was suggested that process discovery can help in the discovery of behaviours used for the configuration of task-based security policies. For instance, several approaches were proposed to check access control constraints on event logs [90, 98, 99, 103, 105]. Accorsi and Stocker [90] used LTL statements to check access control constraints. Likewise, some works discussed the verification of the four eyes principle, whereby a task must be performed by at least two people [17, 87]. The checking of these constraints is of importance when it is applied on models with RBAC policies. Other

process discovery approaches were introduced, and discussions of their ability to mine or verify RBAC models are presented in [14, 36, 31, 95, 96, 97, 101, 102].

### 3.3.2 Evaluation

This section reports the analysis of the selected literature against the evaluation criteria defined in Section 3.2.6. I present an overview of the results of criteria CR1 and CR2 in the following two sections. Tables 3.3 and 3.4 provide an overview of the results, where full coverage (■) indicates that a methodology explicitly addresses the criterion, partial coverage (◪) indicates that a methodology only deals with some aspects of the criterion and no coverage (□) indicates that the criterion is not addressed. Hereafter, I use the term 'covered' to indicate both full and partial coverage; the distinction between them is made explicit when required. Finally, this is followed by an evaluation of the CR3 criterion.

#### CR1: Process mining principles

**Process discovery:** Process discovery methods can help in the discovery of RBAC models from the actual execution of process models. This SLR found four mining approaches in particular that are used for discovery, namely organisational mining, role mining, staff assignment rules mining and social network mining.

*Organisational mining* is a broad term that focuses on the resources (the persons the system uses to carry out the tasks), whereas the organisational model represents the current organisational structure. Role mining discovers the relationships among users based on similar access permissions logically grouped to form a role. *Role mining* allows the discovery of the roles from event logs, whereas organisational mining refines the extracted roles and provides a background for the RBAC model. Based on these two approaches, Baumgrass et al. [91] developed an approach to produce a current-

| | | RBAC analysis | | | | | | RBAC context | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Criteria | Sub-criteria | [86] | [91] | [92] | [93] | [106] | [104] | [101] | [13] | [17] | [14] | [15] | [36] | [31] |
| Process mining types | discovery | ☐ | ■ | ■ | ■ | ☐ | ☐ | ■ | ■ | ☐ | ■ | ■ | ■ | ■ |
| | conformance | ■ | ☐ | ☐ | ■ | ■ | ☐ | ☐ | ☐ | ■ | ☐ | ☐ | ■ | ☐ |
| | enhancement | ☐ | ☐ | ☐ | ☐ | ☐ | ■ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
| Process mining methods | organisational mining | ☐ | ■ | ■ | ■ | ☐ | ◪ | ■ | ◪ | ☐ | ◪ | ■ | ■ | ■ |
| | role mining | ☐ | ■ | ■ | ■ | ☐ | ☐ | ☐ | ◪ | ☐ | ◪ | ■ | ☐ | ■ |
| | social network mining | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ■ | ☐ | ☐ | ☐ | ◪ | ■ |
| | staff assignment | ☐ | ☐ | ■ | ☐ | ☐ | ■ | ■ | ☐ | ☐ | ■ | ■ | ■ | ■ |
| | rules mining | | | | | | | | | | | | | |
| RBAC | RBAC rules | ■ | ■ | ☐ | ☐ | ■ | ■ | ◪ | ☐ | ◪ | ☐ | ☐ | ◪ | ☐ |
| | RBAC model | ■ | ■ | ■ | ■ | ☐ | ■ | ◪ | ◪ | ☐ | ◪ | ◪ | ◪ | ◪ |
| | Hierarchy | ■ | ☐ | ■ | ■ | ☐ | ■ | ■ | ■ | ☐ | ■ | ☐ | ■ | ■ |
| | Temporal | ◪ | ☐ | ☐ | ☐ | ☐ | ◪ | ◪ | ◪ | ◪ | ☐ | ☐ | ◪ | ☐ |

Table 3.3: An overview of the results (Part 1).

| | | RBAC context | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Criteria | Sub-criteria | [87] | [88] | [89] | [90] | [94] | [95] | [96] | [97] | [98] | [99] | [100] | [102] | [103] | [105] |
| Process mining methods | discovery | ■ | ■ | ■ | □ | ■ | ■ | ■ | ■ | □ | □ | ■ | ■ | □ | ■ |
| | conformance | ■ | □ | □ | ■ | □ | □ | □ | □ | ■ | ■ | □ | □ | ■ | ■ |
| | enhancement | □ | □ | □ | □ | ■ | □ | □ | □ | □ | □ | □ | □ | □ | □ |
| Process mining category | organisational mining | ◪ | ■ | ■ | ◪ | ■ | ■ | ■ | ◪ | □ | □ | □ | ■ | □ | ■ |
| | role mining | ◪ | ◪ | □ | □ | ■ | ◪ | ■ | □ | □ | □ | ■ | □ | □ | ■ |
| | social network mining | ◪ | □ | ■ | □ | □ | ◪ | ■ | ◪ | □ | □ | ■ | □ | □ | □ |
| | staff assignment | □ | □ | □ | □ | □ | □ | ■ | ■ | □ | □ | □ | ■ | □ | ■ |
| | rules mining | | | | | | | | | | | | | | |
| RBAC | RBAC rules | ◪ | □ | □ | ◪ | □ | ◪ | □ | ◪ | ◪ | ◪ | □ | ◪ | ◪ | ◪ |
| | RBAC model | ◪ | ◪ | ◪ | ◪ | ◪ | ◪ | ◪ | □ | □ | □ | ◪ | ◪ | □ | ◪ |
| | Hierarchy | □ | ■ | ■ | ■ | □ | ■ | ■ | ■ | □ | ■ | ■ | ■ | □ | ■ |
| | Temporal | ◪ | □ | □ | ◪ | □ | ◪ | ◪ | □ | ◪ | ◪ | □ | ◪ | ◪ | □ |

Table 3.4: An overview of the results (Part 2).

state RBAC model from event logs that represents the implemented RBAC model in a software system.

*Staff assignment rules* partly define the profile of agents capable of or eligible for performing an activity (i.e. the link between the process and organisational structures). For example, for performing the activity 'create bills', agents must possess the role of 'bookkeeper' while having 'computer skills'. Ly et al. [14] showed that knowledge about staff assignments can be used as input for further analysis. For instance, they argued that since the assignment rules are part of RBAC models, they can be used to improve the discovery of RBAC models.

*Social network mining* is based on the handover of work at the level of individual resources [3]; therefore, it is possible to build a social network based on the handover of work from one performer to the next. In contrast to the other three types, social networks focus on the relationships among individuals or groups, while the other types focus on the relationships between resources and the process. Song and Aalst [31] presented approaches to mine a group of resources sharing similar characteristics in the event log and derived two organisational models, one to identify resources that execute similar activities based on resources possessing similar skills and knowledge and the other to identify resources involved in the same case based on resources that work together on the same case. The authors argued that the models discovered can be converted to RBAC models.

It has been shown that a combination of these approaches can be applied together to discover an RBAC model [15, 96, 92, 93, 102]. More specifically, Leitner et al. [92] studied the suitability of process mining to produce RBAC models by using organisational mining, role mining and staff assignment mining. The authors showed that the three results successfully discovered the roles and produced similar RBAC models. Accorsi et al. [95] used process discovery approaches to build models from event logs so

that they could be used for security analysis. Thus, these models are to be compared with security and privacy requirements.

Process discovery methods can help in the discovery of the behaviours used for the design of task-based security policies [14, 36, 97, 102]. Schönig et al. [102] proposed a mining approach to extract rules on individual and group assignments for the process activities. The authors argued that the rules that are discovered can be used to build related RBAC models.

**Conformance checking:** The SLR found that process mining can be used to check if the actual RBAC model complies with the modelled one [86, 93] or to check any violations of access control constraints [86, 106]. More specifically, Leitner [93] considered delta analysis of RBAC models to compare a predictive RBAC model with an actual one to detect security violations. The study showed that comparing RBAC models can be complex and should only be performed by domain experts. Baumgrass et al. [86] proposed an approach to check RBAC policies on process execution traces. The approach converts an RBAC model into LTL statements that are then checked in the corresponding event logs. Likewise, Asare et al. [106] used LTL statements to check SoD in an RBAC policy.

Several publications explored the usage of process mining for access control checking by applying LTL statements in event logs [17, 36, 87, 90]. For instance, Accorsi and Stocker [90] checked SoD and BoD as well as authorisation constraints. Zahoransky et al. [98] proposed a toolkit to analyse security in process models as well as process logs. The toolkit provides predefined security patterns to analyse process models and temporal logic statements (LTL and CTL) to analyse event logs. Cabanillas et al. [105] proposed a tool for the discovery and verification of process models. The authors implemented the tool, which uses model checking methods for vitrification, and called it

the RALph miner.

Other works used alignments for checking [99, 103]. Salnitri et al. [99] proposed a tool-supported method for the analysis of security in event logs by using compliance checking. The authors used SecBPMN2 models to present process models with security constraints. Since most of the available checking methods use Petri nets, this tool converts the SecBPMN2 into a Petri net and mines a Petri net model from the event log. Checking is then performed by aligning these two models and determining which deviations are security critical. The authors implemented their approach as a part of the STS tool. Likewise, Alizadeh et al. [103] proposed an alignments-based approach for checking. However, their approach used data and process perspectives in the analysis. The approach used a process model, event logs and a CRUD matrix for the analysis. In particular, the approach analysed deviations with respect to the intended purpose (CRUD matrix) of the data and the context (process model) in which data are used.

**Enhancement:** Using process mining for enhancement aims to provide extension or improvement for a system. Havur and Cabanillas [104] proposed a risk-aware resource-allocation approach. Given an event log, process model and RBAC model, the approach can make probabilistic decisions ahead of time that satisfy a given probabilistic threshold.

#### CR2: RBAC principles

**RBAC model.** By the RBAC model, I mean the role assignment relationship. Role assignment approaches are limited, in that they only discover or verify roles, users in these roles and the activities that the roles usually perform. Access control rules such as SoD are not considered on the analysis. Several approaches have been proposed to analyse the role assignment relations from event logs [14, 31, 96, 92, 93]. Leitner et al.

[92] used three process mining approaches (organisational mining, role mining and staff assignment mining) to discover RBAC models and compared the results with the original RBAC model. The comparison was based on the number of identified roles, roles to subject assignments and tasks to role assignments. They showed that these approaches can help in discovering RBAC models. Furthermore, by using role and organisational models, staff assignment mining is the most suitable technique to establish task to role assignment relations. Leitner [93] proposed a delta analysis of RBAC models, comparing a prescriptive RBAC model with an RBAC model discovered from event logs. Leitner showed that because discovered RBAC models reflect the operational reality, some users, roles and permissions may not be included if they are not listed in event logs. For example, certain permissions are defined in RBAC models, such as a president having permission to launch a nuclear attack. However, although the president has permission, they might not often exercise it.

**Access control rules.** Access control rules describe the access privileges of resources for subjects. Several approaches have been proposed to discover access control rules [36, 91, 95, 97, 102]. It has been shown that staff assignment mining can discover access control rules such as SoD from event logs [36, 102]. Furthermore, several approaches check access control rules from event logs [17, 87, 90, 98, 99, 103, 105, 106]. Most of the proposed verification approaches focus on static access control rules by using LTL statements [17, 87].

Furthermore, the SLR found three approaches that combined both the RBAC model and access control rules by using historical executions of process models for verification [86], discovery [91] and enhancement [104]. Baumgrass et al. [86] checked RBAC policies by using LTL statements. These statements were generated from process-related RBAC models that contained the access control policies. This could be done by us-

ing a RBAC-to-LTL component that was produced in [91], where the authors presented algorithms to derive corresponding RBAC artefacts and entailment constraints from standardised XML-based log files. Baumgrass et al. [86] presented an approach to generate LTL statements from process-related RBAC models. These LTL statements can be used to check if the process executions recorded via event logs conform to the access control policies defined via a corresponding RBAC model.

**Hierarchical RBAC.** RBAC constraints can be on one level (flat) or considered a hierarchy [42]. The SLR found that hierarchical relations are usually considered in the discovery of RBAC models. However, a means to check hierarchical RBAC by using event logs is still missing.

**Temporal RBAC.** I found 14 works [86, 13, 17, 36, 87, 90, 95, 96, 98, 99, 102, 103, 104, 101] that discussed the temporal factor, but none considered TRBAC. Discussion of the temporal factor in these works was limited to the LTL environment without checking any TRBAC constraints.

### CR3: Practical side

In this section, I discuss the requirements and technologies used in the identified methodologies. The extended results can be found in Table A.2.

**Inputs.** Generally, event logs provide the input needed for the process mining approach to begin. However, some information might not be explicitly clear in the logs, such as when logs do not have role extensions or role hierarchy. Thus, to overcome this problem, the works included in the SLR enhanced the procedure with either extra data on top of an event log or sometimes a process model, such as an organisational model or an RBAC model. For instance, all the proposed methods under the staff as-

signment rules mining category required organisational information or models as inputs [14, 36, 92, 97, 101, 102, 104]. Another approach was to analyse the event logs to identify any necessary information, such as by mining the roles. Table 3.5 shows the required inputs.

| Input | References |
|---|---|
| Process model | [31, 94, 99, 103, 104, 105, 106] |
| RBAC model | [86, 93, 104] |
| Access control rules | [17, 36, 86, 87, 90, 98, 99, 103, 105, 106] |
| Organisational model | [17, 14, 36, 92, 97, 102] |
| Organisational information | [89, 91, 101, 104] |
| CRUD matrix | [103] |

Table 3.5: Inputs

Event logs usually lack the necessary information for the usage of process mining for RBAC checking, since it is difficult to find an explicit hierarchy of organisational units when analysing an event log. This might indicate that the procedure of using process mining for RBAC analysis is a complicated process that needs pre-processing steps to collect the needed inputs.

**Technologies.** The following tools and plug-ins were used for analysis in the identified methodologies. References for the tools used can be found in Table 3.6.

- ProM: Clearly, ProM was the most commonly used tool, as can be seen in Table 3.6. ProM is an open source framework that provides developers a platform with the process mining algorithms that is easy to use and extend [2]. It is thus widely used in the process mining field. ProM has a number of useful plug-ins. For instance, 'organisational mining', 'role hierarchy mining' and 'staff assignment mining' plug-ins have been used to derive RBAC models [91, 92]. The 'LTL checker' plug-in was used to check properties in [17, 36, 90, 87]. The 'originator

by task matrix' plug-in was used in [90] for checking authorisation constraints.

- MiSoN: The work in [13] used The MiSoN tool, which was integrated into the ProM framework.

- DpilMiner: Two works used the DpilMiner tool [97, 102]. The declarative process intermediate language (DPIL) is a declarative process modelling language that enables multi-perspective and multi-modal flexible processes to be specified, and the DpilMiner tool allows for the identification of DPIL models from event logs.

- RALph: Two works used the RALph miner [101, 105]. RALph is an expressive graphical notation for defining resource assignments that is independent of the process modelling notation.

- STS-Tool: STS-Tool enables design business processes and security policies to be graphically designed using SecBPMN2 and business processes to be verified against security policies. The work in [99] proposed an approach implemented as part of STS-Tool that allows for the identification of security-critical deviations in process executions using compliance checking analysis.

- SWAT: The SWAT toolkit was introduced in [98] as a platform for analysing workflows. In particular, SWAT is used to analyse process models and process logs to check rules such as SoD.

- Answer Set Programming (ASP): The authors of [104] implemented their approach using an encoding based on ASP, which is a form of declarative programming orientated towards challenging search problems.

- Role engineering tool: The role engineering tool provides support for the scenario

driven role-engineering process. This tool was used in [91, 92] to extract RBAC information from event logs.

| Tool | References |
|---|---|
| ProM | [17, 14, 36, 31, 86, 87, 88, 89, 90, 92, 94, 95, 103, 105, 106] |
| MiSoN | [13] |
| DpilMiner | [97, 102] |
| RALph | [101, 105] |
| STS | [99] |
| SWAT | [98] |
| ASP | [104] |
| Role engineering | [91, 92] |

Table 3.6: Tools

These eight tools were used successfully to achieve different goals, but it is crucial to note that there was no tool or plug-in that specifically used process mining for RBAC analysis. Combinations of these technologies with an amount of manual effort are required.

## 3.4 Data Analysis and Integration

This SLR shows that the existing research landscape is fragmented, since individual methodologies only partially address the usage of process mining for RBAC analysis, and that more attention should be devoted to this topic. For a start, it would be beneficial to develop a comprehensive framework that covers all the steps of employing process mining for RBAC checking. There is also the need for a thorough discussion about RBAC models (e.g. TRBAC, HRBAC) and how to derive and check them by using process mining. Finally, from a practical perspective, it is possible to analyse RBAC by using process mining; however, improvements to the inputs and tools would ease the process. This section summarises the main findings and the challenges identified and

provides some recommendations.

### 3.4.1 A thorough discussion and research of the usage of process mining for RBAC analysis is needed

Using process mining for RBAC analysis is a field that is largely unexplored. Only six works [86, 91, 92, 93, 106, 104] used process mining for RBAC analysis. The RBAC models and constraints discussed were simple (e.g. authorisation and SSoD). More complex RBAC models and constraints (e.g. TRBAC, HRBAC, DSoD) have not yet been considered, and the ability of process mining methods to mine complex RBAC models has not been tested, nor has the ability to check them. There are many approaches to mining the organisational model and the process model, but there is no tool to mine the RBAC model. Mining an RBAC model from an event log is a challenging process because it considers the organisational model, the process model, the connection between them as well the RBAC constraints. On top of that, considering the hierarchical relations and the temporal factor makes the process even more challenging. Moreover, the ability of process mining to analyse RBAC in various process models (e.g. with loops) has not yet been investigated.

**Recommendation:** In Chapter 4, I conduct practical experiments on the usage of process mining for RBAC analysis to verify and complement the findings of the SLR, thereby focusing in particular on more complex models and RBAC constraints (e.g. TRBAC, DSoD and DBoD) that have not yet been considered. I start with an RBAC model and consider temporal factors (e.g. TSoD) and then make the system more complex by considering, for example, loops. I check the access control constraints with the existing methods to verify the capability of these methods. Furthermore, in Chapter 5, I propose a multi-perspective conformance checking approach, apply it a case study and

discuss the results.

### 3.4.2 The practical side

From a practical point of view, using process mining for RBAC analysis has two main challenges, one concerning inputs and the other regarding tools.

#### Inputs

The first challenge concerns the lack of information on event logs. Although event logs contain information about time, actions and resources, it is difficult to find an explicit hierarchy of organisational units when analysing an event log. Thus, most of the works that considered RBAC used existing organisational models to overcome this challenge. However, one drawback of using an existing organisational model as input is that such a model could be outdated or might have changed, and this would affect the analysis. Moreover, the quality of the derived rules depends greatly on the quality of the organisational model. Another way to solve the problem of a lack of information in the event logs is to mine RBAC models from the event logs, but mistakes or the incompleteness of the model may lead to less meaningful rules.

**Recommendation:** It would be useful to 1) improve the mining of the necessary information from the log as a pre-process step for the usage of process mining for RBAC analysis and 2) introduce a way to compare a derived RBAC model with an existing one (e.g. delta analysis) to improve the derived one so that it can be used as a better input for the analysis.

In Chapter 4, I introduce a pre-process step in the proposed life-cycle and discuss the required inputs and existing methods to achieve that.

**Technologies**

The second challenge is the lack of tools for using process mining for RBAC analysis. In fact, as I have already remarked, the SLR identified that there is no specific tool for checking RBAC policies by using process mining methods. The most commonly used approach for checking properties by using process mining is the LTL checker ProM plug-in, but it has only been used for SoD and BoD, and more complex constraints, such as DSoD, DBoD or TRBAC constraints, have not been considered.

**Recommendation:** The lack of adequate tool support suggests that it is necessary to develop new methods and tools designed for RBAC analysis. These methods should be able to check more complex RBAC constraints (e.g. hierarchical relations between roles and constraints with temporal factors) as well as constraints such as DSoD that require checking roles instead of resources.

In Chapter 6, as work in progress, I introduce a plug-in that I developed to enrich the event log with the required information.

### 3.4.3 A comprehensive framework is needed

The usage of process mining for RBAC analysis sounds promising because it reflects what is really happening in the system. However, the principles extracted in this SLR are scattered across different approaches and works, and a comprehensive framework is missing. In fact, this lack of systematisation is reflected in the current body of research, where some researchers have focused on the usage of process mining for security checking in general and others on the mining of organisational models. Neglecting the correlations between the discovery process and the checking process is problematic when an analyst loses the link between the use of best practice and the objective of using process mining for RBAC analysis. Moreover, this SLR shows that 'enhancement using RBAC'

is especially lacking in the current body of research.

This SLR showed that there is not yet a framework for the usage of process mining for RBAC analysis. Using process mining for RBAC analysis is a complex procedure that could benefit from using the three methods of process mining:

1. Discovery: This is to use process mining to mine roles and organisational models for building RBAC models and to mine assignment rules and social networks to identify enforced constraints from real executions. This mined information reflects the actual behaviour that could be used as input for further investigations.

2. Conformance: The conformance process of RBAC can be done in three ways: 1) by comparing an RBAC model with an event log of the same process; (2) by checking if defined RBAC constraints hold in an event log; and 3) by comparing a prescriptive RBAC model with a current RBAC model (delta analysis). The conformance process of RBAC constraints cannot be established from only the event log or the process model; rather, the event log needs to be preprepared. Data preparation can be a significant part of the process and may require things such as discovering the roles or discovering the RBAC model. Data preparation can be done in the previous step (discovery).

3. Enhancement: This is to improve the existing process model with the desired RBAC model. A unified and comprehensive framework can help the usage of process mining for RBAC analysis.

**Recommendation:** The final recommendation that emerged from this SLR is the need for a framework for employing process mining for RBAC analysis. This framework can explain the pre-processing step (which is most likely a discovery problem) and the actual checking process (which is a conformance problem), and the results can be used for the

(enhancement) reason.

The pre-processing step is a data preparation step for the process. Data preparation may be one of the most difficult steps in any machine learning project, since each dataset is different and highly specific to the project. This preparation can be done by applying discovery methods in the event log. This will mine the needed information so that process mining can be applied for RBAC analysis. In addition, this step will guarantee that the mined information is real and up to date.

The actual checking is a conformance problem that compares current with prescriptive information. A delay between the mining and conformance steps can affect the process of employing process mining for RBAC checking. This is because a system could change its behaviour or, say, a member of the organisation could change their job, which would make the mined information outdated. Thus, a framework that provides a mining step with a conformance one would be helpful.

Finally, the enhancement step, which is the procedure of enriching the system with the RBAC information, could be done to help the developer and the analyst better manage the system.

The definition of a unified and comprehensive framework opens new research paths. First, we need a better understanding of the usage of the three process mining methods for RBAC analysis. This understanding can provide the best practice for mining important information from the event log as well as the model, checking various RBAC constraints (e.g. SSoD, DSoD, TSoD) and enhancing the process model with this information.

In Chapter 6, as work in progress, I introduce a conceptual framework for employing process mining for RBAC analysis that consists of four main concepts: 1) the functional perspective, which discusses the control flow; 2) the data perspective, which focuses on the event logs; 3) the RBAC perspective, which considers the RBAC model and

constraints; and 4) the organisational background. The framework is yet to be evaluated through a user study.

## 3.5 Data Quality Assessment

Construct validity reflects the extent to which the phenomenon under study genuinely represents the area conceived by researchers and the subject being investigated, in line with the relevant research questions [107]. This section discusses construction, internal and external threats to validity and how they have been mitigated.

**Construction threats** involve an erroneous definition of criteria, which could lead to poor conceptualisations. To counter this, the criteria mirror the RBAC and process mining principles. To approach the system design scope adequately, I considered process mining methods and categories as well as RBAC Principles.

**Internal threats** concern the criteria coverage. Determining the coverage for some methodologies required a degree of interpretation. I performed the evaluation by myself, although a single point of view might lead to biased decisions. To mitigate this threat, I elaborated on well-defined criteria to support my decisions as objectively as possible.

**External threats** are related to the completeness of this review. The initial list of items originated from different sources (Scholar, ACM Digital Library, IEEE Xplore, SpringerLink, ScienceDirect). Moreover, I scoped this search based on publication types (article, conference paper, book chapter). This limited the scope, as other relevant methodologies may have been published elsewhere. I countered these threats by performing a snowballing step, which enhances the completeness of the search process [108], allowing the gathering of publications not indexed in

the used sources or that do not match the search criteria.

## 3.6 Related Work

The need for methodologies to analyse RBAC constraints by using process mining has been recognised by both academia and industry over recent years. Some reviews have already attempted to summarise the resulting literature, and I briefly discuss here the ones most relevant to the aim of this chapter, namely two SLRs [109, 110] and two systematic mapping studies (SMS) [111, 112]. SLRs and SMSs aim to make evidence synthesis as transparent, objective and comprehensive as possible. However, SLRs go into depth, while SMSs are concerned with covering a wide field. Table 3.7 provides a comparison of this SLR with these works.

| Aspect | [109] | [110] | [111] | [112] | This SLR |
|---|---|---|---|---|---|
| Process mining types | ◪ | ■ | ■ | ■ | ■ |
| RBAC constraints | ◪ | ◪ | ◪ | □ | ■ |
| Hierarchical RBAC | ◪ | ◪ | □ | ◪ | ■ |
| Temporal constraints | ■ | □ | □ | ■ | ■ |
| Technologies | □ | □ | □ | □ | ■ |

■: Full coverage ◪: Partial coverage □: No coverage

Table 3.7: Comparison of this SLR with related work.

Compared to this SLR, none of the existing literature reviews provided a comprehensive analysis with respect to the coverage of all the types of process mining and RBAC analysis. For instance, the SLR in [109] defined and classified security and security controls in PAIS, which manage and execute operational processes involving people, applications and/or information sources, on the basis of process models. The work studied security issues in PAIS, rather than analysing RBAC by process mining. Thus, requirements, methods, technologies and practical analysis were not included. The SLR

in [110] discussed the common ways of implementing process mining techniques in the field of security in general. An initial assessment of the state of the art in the research area of human resource allocation in BPM and process mining was provided in the SMS in [111].

The SMS in [112] provided an overview of process mining; identified relevant areas of contribution in process mining, as well as the most discussed algorithms; and reported application domains among different business segments. It discussed role hierarchies in the context of RBAC. In contrast, my literature review provides a more comprehensive overview of existing research efforts on methodologies tailored to the usage of process mining for RBAC analysis. This SLR was designed specifically to explore the usage of process mining for RBAC analysis, including special cases such as hierarchical and TRBAC, and consider these in process models with repeated activities.

## 3.7 Conclusion and Next Steps

This SLR investigated process mining usage for RBAC analysis by tackling three main research questions. I examined 27 works and characterised the extent to which existing methodologies use process mining for RBAC analysis.

The existing methodologies were grouped into two main categories, one using process mining for RBAC analysis and the other using process mining in the context of RBAC and access control. Additionally, existing works were grouped into three main types: RBAC principles, process mining principles and the practical side. Thus, I explored the main characteristics of using process mining for RBAC analysis and observed that this fragmented landscape entails a lack of concrete, systematic and holistic methodologies to employ process mining for RBAC analysis, hampering the adoption of existing approaches by practitioners and the development of new methodologies by

researchers. I concluded this SLR by providing recommendations for future work to fill these gaps and turn process mining for RBAC analysis into a mature approach.

This SLR is the opening part of this thesis and identifies its scope according to the following six key areas: 1) the challenges of applying process mining; 2) organisational background and its role in the process; 3) the essential requirements for conducting the analysis; 4) the most suitable checking approaches; 5) the role of contextual information in the analysis; and 6) the tailoring of the analysis of RBAC using process mining. Chapters 4 and 5 of this thesis follow these key areas and the recommendations of this SLR.

# Chapter 4

# AN EXPLORATORY CASE STUDY ON THE CONFORMANCE CHECKING OF RBAC

## 4.1 Introduction

This chapter introduces an exploratory case study to understand how to employ process mining for RBAC analysis while uncovering limitations and identifying improvement opportunities. Chapter 4 has been designed to address Goal1, RQ2 and objectives 4–5 of this thesis (Table 4.1).

### 4.1.1 Motivation

Chapter 3 explored the main characteristics of using process mining for RBAC analysis by conducting an SLR on the growing literature, analysing 27 publications. This analysis showed there to be growing interest in process mining and RBAC, yet it observed that this fragmented landscape entails a lack of concrete and systematic methodologies.

| **Goal 1** | **Research question 2** | **Objectives** |
|---|---|---|
| To investigate the analysis of RBAC properties on organisations by using process mining methods. | To what extent can the existing process mining approaches be used for RBAC analysis? | 4. To determine the challenges when using process mining on complex RBAC constraints.<br>5. To uncover the limitations of using process mining for RBAC analysis. |

Table 4.1: The goal, research question and objectives addressed in Chapter 4.

In particular, it uncovered the following limitations:

**RBAC constraints:** The RBAC models and constraints discussed in the literature are simple (e.g. authorisation and SoD). More complex RBAC models and constraints (e.g. TRBAC and SSoD) have not yet been considered, and the ability of process mining methods to check complex RBAC models remains untested.

**Analysis:** The analysis process was not discussed, particularly the required knowledge and input to check RBAC. Furthermore, the impact of the selected input on the checking methods and, subsequently, the expected results were not discussed.

These findings motivated me to conduct an exploratory case study to determine how the process works while seeking new insights and generating ideas for my next research step. Thus, this chapter addresses this gap by practically experimenting with the process, focusing on compliance checking in ProM [2]. This work is applied on a simulated and real-life event logs. The investigation follows the guidelines of Runeson et al. [113] for conducting and reporting exploratory case studies.

### 4.1.2 Contributions

The main contribution of this chapter is *an exploratory case study on the conformance checking of RBAC*. More specifically, I provide two concrete contributions:

- **Evaluation:** I implement the case study on two event logs, namely synthesised and real-life logs. Thus, the case study is evaluated by real-life data from incident and problem management processes supported by Volvo IT's VINST system (available in [27]).
- **Life-cycle:** I propose a life cycle of the checking process. The life cycle consists of five main stages and can be used as a guideline for designing conformance checking on RBAC.

### 4.1.3 Outline

Section 4.2 introduces the methods and objectives applied in this case study. Section 4.3 describes the data collection process, introduces the process design and explains the RBAC model considered. The implementation of the experiment is discussed in Section 4.4. Section 4.5 describes the process of investigating the quality of the considered data. The experimental findings are discussed in Section 4.6, and the related work is discussed in Section 4.7. Finally, Section 4.8 presents the conclusion.

## 4.2 Methodology

A case study is a well-known research methodology in information science and software engineering [114, 113]. Yin et al. [114] defined a case study as an empirical method to investigate contemporary phenomena in their context. Runeson et al. [113] identified

four purposes of research: 1) exploratory, to find out what is happening, seek new insights and generate ideas and hypotheses for new research; 2) descriptive, to portray a situation or phenomenon; 3) explanatory, to find an explanation of a situation or a problem; and 4) improving, to try to improve a particular aspect of the studied phenomenon.

A case study is the most appropriate research methodology for this investigation; since its primary objective is exploratory, it enables a flexible design and the collection of qualitative data. The work in in this chapter follows the guidelines of Runeson et al. [113] in designing and performing case studies by conducting the five main steps of the process:

- **Case study design:** To define the objectives and plan of the case study (given later in this section);
- **Preparation for data collection:** To define the procedures for data collection (Section 4.3);
- **Evidence collection:** To use the data from the studied case (Section 4.4);
- **Analysis of collected data:** To use suitable analysis methods (Section 4.6);
- **Reporting:** To report the case study (writing this chapter).

Figure 4.1 provides a visual presentation of the steps, and thorough discussions of each step are provided in the related sections.

### 4.2.1 Research questions

The main goal of this case study is to demonstrate the feasibility of conformance checking as a tool for RBAC checking; hence, it is to answer the second research question of this thesis:

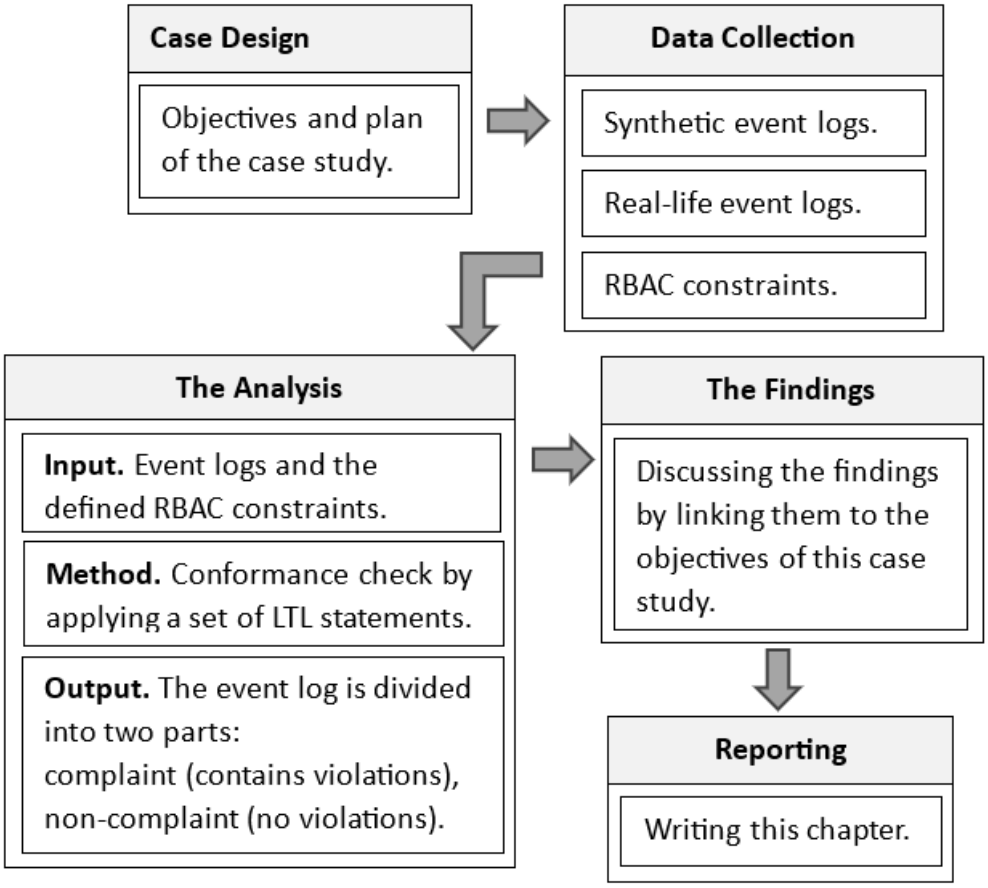



Figure 4.1: The methods of the case study in Chapter 4.

**RQ2: To what extent can the existing process mining approaches be used for RBAC analysis?**

To this end, I identified the following questions based on the two corresponding objectives (3 and 5) in Table 4.1:

- Q1: Does conformance checking generally allow the checking of RBAC constraints, and what are the limitations?
- Q2: How can process mining check RBAC, and how can it be improved?

This chapter aims to understand how to employ process mining for RBAC analysis while uncovering the limitations and identifying improvement opportunities. Thus, the

study is exploratory.

## 4.3 Data Collection

### 4.3.1 Synthetic event log

Process mining technology is generic and can be used in any domain [24]. Using process mining in healthcare is a promising direction that has been studied in the literature [115, 116, 117]. In [117], Martin et al. highlighted the potential of process mining in healthcare organisations due to their societal structures and complex nature.

This chapter conducts a case study based on the real-life scenario of a hospital's A&E process for handling patients, as obtained from and validated in interviews with doctors. Through the interviews, I gained knowledge about the steps of the A&E process and the possible scenarios. Also, I identified the roles required to perform certain tasks as well as possible and reasonable restrictions on the given permissions. This design went through iterations to provide suitable data elements that contain users, tasks and roles as well as the RBAC artefact that assigns users to roles and roles to tasks. To ensure a realistic setting, I checked the overall design with the doctors and made adjustments where required. For privacy, I simulated the log, including random violations of the RBAC constraints. Figure 4.2 provides an outline of the process of data collection at the A&E.

#### Process design

Figure 4.3 shows the Petri net presentation of this process (note that each activity is labelled with an abbreviation for the respective task). The process starts with the identification of a patient (IP). Then, the degrees of urgency are determined in triage (Tr), followed by the patient being discharged (Di_T) or admitted to the emergency room

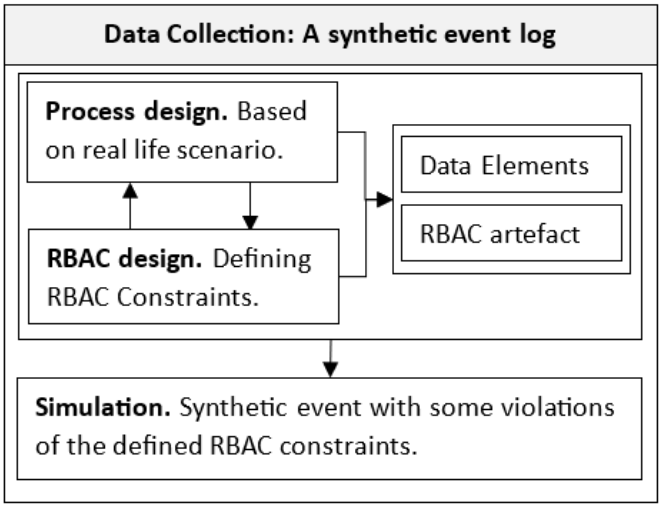



Figure 4.2: An outline of the data collection process.

(Ad_Er). Next, the patient is visited by a treating doctor (Vi), who discharges them (Di_V), requests lab tests (LR) or requests colleague consultation (CC).

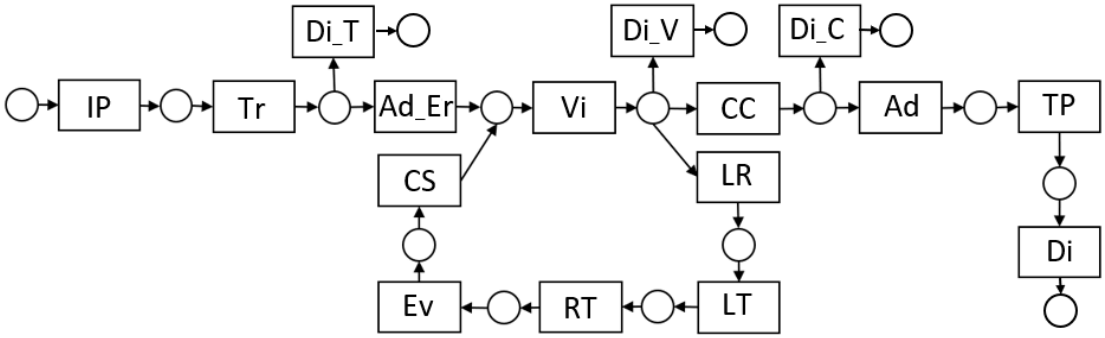



Figure 4.3: The Petri net of the A&E treatment process.

If the doctor requests lab tests, a phlebotomist will take a sample for the lab test (LT). Next, a technician runs the test (RT). The results are evaluated by a doctor (Ev) and co-signed by another doctor (CS). After that, the patient receives a doctor's visit (Vi).

If a consultation (CC) is requested, the consulted doctor will either discharge the patient (Di_C) or admit them to the hospital (Ad). Then, the doctor will prescribe a treatment plan (TP). Finally, the patient is discharged (Di).

### Data elements

I designed the data of this A&E process to consist of 16 objects, five roles and 13 subjects. The objects represent the transitions in Figure 4.3, while Table 4.2 shows the details of the data elements.

| ID | object | ID | Role | ID | Subject |
|---|---|---|---|---|---|
| Ob1 | Identify Patient (IP) | R1 | Doctor | S1 | Susan |
| Ob2 | Triage (Tr) | R2 | Nurse | S2 | Bob |
| Ob3 | Admission to ER (Ad_Er) | R3 | Receptionist | S3 | Clare |
| Ob4 | Discharge after Triage (Di_T) | R4 | Phlebotomist | S4 | Max |
| Ob5 | Visit (Vi) | R5 | Technician | S5 | Sara |
| Ob6 | Discharge after Visit (Di_V) | | | S6 | Abby |
| Ob7 | Lap Request (LR) | | | S7 | Tom |
| Ob8 | Lap Test (LT) | | | S8 | Alice |
| Ob9 | Run Test (RT) | | | S9 | Henry |
| Ob10 | Evaluate (Ev) | | | S10 | David |
| Ob11 | Co Sign (CS) | | | S11 | Harry |
| Ob12 | Colleague Consultation (CC) | | | S12 | Kate |
| Ob13 | Discharge after Consultation (Di_C) | | | S13 | Adam |
| Ob14 | Admission (Ad) | | | | |
| Ob15 | Treatment Prescription (TP) | | | | |
| Ob16 | Discharge (Di) | | | | |

Table 4.2: The data elements.

### RBAC artefact

I structured the RBAC model to include the data elements, as the 13 subjects were assigned to the five roles, which were then assigned to the 16 objects. Figure 4.4 represents the RBAC artefact of this case study. Some objects could be executed by more than one role, such as object RT (executed by a phlebotomist or technician). Also, some users were assigned to more than a role, such as Abby (assigned to Nurse and Receptionist).

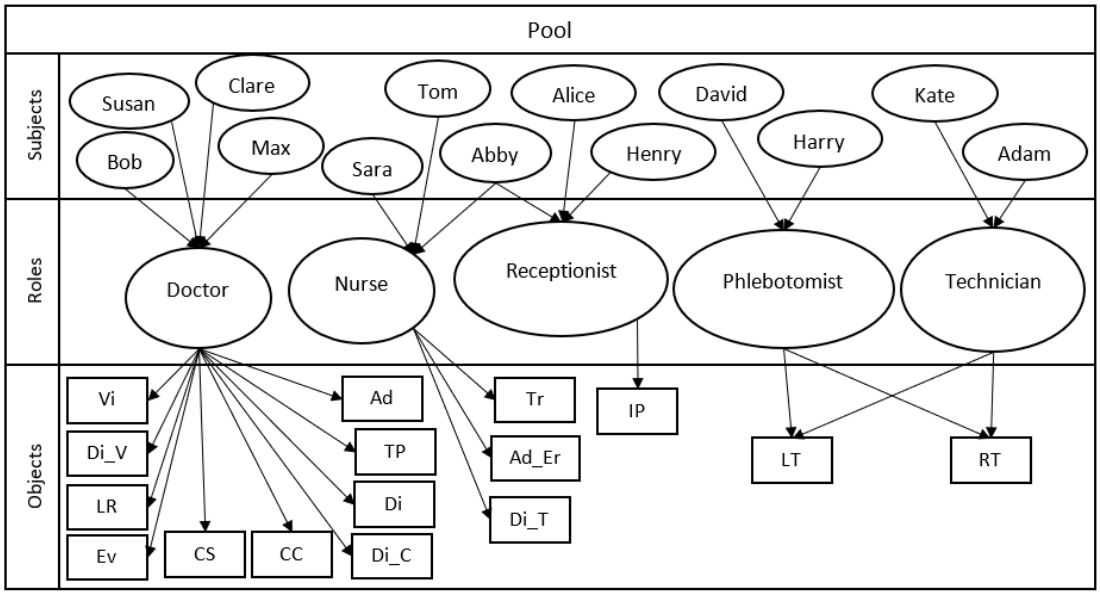



Figure 4.4: The RBAC artefact.

### RBAC constraints

Simon and Zurko [55] defined RBAC constraints on users, roles and activities. In particular, they defined operational SoD, which prevents a user from performing more than one activity in a case. This section defines constraints on activities; for example, SSoD states that two tasks must never be assigned to the same role or executed by the same subject, whereas DSoD states that two tasks must never be executed by the same subject. Thus, bound tasks can be assigned to the same role; however, they must be executed by different subjects. SBoD states that bound tasks must be executed by the same subject, whereas DBoD states that they must be executed by members of the same role. Later, I use process mining to check these constraints. I define the main constraints for the A&E process as follows:

- **Authorisation:** Each user should perform only permissible tasks.
- **DSoD:** The second signature (Cs) and evaluation (Ev) activities must be executed by two different doctors.

- **SSoD:** Running tests (RT) and performing lab tests (LT) must be executed by two different roles.
- **DBoD:** Discharge after triage (Di_T) must be conducted by the same nurse who performs the triage (Tr).
- **SBoD:** Requesting a test (LT) and visiting a patient (Vi) must be conducted by doctors.
- **TSoD:** The colleague consultation (CC) and discharge after the consultation (Di_C) activities must be executed by two different doctors if the time between these activities is less than 2 hours.
- **TBoD:** Triage (Tr) must be conducted by the same nurse who grants admission to the ER (Ad_Er) only if the time between these activities is longer than 1 hour.
- **Cardinality:** The number of nurses in one session should not exceed two.

**Process simulation**

A managed simulation approach was chosen to provide the basis for analysing RBAC properties within a realistic setting. I used the play-out process [3], which means that I used the process model to extract an event log. The background to and structure of event logs were introduced in Section 2.1.1. I designed the event log to include five attributes, as follows:

- **ID:** A unique value for each event;
- **Activity:** The operations;
- **Resources:** The subjects;
- **Time stamp:** When the activity is completed;

- **Role:** The role to which a subject belongs.

**Log simulation.** I used the CPN tool [39] to simulate the event log. In a nutshell, I created a process and then simulated the process executions to generate the event log. First, I created a Petri net of the A&E process to generate an SML file, which I used to define the five attributes in the event log, namely ID, activity name, resource, timestamp and role. Second, I defined appropriate activity transition probabilities to allow the execution of each branch in the model, controlling the admission rate and process distribution. Third, I assigned roles to activities and users to roles, as described in the RBAC artefact and RBAC requirements. Finally, I ran a simulation that generated an event log in CPN-XES format. The extracted file was then ready to be opened and explored as an event log file for further analysis.

This simulation was controlled and parametrised by typical business conditions, such as the acceptance rate and number of individuals involved. In this case, I simulated a 6-month period, the usual time considered by auditors. The simulation automatically added traces where violations happened to include deviations from the prescribed process.

**Generating violations.** After the simulation, I introduced behaviours that deviated from the prescribed process model and violated the security requirements. The simulation process determined resources randomly from the correct roles' groups to perform activities so that an authorised resource completed each activity. First, I selected some unauthorised resources to perform activities, to represent one of these deviations on the logs. I then introduced another violation by randomly assigning resources from the right roles to perform authorised activities. However, this random assignment held the authorisation constraint, so it could not guarantee the BoD and SoD constraints. Moreover, I used the 'Anonymize: Randomly shift time of traces/event' plug-in, causing the

activities to be delayed for a specific time range to violate the temporal constraints.

**Towards a real-life event log.** The simulation assumed the following constraints to ensure a realistic event log:

**Admission rate and process distribution.** A recent study stated that around 34% of ER patients are admitted to hospitals [118], which I considered in the simulation. I set it so that 17.6% of patients were discharged after triage, where no admission to the hospital is needed. The other part was where a nurse sees patients who need to be seen by a doctor, so I set it so that 24.9% of the patients were discharged after the visit and 23.3% required additional investigation but no admission. Finally, I set the last 34.2% to be admitted to the hospital.

**Noise.** Due to the logging process, real-life event logs usually contain rare and infrequent behaviour [3], which is called noise. The generated event log from this simulation followed the model perfectly, which is not the case in real life. To make the log similar to real life, I added noise to the logs using the 'Add noise to log filter' plug-in. This generates unexpected events and deletes random ones from the log. Using this plug-in, the level of noise can be controlled by setting a threshold. Clearly, lower noise provides fewer events and higher noise provides more events. It has been shown that noise of 5% provides realistic behaviour, as it does not differ from the original model, whereas noise of 20% provides a complex model and requires filtering steps [119]. Thus, I set the noise level for moving and adding an event to 5% for each, adding random missing and moving nodes.

**Alignments.** The relation between the control flow and the event log can be measured by alignments. To ensure that the generated event log reflected the A&E process, I used the 'Conformance checking of DPN' plug-in that took a Petri net file and

an event log as inputs. The results showed a fitness value of 99%, so all log traces were successfully replayed in the model. The remaining 1% could be associated with noise.

### 4.3.2 Real-life event log

For the experiments with real-life logs, I used a dataset taken from incident and problem management processes supported by Volvo IT's VINST system (available in [27]). The incident and problem management logs contained 7554/2306 cases and 65533/9011 events respectively. The primary goal of the incident management process is restoring a customer's normal service operation as quickly as possible when incidents arise, ensuring that the best possible levels of service quality and availability are maintained. The problem management system includes the activities required to diagnose the root cause(s) of incidents and to secure the resolution of those problems to enhance the quality of IT services delivered and/or operated by Volvo IT.

The process starts by receiving a case, which is then assigned to a suitable team. The case is then to be solved and closed. However, if the case is significant or cannot be solved by the first line, it can be transferred to the second or third line.

The process contains 13 objects, 23 function roles and 1440 users. Users can be assigned to more than one function role, and a function role has many users. Furthermore, users work in support teams that are assigned to one of the three lines. A support team can have more than one function role, and a function role can work in more than a team.

To identify suitable RBAC constraints for the analysis, I studied the log description and the manual of the VINST system in [120] and the behaviour analysis in [121, 122, 123]. Also, I studied the logs by applying a number of ProM plug-ins for filtering and discovering and was assisted by Excel. Furthermore, I used the IT security incident management policy and access control guide in [124] as guidelines for the identification

and extraction of RBAC constraints from the log. I found that the process had three interesting characteristics, which I used to identify the RBAC constraints, as follows.

First, a case can be completed during a call to the first line (help desk). Thus, the case is to be handled, solved and closed by the first line as fast as possible. In the incidents log, I found 1882 cases completed from calls. In all these cases, solving and closing the cases was done be the same person. Therefore, I identified the following constraint:

**DBoD:** In cases completed during calls, solving and closing the incident must be done by the same person.

Second, the goal of the incident management process is to solve a case within the first line. However, if an incident cannot be solved by the first line, it is transferred to a higher line. Thus, it is assigned to different roles or users.

The IT security incident management policy [124] states that if an incident needs to be escalated, it should follow the chain of command through the incident response command structure. Thus, the exact chain of escalation should be outlined in the IT system's incident response plan.

In the incident log, I found 4511 escalated cases. In such cases, a transferred case is always assigned to a different role. However, there is an exception when the transferred case can be assigned to the same role if the role is V3-2. Therefore, I identified the following constraint:

**SSoD:** When a case is transferred to a higher line from a role that is not V3-2, it must be assigned to a different role.

Third, knowing that organisations measure the total resolution time of an incident, people try to stop the clock on a particular case to decrease the total turnaround time for the completion of a task. One way to do this is by activating the sub status Wait–User. While there are legitimate reasons to use this status, some people misuse it to improve their own performance metrics. A main goal of the incidents management system is

to solve incidents as quickly as possible. In the incidents log, I found 2495 records of Wait–User. Spiegel et al. [122], studied the top 10 incidents that have the longest Wait–User time, where the wait time is more than 100 days. However, the average wait in the log was 7.74 days. By studying the log, I found that when the wait time was more that 90 days, the case would be closed by a different person. This can mean an extra check before closing a delayed case.

The IT security incident management policy [124] states that a separation of duties is used to help to reduce the possibility of malicious activity taking place without detection, requiring users to perform only one role in accomplishing a task, such as review, inspection or approval of a request. Therefore, I identified the following constraint:

**TSoD:** A delayed case with more than 90 days Wait–User time must be closed by a different user.

It is important to state that these constraints were estimated based on my analysis of the logs, the analyses in [121, 122, 123] and the guidelines in [124].

## 4.4 Data Analysis

### 4.4.1 Conformance checking and ProM

To check the RBAC constraints, I applied conformance checking approaches with the ProM [2] tool on the event logs. I used a number of ProM plug-ins, as I explain in the analysis. However, the main method that I used for the analysis was to apply LTL statements by using the 'LTL checker' plug-in.

#### LTL checker.

Given the linear nature of event logs, LTL has proven to be a good way to analyse them [125]. LTL language used in process mining was introduced by Aalst et al. [126]

and then used widely in the context of process mining for conformance checking. By using LTL, one can formulate expected or unexpected behaviours that can be directly compared with the event log.

The LTL checker plug-in within the ProM tool provides several predefined constraints. However, these constraints are not specified or categorised under the RBAC area. Furthermore, the LTL checker plug-in provides the option to define and customise the required constraints. After their definition, checking the constraints should be performed smoothly. The compliant traces should be separated from the non-compliant ones. These methods should be clear and easy to use.

The LTL formulas used in this work are adopted from the LTL language by Aalst et al. [126]. The language includes type definitions, renaming, formulas, sub-formulas, regular expressions, date expressions, propositional logic and temporal operators, such as next time ($\circ F$), eventually ( $\Diamond F$) and always ($\Box F$). The notation ate.X refers to some attribute X of an audit trail entry (ate). There are several predefined attributes, such as *ate.WorkflowModelElement*, which refers to the activity executed, while *ate.Originator* is the resource executing it and *ate.Timestamp* is the time stamp of the event. A complete description of the language was provided by Beer et al. [127].

Various RBAC constraints are used nowadays, including authorisation, SoD and BoD. To my knowledge, no comprehension method has been designed to translate RBAC constraints into a language used in process mining. Understanding these constraints and defining them in a usable and modifiable way can hugely improve the process, yet it requires a high level of understanding of the constraints and the RBAC model. I translated a number of RBAC violations into LTL formulas to check the RBAC constraints. The formulas are saved in an open-source LTL file ready to be used, and they can be modified to meet the desired needs, including time, originators and activities as variables. Hence, these variables can be modified to meet the required target, such as a

specific person or time, or they can be used 'as is' to check all available options. Figure 4.5 shows the basis of the definitions that all formulas share. The notation ate.X refers to some attribute X of an event in the event log. The ate stands for an audit trail entry, which, in this case, is an event. Lines 1–6 declare that these attributes can be used for quantification. Line 1 defines the type of event, while line 2 defines the resource executing an activity. Line 3 defines the organisational role to which a resource belongs, and line 4 defines the format of the timestamps of the event. Line 5 defines the activity executed, while line 6 defines an attribute of a process instance (pi) used to count the number of cases. Lines 8–19 rename these attributes with shorter names to be used more efficiently in the subsequent formulas.

```
1  set ate.EventType;
2  set ate.Originator;
3  set ate.Role;
4  date ate.Timestamp := "yyyy-MM-dd";
5  set ate.WorkflowModelElement;
6  number pi.numSimilarInstances;
7
8  rename ate.EventType as eventtype;
9  rename ate.EventType as event;
10 rename ate.Originator as originator;
11 rename ate.Originator as person;
12 rename ate.Timestamp as timestamp;
13 rename ate.Timestamp as time;
14 rename ate.Role as role;
15 rename ate.WorkflowModelElement as modelelement;
16 rename ate.WorkflowModelElement as activity;
17 rename ate.WorkflowModelElement as place;
18 rename ate.WorkflowModelElement as element;
19 rename pi.numSimilarInstances as freq;
```

Figure 4.5: Definitions in LTL

I created a file containing RBAC violations in LTL formulas, ready to be used in an LTL checker, along with event logs to verify the constraints. The file is open source and can be modified to meet certain targets and is available in Appendix C.

### Checking DSoD

The DSoD constraint in this work is described as 'the person who performs activity A should not perform activity B'. In other words, activities A and B should be performed by different people. So, if person P performs activity B, that means that P should never before have performed activity A.

The logical representation of DSoD is as follows:

$$\Box(Ex(Ag_1, A_2) \rightarrow \neg\blacklozenge Ex(Ag_1, A_1)) \tag{4.1}$$

Thus, the violation of DSoD can be explained as 'there is a person who performed activities A and B', which means that person P, who performed activity B, has also in the past performed activity A.

The logical representation of the violation of DSoD can be defined as

$$\Diamond(Ex(Ag_1, A_2) \wedge \blacklozenge(Ex(Ag_1, A_1))) \tag{4.2}$$

Next, I present the formula using only the future LTL operators. This step is important, since the LTL checker plug-in only uses future tense operators, so the formula equivalent to the violation of DSoD using only future operators is

$$\Diamond(Ex(Ag_1, A_1) \wedge \Diamond(Ex(Ag_1, A_2))) \tag{4.3}$$

To use this representation in the 'LTL checker' plug-in, I defined the formulas in Figure 4.6, which are equivalent to my logical representation.

Lines 1–3 describe a person who is performing activity A, but not B. This subformula checks that activities A and B are different activities, $A \neq B$. Lines 5–8 show the main formula to check for DSoD violation. This process checks if there is a person

```
subformula P_does_A-A_not_B( P: person, A: activity, B: activity ) :=
{}
<>( ( activity == A /\ ( activity != B /\ person == P ) ) );

formula exists_person_doing_task_A_and_B( A: activity, B: activity ) :=
{}
    exists[ p: person |
    ( P_does_A-A_not_B( p, A, B ) /\ P_does_A-A_not_B( p, B, A ) ) ];
```

Figure 4.6: Representation of the SoD constraint in LTL formulas.

doing two different activities.

In these formulas, I created two variables for activity, A and B. These variables can be assigned to any value of type strings. Thus, A and B can be given the names of activities that should not be performed by the same person. However, I did not define a variable for Person, since there is no need to specify the name of the person who performed the tasks. Instead, the function will check all the subjects to identify any violation. Since I described the violation of the C2 as formulas in LTL, the compliance of execution traces can be verified with the 'LTL Checker' plug-in.

Because SoD enforces restrictions on two activities, I begin by extracting traces where the two activities had occurred. Thus, I used an LTL formula to extract only traces that contained activities A and B, which can be defined as $\Diamond(A \wedge \Diamond B)$. On the one hand, this step is to reduce the candidate dataset; on the other, it excludes the case where the process ends by performing one of the activities, not the other. To do so, I used the function represented in Figure 4.7.

```
formula eventually_activity_A_and_eventually_B
( A: activity, B: activity ) :=
{}
  ( eventually_activity_A( A ) /\
  eventually_activity_A( B ) );
```

Figure 4.7: Checking the occurrence of two activities in a trace represented in the LTL formula.

### Checking SSoD

RBAC can ensure that a user cannot be a member of two roles during a session. This constraint is therefore similar to DSoD; however, instead of checking individuals, SSoD checks roles. The SSoD constraint in this work can be described as 'the role that performs activity A should not perform activity B'. In other words, activities A and B should be performed by different roles. So, if role R performs activity B, that means that the same role R should never have performed activity A in the past.

The logical representation of SSoD is as follows:

$$\Box(Ex(R_1, A_2) \rightarrow \neg\blacklozenge Ex(R_1, A_1)) \tag{4.4}$$

Thus, the violation of the property can be explained as 'there is a role that performed activity A and B', which means that role R who performed activity B has in the past also performed activity A.

The logical representation of the violation of SSoD can be defined as

$$\Diamond(Ex(R_1, A_2) \wedge \blacklozenge(Ex(R_1, A_1))) \tag{4.5}$$

Since the LTL checker plug-in uses only the future LTL operators, I present the equivalent formula for the violation of SSoD using only future operators, as follows:

$$\Diamond(Ex(R_1, A_1) \wedge \Diamond(Ex(R_1, A_2))) \tag{4.6}$$

Figure 4.8 presents the LTL formulas for checking SSoD violations.

Lines 1–3 describe a sub-formula that checks if a role is doing two different activities, A and B, and so $A \neq B$. Lines 5–10 show the main formula to check for a violation of SSoD. This formula checks if there is a role performing two different activities. This

```
1  subformula R_does_A-A_not_B( R: role, A: activity,
2  B: activity) :=
3  {}
4  <>((activity == A /\ (activity != B /\ role == R )));
5
6   formula Role_R_does_activity_A_and_B( R: role,
7   A: activity, B: activity ) :=
8  {}
9      ( Role_R_does_activity_A( R, A )
10 /\ Role_R_does_activity_A( R, B ) );
```

Figure 4.8: Representation of the SSoD constraint in the LTL formula.

function can be used to check SSoD, where a role performs two restricted activities.

In this definition, I created two variables for activities A and B. These variables can be assigned to any value of type strings. Thus, A and B can be given the names of activities that should not be performed by the same person. I also assigned a variable for Role. Hence, to check this property, two activities and a role must be assigned.

### Checking DBoD

The DBoD constraint describes the situation whereby two activities, A and B, must be performed by the same person. This means that if person P performs activity B, they must have previously performed activity A, where $A \neq B$. Thus, the logical representation of this property is as follows:

$$\Box(Ex(Ag_1, A_2) \rightarrow \blacklozenge Ex(Ag_1, A_1)) \tag{4.7}$$

Subsequently, the violation of DBoD is the case where these two activities are being performed by two different persons. This can be represented as person P performing activity B while person Q has previously performed activity A. The logical representation of the violation of DBoD can be seen as follows:

$$\Diamond(Ex(Ag_1, A_2) \wedge \blacklozenge(Ex(Ag_2, A_1))) \tag{4.8}$$

The equivalent logical representation of the violation of DBoD by using only future LTL operators is as follows:

$$\Diamond(Ex(Ag_2, A_1) \wedge \Diamond(Ex(Ag_1, A_2))) \tag{4.9}$$

Figure 4.9 shows the LTL function of the DBoD property. Assuming that there are many individuals, the sub-formula in lines 1–4 checks if an activity has been performed by one person but not by others. The main formula in lines 6–14 describes if two different activities have been performed by two individuals. The sub-formula is to be used in the main formula to check if the two performers are different, $P \neq Q$.

```
1  subformula A_done_by_P-P_not_Q(A: activity, P: person,
2  Q: person ) :=
3  {}
4  <>((person == P /\ ( person != Q/\ activity == A)));
5
6  formula exists_activity_B_done_by_person_P_and_activity
7  _A_done_by_person_Q(A: activity, B: activity) :=
8  {}
9      exists[ p: person |
10         exists[ q: person |
11                 ( A_done_by_P-P_not_Q( B, p, q )
12                 /\  A_done_by_P-P_not_Q( A, q, p) )
13             ]
14         ];
```

Figure 4.9: Representation of the DBoD constraint in LTL formulas

This function has two variables for activities, A and B. Thus, to run the checking, two activities must be assigned. There is no need to assign a variable for Person, as this function checks all the available persons in an event log. This function is ready to be used within the LTL checker plug-in.

**Checking SBoD**

SBoD is similar to DBoD; however, instead of checking persons, SBoD checks roles. I defined the violation of SBoD in the same way as in the previous constraints. Unfortunately, applying the predefined statements on ProM yielded no results. In other words, the result of the check would be 'no violation is found'. This is because the LTL checker is designed to check MXML event logs only. Thus, using it on logs with extensions such as XES logs may produce inaccurate results.

### 4.4.2 Synthetic event log

**Authorisation.** I checked the constraint by applying the 'Originators by Task matrix' plug-in and using the RBAC artefact of the A&E process in Figure 4.4. This plug-in provides a matrix of which subjects 'originators' performs which objects 'tasks'. To check the authorisation constraint, I used 'Originators by Task matrix' in ProM 5.2, which is not available in newer versions of ProM. Authorisation ensures that users should perform only the tasks that they have permission for. In RBAC, the authorisation method gives users access to information based on their role within an organisation. A violation occurs when an activity is performed by an unauthorised subject. Thus, to detect violations, I manually analysed the matrix with respect to the predefined roles. This enabled the detection of five occurrence of unauthorised access.

**SoD.** Considering the RBAC constraints in the A&E treatment process, there were DSoD and SSoD constraints formulating obligations for task executions.

**DSoD:** The second signature (Cs) and the evaluation (Ev) activities must be executed by two different doctors. To check this constraint, I applied the formulas in Figure 4.6 by adjusting A and B to Ev: evaluation and CS: second signature.

The LTL formula successfully identified the violations by extracting 35 compliant cases.

**SSoD:** Running the test (RT) and performing the lab test (LT) activities must be executed by two different roles. I used the formulas in Figure 4.8 to check SSoD. As the SSoD constraint states, Running the test (RT) and performing the lab test (LT) activities must be executed by two different roles. I assigned activities A and B to RT and LT, respectively. As for role, two roles were authorised to perform RT and LT: technician and phlebotomist, respectively. However, this function checks one role at a time. Hence, the checking must be done twice, once with R = Technician and other with R = Phlebotomist. The results identified a trace where RT and LT had been performed by technicians and another trace where RT and LT had been performed by phlebotomists.

**BoD.** Discharge after triage (Di_T) must be done by the same nurse who conducts the triage (Tr). For checking, I applied the formulas in Figure 4.9 by adjusting activities A and B to (Di_T): discharge after triage and (Tr): triage respectively.

Figure 4.10 shows some of the identified complaint traces as they were presented in ProM.

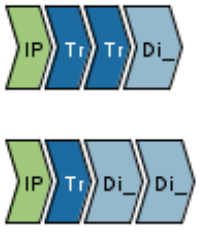



Figure 4.10: Traces that violated BoD.

**Temporal constraints.** To check for temporal constraints, I used the SCIFF checker plugin, since it is more capable of analysing the time factor. The SCIFF checker

is only available in ProM 5.2 and not in newer versions. Considering the RBAC constraints for the A&E treatment process, there were two temporal constraints (TSoD and TBoD).

**TSoD:**

The colleague consultation (CC) and discharge after consultation (Di_C) activities must be executed by two different doctors if the time between these activities is less than 2 hours. To check TSoD, I performed the SCIFF command in Figure 4.11 in the SCIFF checker. The results showed that there were three traces that violated the TSoD.

```
IF activity A is performed by O_A at time T_A
    having
    A equal to CC
  THEN
activity B should be performed by O_B  at time T_B
    having
    T_B after T_A
    and
    B equal to Di_C
    and
    O_B equal to O_A
    and
    T_B before T_A + [2 h, 0 ms]
```

Figure 4.11: Representation of the TSoD constraint using the SCIFF checker plug-in.

**TBoD:** Triage (Tr) must be done by the same nurse who carries out the admission to ER (Ad_Er) only if the time between these activities is longer than 1 hour. As with TSoD, to check this constraint, I applied the SCIFF command in Figure 4.12. The results showed that there were two traces that violated the TBoD.

**Cardinality.** This constraint states that the number of nurses in one session should be no more than two. In contrast to other constraints, it does not relate to a specific activity. A session here can be represented using traces in an event log, where each trace is a unique session. Thus, to check this constraint, each session needs

```
IF activity A is performed by O_A at time T_A
    having
    A equal to Tr
  THEN
activity B should be performed by O_B  at time T_B
    having
    T_B after T_A
    and
    B equal to Ad_Er
    and
    O_B different from O_A
    and
    T_B after T_A + [1 h, 0 ms]
```

Figure 4.12: Representation of the TBoD constraint using the SCIFF checker plug-in.

to be checked individually. Each trace should not have more than two nurses performing activities. This can be done by either of the following approaches:

1. **Using a trace, only the activities that nurses have permission to perform are to be checked.** This approach checks only the activities that are assigned to nurses, which are the only four activities in this model.

2. **Using a trace, all the activities in that trace are to be checked.** This approach is more inclusive and provides the bigger picture, as it checks all activities, not just the activities that nurses have permission to perform. A nurse or a member of a role can perform an activity that is not assigned to them in two cases, which could be 1) a gap in security that allows such behaviour or 2) special circumstances whereby an activity is delegated to an authorised person or role, such as in 'Break glass', which refers to a person who does not have access privileges to certain information to gain access when necessary.

The beauty of process mining is that it allows either approach to check the cardinality requirement. To check cardinality, I followed the second approach. Thus,

I defined two LTL formulas to check if three different individuals had performed three activities in a trace. Figure 4.13 shows these LTL formulas. I assigned Sara, Abby and Tom to A, B and C, as thy were the three nurses in this system. The results show that there were four traces that violated cardinality.

```
1  formula eventually_person_P( P: person ):=
2  {}
3      <>( person == P );
4
5  formula eventually_person_A_and_eventually
6  _person_B_and_eventually_person_C( A: person,
7  B: person, C: person) :=
8  {}
9    ( eventually_person_P( A ) /\ (eventually
10   _person_P(B) /\ eventually_person_P(C)));
```

Figure 4.13: An LTL formula to check if three different users performed three tasks.

### 4.4.3 Real-life event log

**Authorisation.** Since the data were large, with 1440 users, 23 roles and 13 activities, the 'Originators by Task matrix' would not be an efficient method to check authorisation. Instead, for each task in the event log, I filtered the log twice, once with the event name and once with roles to build a role by task matrix. The matrix showed that there was no occurrence of unauthorised access.

**SoD.** Considering the identified constraints, there is SSoD. When a case is transferred to a higher line from a role that is not V3-2, it must be assigned to a different role. The activity 'Accepted+In Progress' represents the transfer process, while the activity 'Queued' represents a new assignment. Thus, to check this constraint, I began by extracting cases in which these two activities had been performed by the same user by applying the formula in Figure 4.8 with modifying activities to the events when needed. Finally, I checked the role performing these two activities

and eliminated it if it was V3-2.

**BoD.** Considering the identified constraints, there is DBoD. In the cases completed during calls, solving and closing the indecent must be done by the same person. To check this constraint, using LTL statements, I extracted cases ending with 'Completed+In Call'. I then checked whether a case had been solved and closed by a different person by using the LTL statement in Figure 4.9 and assigning A and B to Accepted and Completed, respectively.

**Temporal constraint.** There is a temporal constraint on the log as TSoD. A delayed case with more than 90 days of Wait–User time must be closed by a different person. To check this constraint, first by using LTL statements, I extracted cases where Wait–User occurred. Then, using the SCIFF plug-in, only cases with a wait time of more than 90 days were selected. Finally, using LTL statements, I checked whether a case had been resolved and closed by the same person.

## 4.5 Data Quality Assessments

### 4.5.1 12 guidelines for logging

In his book [3], Aalst emphasised that the most important building block of process mining is the event log. If the data in the event log cannot be trusted, the results will be less valuable. Twelve guidelines for logging that aim to provide high-quality event data were proposed in [128]. To ensure the quality of the real-life event logs, I applied the 10 relevant guidelines, as follows:

**GL1** *Reference and attribute names should have clear semantics.*All events in this log were interpreted in the same way.

**GL2** *There should be a structured and managed collection of reference and attribute names.* All events in this log followed the same structure and had clear sets of attributes and references.

**GL3** *References should be stable.* Each event had a unique identifier. Thus, an event could be distinguished by its identifier.

**GL4** *Attribute values should be as precise as possible.* Each attribute followed the same format and data type in all events.

**GL6** *Events should be at least partially ordered.* The events were ordered through an attribute indicating the event's timestamp.

**GL7** *If possible, transitional information should be stored.* Each event had transitional information, such as assign and wait.

**GL8** *Perform regularly automated consistency and correctness checks to ensure the syntactical correctness of the event log.* I checked missing references and attributes by filtering the log.

**GL10** *Do not aggregate events in the event log used as input for the analysis process.* The analysis was applied on the event data without a prior aggregation.

**GL11** *Do not remove events and ensure provenance.* I clearly stated any filtering of the data.

**GL12** *Ensure privacy without losing meaningful correlations.* The data provided information that enabled the analysis without revealing any private or sensitive information.

GL5 checks uncertainly with respect to the occurrence of events, and GL9 ensures that the logging is changed without reporting. These two guidelines can be applied

during the logging and cannot be checked after that. Thus, I assumed they were satisfied in these logs.

### 4.5.2 OneR method

I applied data mining methods to ensure the quality of the synthesised event log, in particular the OneR method. OneR is a well-known supervised-learning data mining method. Weerdt et al. [129] showed the benefits of using supervised approaches in event logs, as they often leverage additional domain knowledge. Supervised learning assumes labelled data, and in event logs, the labelled data are the events' attributes.

OneR is a simple yet accurate algorithm that generates one rule for each predictor in the data and then selects the rule with the smallest total error as its 'one rule'. To create a rule for a predictor, I constructed a frequency table for each predictor against the target. In this case study, I used only three attributes from the event log, namely role, resource and activity.

I chose the role attribute as the response variable and the resource and activity attributes as the predictor variables. Thus, this algorithm predicted the roles based on the resource and the activity. I applied this using Python; the source code can be found in Appendix B.

The results showed that the number of correct predictions was 10717.0 out of 10723, which presents 99% accuracy. Since the algorithm successfully predicted the roles based on the resource and activity attributes, it was indicated that the event log was well structured and simulated.

## 4.6 Findings

Recall the research questions:

- Q1: Does conformance checking generally allow the checking of RBAC constraints, and what are the limitations?
- Q2: How can process mining check RBAC, and how can it be improved?

These research questions have been addressed as follows: Q1 was answered in Sections 4.4 and 4.6.1, and Section 4.6.2 answered Q2.

### 4.6.1 Detecting violations

This chapter answers Q1 by showing that process mining approaches, and conformance checking in particular, can be successfully used to check the desired RBAC constraints for the A&E and the incident and problem management processes. However, I identified the following theoretical limitations:

**Required inputs** in this case study imply that checking RBAC requires knowledge beyond the event logs, such as the organisational model and the RBAC artefact. The required input for this process has not previously been identified.

**Designing the checking process** depends on the available inputs, as they impact the selecting methods and, subsequently, the achieved results. Guidelines for checking RBAC by process mining have not previously been identified.

**Exploiting the capacity** of process mining capacities has not yet been tested. Therefore, in this case study, I only checked the enforced constraints without considering other aspects, such as control flow.

Furthermore, I identified the following technical limitations:

**No tools for checking RBAC,** extensive adjustments or manual efforts were required. Furthermore, as the checking of RBAC required access to organisational infor-

mation, a tool allowing data detection, such as an 'LTL checker' that would be suitable for more complicated event logs, was missing.

**Needed skills** had to be defined at a low level to enable checking with a robust knowledge of the representation. In this case study, I used LTL formulas. Furthermore, skills for designing and editing Petri nets are required by analysts.

**Special events** either in the design or subsequently in the analysis. Control flows can consist of sequential steps, repetition (loops) and conditions. Thus, the impact of control flows on the checking has not yet been tested.

- **Duplicate activities,** also known as recurrent activities, occur when two or more activities have the same name. Since an event is usually identified by the name of the activity, more than one event in this case would refer to the same activity. To this end, a unique event identifier was used to distinguish between the events. However, not all plug-ins allowed for an event identifier, which should be considered when developing new plug-ins.
- **Repeated activities** occur when an activity occurs more than once, which usually happens by using loops. Duplicated and repeated activities could have affected the checking of constraints consisting of two activities, such as SoD and BoD, when one of the activities occurred more than once. Hence, human effort was required to refine the checking and interpret the results.

To provide solutions for the theoretical limitations, in Chapter 5, I check the RBAC constraints by linking two perspectives (control flow and data). Finally, in Chapter 6, as work in progress, I contribute to the technical side by developing a plug-in to enrich event logs with the required information.

### 4.6.2 Life-cycle

Process mining for RBAC analysis is a largely unexplored field. This chapter answered Q2, having identified through the case study that the process of checking RBAC is not straightforward, with no custom tools to check RBAC constraints, no entrance point, no exit gate and no map provided. Therefore, guidelines are needed. This section proposes a life cycle employing process mining for RBAC checking by adapting the methodology from [130]. Weske [130] identified life-cycle stages by showing their logical dependencies. This life cycle applied the following four main stages: 1) design and analysis, where the requirements and environment are identified; 2) execution, where the implementation takes place; 3) evaluation, where the process in reviewed and evaluated; and 4) change, where the outcome can be used to enhance the process. Figure 4.14 illustrates a life cycle that consists of the following five stages:

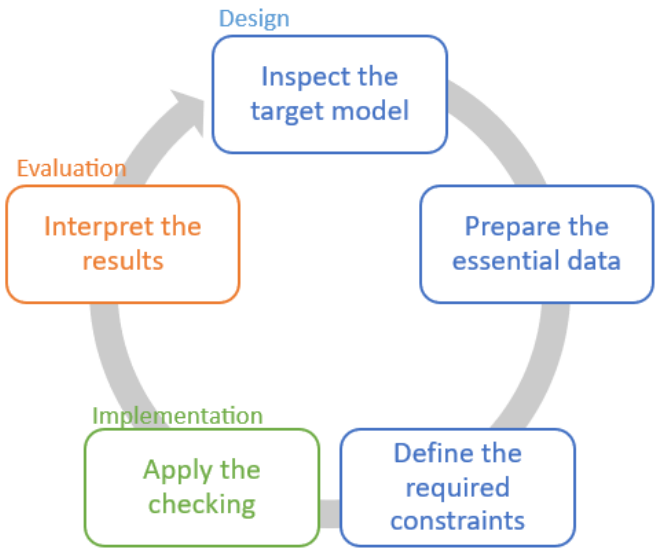



Figure 4.14: Life-cycle support for employing conformance checking for RBAC analysis.

**Inspect the target model.** This aims to decide what information is needed. The analysis type reflects the desired data perspectives. Since analysing RBAC requires

knowledge of the control flow and the organisational structure, I found it essential to spend time understanding the system. The process begins by realising the process model, event log, RBAC requirements and artefact.

Process mining approaches start from event logs that reflect reality. Log inspection helps to understand the available event log and make the most of it. Available event logs typically include the activities executed, temporal information and resources in charge of their execution. With such data, a process's functional, behavioural and organisational perspectives can be identified [101]. Chapter 2 discussed the definitions of event logs and their components in detail. The reason for inspecting logs is to become familiar with the process represented by the event log at first glance. This step can help to identify the required information.

**Prepare the essential data.** Although event logs contain much information, such as the ID and task, they do not usually show the information necessary for the RBAC analysis to begin, which is mainly organisational information containing users' names, roles and permissions. This section discusses preparing event logs by providing the required information. Log preparation is an important step that has been studied and used in previous research [131]. This step can provide the required data as data attributes. For example, Weerdt et al. [129] showed three main purposes for having additional attributes:

1. To refine cases and events to provide a more focused viewpoint;
2. To enrich event logs with valuable context information;
3. To inspire process mining with new insights.

There are two possible methods to overcome the issue of a lack of necessary information, namely mining the information and uploading it.

- **Uploading the information:** If the required information is available in separate documents or known by the specialist, this information can be uploaded to enrich the event log. For instance, a manager can provide the roles and permissions in the system. Enriching event logs with extra information was used by Sellami et al. [88] and Hanachi et al. [89].
- **Mining the information:** Another way to acquire the necessary information is by mining it from the available event log using one of the process mining discovery techniques. Promising discovery techniques under the organisational perspective have already been published [13, 14, 15, 36, 31, 88, 89, 94, 96, 97, 100, 101, 102, 132, 133, 91, 92].

**Pros and cons:** The advantage of uploading the information is the simplicity of the process, although a drawback of using an existing organisational model as input is that such a model could be outdated or have changed, which will affect the analysis. The advantage of mining the information is ensuring that it reflects reality, while its drawback is that the quality of the mined results depends on the quality of the approach that is taken. Thus, mistakes or incompleteness of the model may lead to less meaningful results. Although there have been great efforts in this area, the link between them and process mining for RBAC analysis has therefore been neglected, thus far.

**Define the required constraints.** It was necessary for me to define the required RBAC constraints efficiently, enabling checking using process mining methods. In this experiment, I mostly used LTL definitions. The first three stages represented the design stage.

**Apply the checking.** This required reviewing the first three stages to ensure accurate results. This represented the execution stage.

**Interpret the results.** This required me to link the generated results with the organisation's background (i.e. the RBAC model), which is fundamental for understanding what has gone wrong. This represented the evaluation stage, which could then be used for the change stage.

By applying the life cycle to this case study, I have illustrated the steps for checking the RBAC in Figure 4.15.

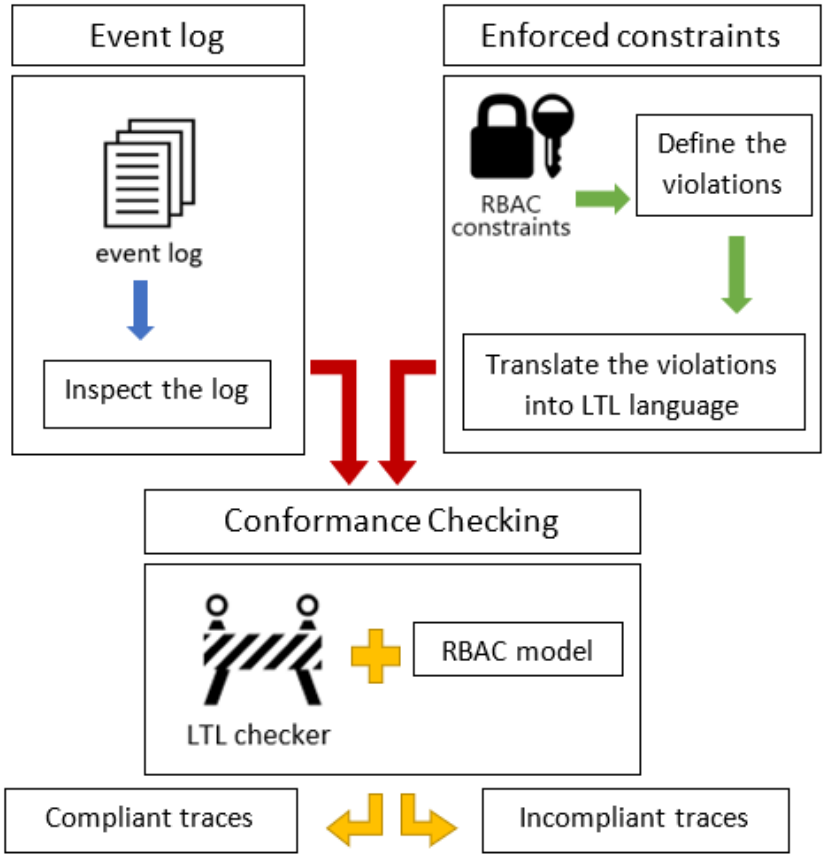



Figure 4.15: Presentation of the checking process by following the life cycle.

## 4.7 Related Work

I begin by discussing the related work on checking RBAC using process mining methods. I then discuss the related work on conformance checking.

### 4.7.1 Checking RBAC

The work most closely related to this chapter is process mining for RBAC checking [86, 93]. For instance, Baumgrass et al. used LTL to check the conformance to RBAC of a PAIS credit application system [86], proposing the transformation of a process-related RBAC model into LTL statements representing RBAC properties. They then used the LTL checker plug-in to check if business process executions complied with the corresponding process-related RBAC model [86].

Although their work and this study share the same concept, our aims and viewpoints differ. For instance, their work demonstrated the approach of converting an RBAC model into LTL statements without considering dynamic and temporal constraints. In contrast, this chapter has discussed in detail the LTL formula of each RBAC constraint and shown how these formulas can be modified to meet particular needs. Moreover, the evaluation in this chapter was applied on synthetic and real-life event logs.

Leitner [93] proposed a delta analysis approach to RBAC models, comparing an expected RBAC model with a current RBAC model derived from event logs. This approach aimed to identify differences between the two models to indicate security violations. The main difference between the work in this chapter and that of Leitner [93] is the type of checking. Leitner used delta analysis, where the checking was applied between two models. However, this chapter applied the analysis between defined rules and an event log. In [134], Aalst showed delta analysis to have two drawbacks: 1) there may not be enough events to actually discover the process model and 2) delta analysis does not provide quantitative measures for the 'fit' between the expected model and the log.

### 4.7.2 Conformance checking

The work on using conformance checking in process mining is directly related to the work presented in this chapter. In [135], Aalst et al. proposed an auditing framework based on process mining with different auditing approaches using process discovery and conformance checking. The authors discussed auditing using either models or event logs alone and described the challenges of auditing by process mining. In [87], Jans et al. applied rule-based checking in the context of transaction fraud, using process discovery to identify the actual process model and analyse the detected flaws. This work showed that a process mining approach could assist security analysts in checking individual security properties. Moreover, in [90], Accorsi et al. used rule-based checking for security auditing. Indeed, this case study was similar to that of Accorsi et al., using process mining for security auditing. The main difference was the focus on checking RBAC with insights into the checking process, the organisational background and the control flow aspect.

## 4.8 Conclusion

This chapter provided an exploratory case study using conformance checking for RBAC analysis of synthetic and real-life event logs. I defined several RBAC constraints and translated their violations into LTL formulas used in the LTL checker. The event log was tested against the defined RBAC requirements using the ProM framework. Specifically, I used two plug-ins for the checking, SCIFF and an LTL checker, where most violations were detected. This was followed by a discussion of the findings of the conformance checking of RBAC.

This chapter took a fundamental step of checking the compliance of an event log with defined rules. First, it explained in detail the checking process, providing a life cycle

consisting of five main stages as a guideline to design conformance checking on RBAC. Second, it introduced various RBAC requirements and formally defined the violations of each RBAC constraint. Third, it translated RBAC violations into formulas, explaining how they can be modified to meet particular aims. Fourth, it discussed the inputs and their impact on the case study. Last, it explained the process design in detail, including the design of the RBAC model and constraints and the generation of synthetic event logs.

The work in this chapter can be used as the first step for further investigations and improvements. The following chapter builds on this study to improve the checking by considering the context of the detected violations and measuring its impact on analysing the results.

# Chapter 5

# MULTI-PERSPECTIVE CONFORMANCE CHECKING OF RBAC

## 5.1 Introduction

Although several process mining checking approaches have been proposed to verify actual behaviour in recorded event logs, existing approaches have focused on either checking data against security policies (the data perspective) or checking the executed activities against those required to conduct processes (the functional perspective). However, processes are executed by people within an organisation to reach a specific goal. By relating these perspectives, an organisation can obtain better insight into the execution of their processes with regard to the expectations described in the models.

In Chapter 4, I conducted an exploratory study to check the employment of process mining for RBAC analysis, focusing mainly on the data perspective. This chapter builds on the work in Chapter 4 by linking the data and control flow perspectives for RBAC

analysis. Table 5.1 shows the goal, research question, and objectives addressed in this chapter.

| **Goal 2** | **Research question 3** | **Objectives** |
|---|---|---|
| To propose improvements to the way that process mining methods are used for analysing RBAC. | RQ3. If applied, will the proposed improvement enhance the effectiveness of RBAC analysis by using process mining? | 6. To propose improvements to the way that RBAC is analysed using process mining.<br>7. To implement the proposed improvements and test their effectiveness. |

Table 5.1: Chapter 5 has been designed to address goal 2, research question 3, and objectives 6 and 7.

### 5.1.1 Motivation

Taghiabadi et al. [136] categorised the domains of constraints into two main types:

1. **Constraints enforcing a restriction on control flow:** This type of constraint restricts activities when a certain data condition holds, for example, a constraint stating that "Activity B must not be executed for gold customers". This constraint restricts the execution of activity B when a certain condition holds. Here, the customer is gold; thus, there is a restriction on control flow.

2. **Constraints enforcing a restriction on data attributes:** This type assumes that the underlying control flow sequence is correct, for example, a constraint stating that "Activities A and B must not be executed by the same person P". Regardless of the execution order, A and B must not be executed by the same person. Thus, if A is executed first by P1, then the restriction would be on the person executing activity B.

Clearly, RBAC is in the second category. In Chapter 4, I only checked RBAC constraints from the data perspective, so the checking applied mainly to the event log. Although checking techniques to verify rules in logs detect violations efficiently, the overall conformance of the event log concerning a given process model is not determined [137].

Although RBAC does not enforce restrictions on control flow, Ferreira et al. [138] emphasised that managing access control is more than a technical solution. It involves the context of the process, organisational structure, and cultural background. In [117], Martin et al. argued that process mining is not only about technology, but also about people. It encompasses the entire process linking events and users to explain stories, rather than studying single events.

Considering the control flow perspective provides an understanding of the context in which an activity is executed and an organisational background of the people involved in the system. This chapter aims to fill this gap by linking these two perspectives: control flow and data.

### 5.1.2 Contributions

The main contribution of this chapter is *linking two perspectives (data and control flow) for conformance checking of RBAC to detect violations while considering their contexts, analysing each action and measuring their risks*. More specifically, I provide four concrete contributions.

- **Exploiting the control flow perspective:** In Chapter 4, I used rule-based checking to classify deviations for the intended goal: checking RBAC constraints. Since 'Activities A and B must not be executed by the same person P', I checked only the data on activities A and B. If the same person performed them, then the constraint was violated; otherwise, the constraint held. Thus, the checking was based

mainly on the data perspective. In this chapter, I discuss the data's context using conformance checking on the control flow perspective.

- **Combining multi-perspective insights:** The combination links the goal (data perspective) with the context (control flow perspective) to enhance the detection of RBAC violations with the contexts of the occurrence. Therefore, I introduced a multi-perspective alignment between the process model and event log combining data and control flow alignments. I also introduced the possible combinations of moves on the constraint consisting of single or multiple activities, such as SoD.

- **Computing the cost of the combination:** To properly link the data and control flow perspectives, I introduced a cost function assigned to each multi-perspective alignment and braced it with a colour-coded risk scale to document and visualise the cost.

- **Testing my approach:** To demonstrate the strength of this approach, I used the case study from Chapter 4, followed by a discussion of possible scenarios and behaviour analyses.

### 5.1.3 Outline

Section 5.2 explains the methodology of conducting this chapter. Section 5.3 presents the background of the multi-perspective approach, discussing the control flow alignments and data alignments and the purpose of each. Section 5.4 explains the multi-perspective approach and how it can be used for RBAC checking. Section 5.5 evaluates the approach. The findings are in Section 5.6, while the related work is presented in Section 5.7. Section 5.8 concludes this chapter.

## 5.2 Methodology

I adapted and employed a design science research methodology (DSRM) [5], as illustrated in Figure 5.1. I began the DSRM process by using the results from Chapters 3 and 4 as the main research problem (step 1) and proposed a multi-perspective conformance checking approach as a solution (steps 2 and 3), followed by modelling and evaluating the approach (steps 4 and 5). The approach is reported in this chapter (step 6).

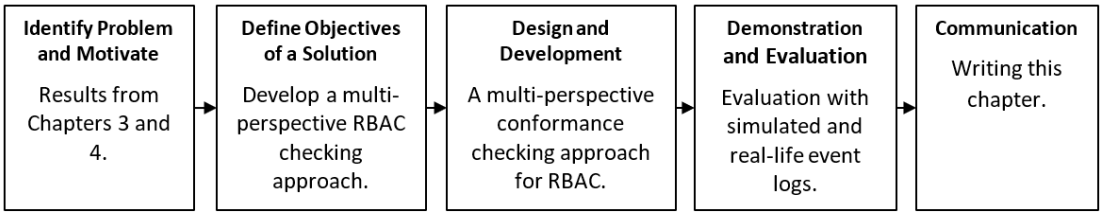



Figure 5.1: Applying a DSRM [5] to develop the checking approach.

### 5.2.1 Objectives

The objective of this chapter is to answer the first research question of this thesis, to improve the conformance checking of RBAC by combining the control flow and the data perspectives.

**RQ3: If applied, will the proposed improvement enhance the effectiveness of RBAC analysis by using process mining?**

To that end, I identified the following two questions based on the two corresponding objectives (6 and 7) in Table 5.1.

- Q1: What are the benefits of checking RBAC using a multi-perspective approach?
- Q2: Does the multi-perspective approach improve the checking of RBAC?

### 5.2.2 Methods

As this approach aimed to improve the conformance checking of RBAC, I proposed improvements in the case study in Chapter 4 by including the control flow perspective. Figure 5.2 presents the methods of conducting this approach.

To validate the data perspective, a rule-based checking is applied on the required RBAC constraints and the events logs. To validate the functional perspective, an alignment is applied on event logs and control flow. The results then are integrated to build a multi-alignments perspective. The alignments are assigned to cost which determine their risk level. To evaluate the approach, I use the synthetic and real-life data shown in Chapter 4.

This chapter aims to improve the checking by linking the detection of the violations with their contextual information. Finally, in Section 5.5, enhanced checking is applied.

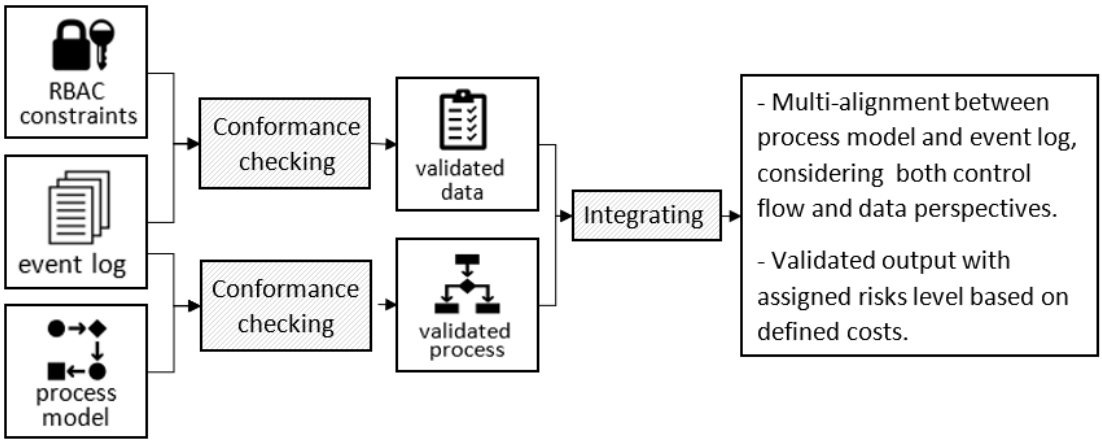



Figure 5.2: The methods of the multi-perspective approach in Chapter 5.

## 5.3 Alignments in Control Flow and Data

This section introduces the control flow and data alignments. For the demonstration, I used part of the accident and emergency department (A&E) process of a hospital for

handling patients, the first part of the process that identifies the patient (IP). The degrees of urgency were then defined during the triage (Tr). In this case, the patient could be discharged (Di_T) if no further investigation or treatment was needed. Figure 5.3 represents the part of the Petri net of this process.

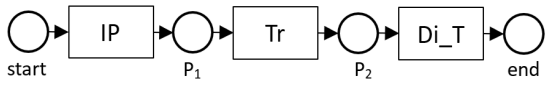



Figure 5.3: A Petri net of a part of the A&E treatment process.

### 5.3.1 Preliminaries

In this section, I introduce the main concepts and preliminaries used to model the process and data perspectives, which are the basis of the proposed approach. Here I consider event logs and process models. A process model describes the activities performed to reach a certain business goal (modelled behaviour), while an event log describes the execution of the activities (observed behaviour). Considering process model $M$ and event log $L$, I introduce two case types: process case and log case.

#### A process case

A process case shows the expected moves on the process.

**Definition 7.** ***(Process model).*** *A process model $M = (P, T, F)$ where $P$ is a finite set of places, $T$ is a finite set of transitions such that $P \cap T = \phi$. $F$ is a finite set of arcs such that $F \subseteq (P \times T) \cup (T \times P)$ is called the flow relation, which connects places and transitions.*

The Petri net shown Figure 5.3 can be formalised as follows: P = {start, $p_1$, $p_2$, end}, T = {IP, Tr, Di_T }, and F = {(start, IP), (IP, $p_1$), ($p_1$, Tr), (Tr, $p_2$), ($p_2$, Di_T), (Di_T, end)}.

**Definition 8.** ***(Process case).*** *A process case $A_M$ is a sequence of events in the process model $M = (P, T, F)$. Hence, $A_M \subseteq M$, and $y \in A_M$ where $y = (p, t, f)$, such that for each $y = (y_p, y_t, y_f) \in A_M$, $y_t$ is enabled in $y_p$ through $y_f$.*

Figure 5.4 shows a process case of the process in Figure 5.3, in which $A_M = y_1, y_2, y_3$. In this chapter, I present process cases in purple.

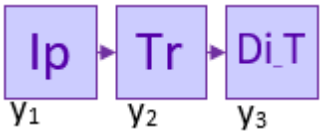



Figure 5.4: A process case.

**A log case**

A log case shows the actual moves on the event log.

**Definition 9.** ***(Log case).*** *A Log case $A_L$ of an event log $L$ is a set of ordered events $A_L \subseteq L$, where each $x \in A_L$, is related to a single process execution.*

Figure 5.5 shows an example of a log case executed in Figure 5.3, in which $A_L = x_1, x_2, x_3, x_4$. In this chapter, I present log cases in yellow.

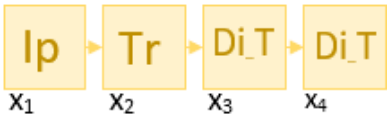



Figure 5.5: A log case.

### 5.3.2 Control flow alignments

Control flow alignments provide a robust approach to conformance checking. They relate the events in a process case to the events in a log case, thus discovering deviations causing nonconformity [139]. Skipped and inserted events are easily discovered by

directly aligning these behaviours. Here, I introduce the definition of alignment from Aalst et al. [139].

**Definition 10.** ***(Control-flow alignment).*** *Assume the set* $A\frac{1}{L} = A_L \cup >>$*, where* $x \in A_L$ *refers to "move x in log" and* $>>$ *refers to "no move in log". Similarly, they used* $A\frac{1}{M} = A_M \cup >>$*, where* $y \in A_M$ *refers to "move y in model" and* $>>$ *refers to "no move in model". One step in an alignment is represented by a pair* $(x, y) \in A\frac{1}{L} \times A\frac{1}{M}$ *, which results into four possible moves of the control flow alignments.*

- $(x, y)$ *is a move in log if* $x \in A_L$ *and* $y =>>$*. Hence,* $(x, >>)$*.*
- $(x, y)$ *is a move in model if* $x =>>$ *and* $y \in A_M$*. Hence,* $(>>, y)$*.*
- $(x, y)$ *is a move in both if* $x \in A_L$ *andy* $\in A_M$*. Hence,* $(x, y)$*.*
- $(x, y)$ *is an illegal move* $x =>>$ *and* $y =>>$*. Hence,* $(>>, >>)$*.*

$A_{LM} = \{(x, y) \in A\frac{1}{L} \times A\frac{1}{M} | x \in A_L \vee y \in A_M\}$ is the set of all legal moves. I call$(x, y)$ a synchronised move, $(>>, y)$ a model move, and $(x, >>)$ a log move. Figure 5.6 shows the control flow alignment between the process case in Figure 5.4 and the log case in Figure 5.5. The alignment contains three synchronised moves $(x_1, y_1)$, $(x_2, y_2)$, and $(x_3, y_3)$, and a log move $(x_4, >>)$.

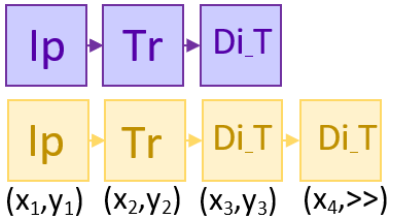



Figure 5.6: The control flow alignment between a log and process case

The aim of using the control flow perspective is *to provide the context in which an activity is executed.*

**Distant function and cost**

To qualify the quality of an alignment, the work in [139] introduced a *distance function* on legal moves: $\delta \in A_{LM}$. The distance function associates costs to moves in an alignment.

- if $x \in A_L$ and $y =>>$, then $\delta(x, y)$ is the cost of "move x in log",
- if $x =>>$ and $y \in A_M$, then $\delta(x, y)$ is the cost of "move y in model", and
- if $x \in A_L$ and $y \in A_M$, then $\delta(x, y)$ is the cost of "move x in log and move y in model" (typically $\delta(x, y) = 0$ if $x = y$).

Some alignments may be more desirable or likely than others. Often, the choice which of the alignments is better can only be made using domain knowledge. Mannhardt [137] defined a standard cost function that assigns 1 to each divination to form the four possible moves of the control flow alignments above. In the following sections, I define cost function to measure deviations related to RBAC constraints.

### 5.3.3 Data alignments

As process mining focuses on processes requiring the execution of activities recorded in event logs [24], this section introduces data alignments that align enforced constraints with related events in log cases.

Figure 5.7 shows stored data on event $x_2$. Each event has a number of attributes such as recourse and timestamps (see Chapter 2 for more details). Using this information allows the checking of RBAC constraints, as RBAC enforces restrictions on data, not the control flow [136]. Thus, I introduce the following notations for events related to the addressed constraint:

- **c:** is an event where a constraint is held.
- **¬c:** is an event where a constraint is violated.

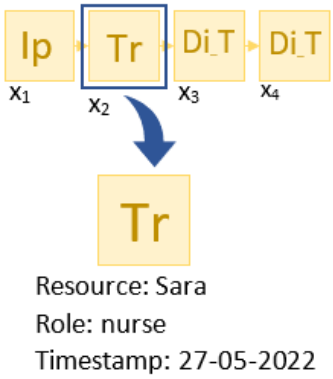



Figure 5.7: Data on an event from an event case.

For instance, to check the authorisation constraint on the log case in Figure 5.5, we can obtain the performers' names of the events from the event log. Figure 5.8 shows the results of checking the authorisation constraint on the log case by considering the RBAC artefact in Figure 4.4. The occurrence of the three first events is compliant with the constraint, while the fourth event violates it, as Alice is not authorised to perform the Di_T .

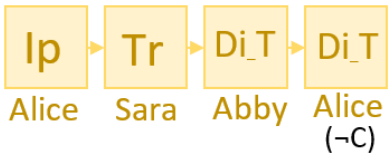



Figure 5.8: The data alignments on the log case.

In Chapter 4, I used a data perspective *to check the conformity of the RBAC policies on an event log*. The checking depended mainly on the event log and RBAC constraints. The result of checking a constraint is a case that is either compliant or non-compliant. Log cases can be studied in more detail with data alignments to discuss the events and their attributes.

## 5.4 Multi-Perspective Approach: Control Flow and Data

This section introduces an approach to checking RBAC by considering the data and control flow perspectives. I link these perspectives *to study the checking of the RBAC constraints considering the contextual aspect*. To my knowledge, no previous work has considered these perspectives for RBAC checking using process mining.

### 5.4.1 A multi-perspective alignment method

To improve the checking by considering the data and control flow perspectives, I combine the control flow and data alignments and introduce the multi-perspective alignment. First, I define the concept of combined move, which connects these perspectives.

**Definition 11.** ***Combined move.*** *Let $y \in A_M$ be an event in a process case $A_M$ and $x \in A_L$ be an event in a log case $A_L$. The control flow alignment is represented by the pair $(x, y)$. Let the condition of constraint $C$ be $c$ when a constraint is held, and $\neg c$ otherwise. A combined move is a tuple $((C), (x, y))$.*

To annotate a move of an event, I first provide the data notation, if applicable, followed by the control flow alignment. For example, checking the authorisation constraint by considering both the control flow and data alignments results in the multi-perspective alignment presented in Figure 5.9.

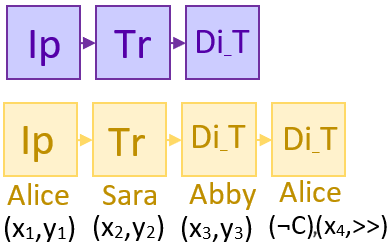



Figure 5.9: A multi-perspective alignment of checking the authorisation.

This combination shows the control flow alignment and data condition related to a specific constraint. There are three synchronised moves: $(x_1, y_1)$, $(x_2, y_2)$, and $(x_3, y_3)$, a violated constraint on a log move $\neg(c), (x_4, >>)$. Given the combined move $\neg(c), (x_4, >>$ ), I refer to the control flow move $(x, y)$ as the context of the execution and the data move $(c)$ as the condition of the constraint.

The condition and context are used to assess an event's conformity with the specification. The (non)conformity at the process (control flow) level provides contextual information as it determines whether moves in a case occurred per the expected control flow. The (non)conformity of data is used to determine the validity of the addressed constraint.

Based on the definition of the combined move above, I distinguish eight possible moves in Table 5.2. I refer to the move indicating the row and column, so (2,3) is the move in row 2 and column 3, which is $\neg(c), (x, >>)$.

| **Data/Control flow perspective** | **Synchronous move** $(x, y)$ | **Model move** $(>>, y)$ | **Log move** $(x, >>)$ | **No move** $(>>, >>)$ |
|---|---|---|---|---|
| Compliant data $(c)$ | $((c), (x, y))$ | $((c), (>>, y))$ | $((c), (x, >>))$ | $((c), (>>, >>))$ |
| Incompliant data $(\neg c)$ | $(\neg(c), (x, y))$ | $(\neg(c), (>>, y))$ | $(\neg(c), (x, >>))$ | $(\neg(c), (>>, >>))$ |

Table 5.2: Possible moves

I refer to the control flow move $(M)$ when the move is synchronised and $(\neg M)$ when the move is in a log or model only. I group these moves into five main categories as follows:

- **A legitimate move** $(C, M)$**:** This type indicates fully compliant behaviour. An event is executed with compliant data under a valid context. This group contains only one move (1,1).

- **A compliant missing move** $(C, \neg M)$**:** This type indicates a compliant data move under an invalid context. This group contains moves (1,2) and (1,3).

- **An incompliant move** $(\neg C, M)$**:** This type indicates an incompliant data move under a valid context. This group comprises only moves of type (2,1).

- **An incompliant missing move** $(\neg C, \neg M)$**:** This type indicates an incompliant data move under an invalid context. This group comprises moves of types (2,2) and (2,3).

- **An illegal move:** This type does not correspond to any legal multi-perspective moves. This group consists of moves of types (1,4) and (2,4). This chapter does not include moves of this type as they cannot be detected in event logs or process models.

### 5.4.2 The multi-perspective risk scale

Within the security field, many specialised approaches exist to document and visualise risks. The most common way of documenting risks is the risk matrix approach (RMA) developed by the U.S. Air Force [140], identifying which risks are most important to a programme. The use of RMA has spread, and is used in weapons manufacturing, finance, transport, and project management [141]. Many other approaches using colour-coded scales for cybersecurity threats have been introduced, such as the use of bow tie diagrams to assess security risks by Bernsmed et al. [142]. In 2016, the White House released a colour-coded cyberattack scale [141] that assigned specific colours to the dangers of cyber attack.

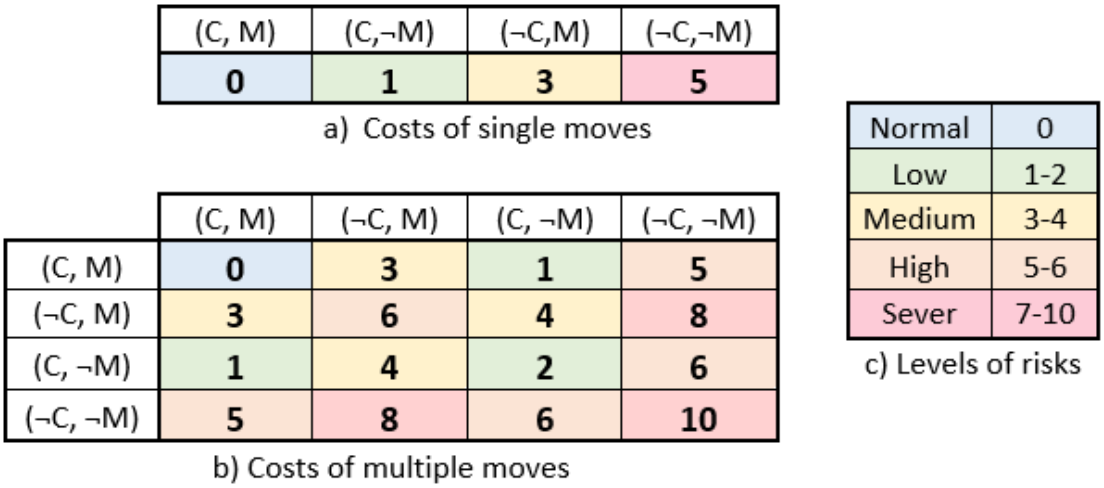
Here, I introduce a multi-perspective risk scale to assign costs to multi-perspective alignments, measuring and visualising the risks of the cases detected using the multi-perspective process mining approach for RBAC analysis. The goal of the scale is to

consider the multi-perspective alignment and measure the costs, presenting the results with a colour-coded approach indicating risk levels. Cost function is a technique that measures the alignments between a process model and an event log, taking into account the cost of skipping and inserting individual activities [139, 143, 137]. The scale uses the four aforementioned groups of moves as the main input: a legitimate move, a compliant missing move, an incompliant move, and an incompliant missing move. Thus, I assign a cost to each of these moves as follows: a legitimate move = 0, a compliant missing move = 1, an incompliant move = 3, and an incompliant missing move = 5. The scale consists of five risk levels:

- Normal (blue): No violation is detected. The score of this level is 0.
- Low (green): No data violation is detected. However, a deviation in the control flow perspective is present, which could mean a possible violation. The score of this level is 1–2.
- Medium (yellow): The detection of a violation in the data perspective could be combined with a deviation in the control flow perspective on a different event. The score of this level is 3–4.
- High (orange): The presence of an incompliant missing move or two incompliant moves indicates violations from the data and control flow perspectives. The score of this level is 5–6.
- Severe (red): A high level of data violation occurs when an incompliant missing move is combined with data violation on a different move. The score of this level is 7–10.

As the checking aims to detect RBAC violations, nonconformity at the data level is at higher risk (higher cost) than nonconformity at the process level. Figure 5.10 shows

the details of the multi-perspective risk scale.



| (C, M) | (C,¬M) | (¬C,M) | (¬C,¬M) |
|---|---|---|---|
| 0 | 1 | 3 | 5 |

a) Costs of single moves

| | (C, M) | (¬C, M) | (C, ¬M) | (¬C, ¬M) |
|---|---|---|---|---|
| (C, M) | 0 | 3 | 1 | 5 |
| (¬C, M) | 3 | 6 | 4 | 8 |
| (C, ¬M) | 1 | 4 | 2 | 6 |
| (¬C, ¬M) | 5 | 8 | 6 | 10 |

b) Costs of multiple moves

| | |
|---|---|
| Normal | 0 |
| Low | 1-2 |
| Medium | 3-4 |
| High | 5-6 |
| Sever | 7-10 |

c) Levels of risks

Figure 5.10: The multi-perspective risk scale.

Since most of the RBAC constraints addressed in this thesis enforce restrictions on two activities (i.e. SoD and BoD), I defined a method to measure the cost of single and multiple moves. Figure 5.10.a shows the costs of single moves. All possible combinations of multiple moves and their costs appear in Figure 5.10.b, while the level of risk indicator is in Figure 5.10.c.

By relating actual and modelled behaviour, conformance checking unleashes the full power of process mining. This multi-perspective conformance checking of RBAC aims to detect violations while considering their contexts, analysing each action, and measuring their risks.

## 5.5 Evaluation

This chapter builds on the previous chapter using the same case studies and RBAC constraints as Chapter 4. The aim is to show the improvement of the results and the benefits of the multi-perspective approach.

In this section, I apply the multi-perspective checking approach of RBAC by combining the control flow and data perspectives. The checking of the data perspective is applied using the rule checking approach as described in Chapter 4, while the control flow checking is applied by aligning the process model and event logs. The cost is computed by assigning values to activities during the alignments, which are then used to apply the risk scale. I highlight the events that caused the violations and the context of their occurrences.

To find control flow deviations between the model and the event log, I used the work presented in Lu et al. [144], which is implemented as a plug-in named 'Partial-Order Aware Replayer' in the ProM framework. This plug-in takes as inputs a Petri net and an event log. For each constraint, I applied the checking and assigned costs for the targeted activities. Constraints are enforced on either individual activities or pairs of activities. Thus, the checking takes one of the following formats:

- Checking single move: As checking authorisation checks individual activities, the costs of deviations on the model and log are set to 1 for all activities, while deviations on data are set to 3 for each activity. This successfully discovered deviations on both perspectives are reported with assigned costs.

- Checking multiple moves: Checking SoD and BoD checks two activities for each constraint. When aligning, costs of 1 are assigned only to the target activities, while, deviations on data are set to 3 for each activity. Thus, for each constraint the alignments for the target activities are discovered. For example, to check SSoD from the A&E process, I compute the alignments of RT and LT between the process model and the event log, as a deviation cost of 1 on each. Deviations on data by using LTL statements cost 3 for each activity. The analysis successfully checked each constraints with adjustments of the target activities.

### 5.5.1 Synthetic event log

This case study is based on a real-life scenario of a hospital's A&E process for handling patients, as presented in Chapter 4. Based on a literature review and the interviews with doctors, I identified a number of scenarios and simulated them in the log to violate RBAC constraints. I generated the event log using the process simulation tool CPN [39].

The evaluation in this section discusses the checking of RBAC constraints on the context of those behaviours. As this analysis is applied to a synthetic event log, I evaluated the approach by the success of discovering violations and measuring their risks.

**Scenario 1.** When a patient was admitted to the ER, he received a visit from a doctor (Vi). Based on the visit, the doctor either consulted a colleague (CC) or requested a lab test (LR). A test was requested by an employee for a *personal reason*, such as for a relative. A request can be accepted or declined.

- **Authorisation:** Each user should perform only permissible tasks.
- **Simulated violation:** A case where an LR occurred out of context by David, a phlebotomist. As shown in the RBAC artefact in Figure 4.4, a phlebotomist cannot perform an LR. The request is declined.
- **Analysis:** The multi-perspective alignment between the log case and the process case appears in Figure 5.11. The log case shows a deviation where an LR occurred out of context. The log does not show any further activities in this direction (i.e. the test was not performed), indicating that the request was denied.

  The multi-perspective analysis shows deviations on data and control flow. Thus, this case scored high on the scale, as shown in Figure 5.12.

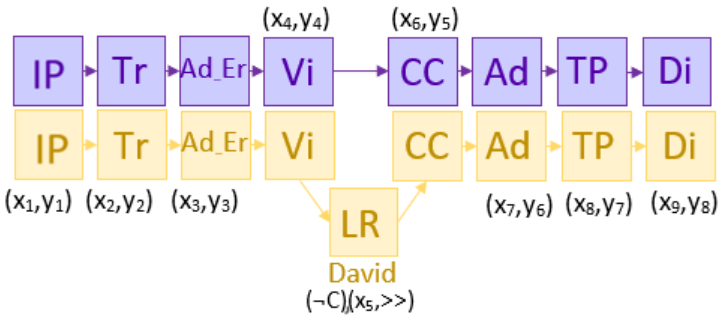



Figure 5.11: The multi-perspective alignment of checking Authorisation (Scenario 1).

| Scenario 1 | (¬c),(x5,>>) | High |
|---|---|---|

Figure 5.12: The cost of Authorisation on the scale (Scenario 1).

**Scenario 2.** *Granting permission* is common in cases of a lack of a staff. Although it could be applied based on the qualifications of a subject [97], it must be controlled as it could result in a data breach.

- **Authorisation:** Each user should perform only permissible tasks.
- **Simulated violation:** This case represented the first steps of entering the A&E department, where a patient was identified (IP), triage was applied (Tr), and the patient was discharged (Di_T). Here, Di_T was performed twice while only once was expected. Both occurrences of the Di_T were performed by Henry. This case violated authorisation as Henry is a receptionist, and receptionists are not authorised to perform a Di_T.
- **Analysis:** Figure 5.13 shows the multi-perspective alignment between the log and process case. The multi-occurrence of Di_T could indicate that the first one was not successful. However, this had to be solved by Henry being granted permission to discharge a patient.

  As the alignments show two deviations on data and one on control flow, the case scored severely on the scale, as shown in Figure 5.14.

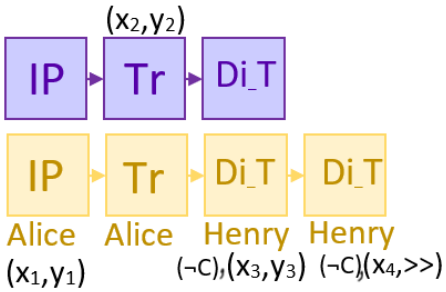



Figure 5.13: The multi-perspective alignment of checking Authorisation (Scenario 2).

| Scenario 2 | $(\neg c),(x_3,y_3)$ | $(\neg c),(x_4,>>)$ | Severe |
|---|---|---|---|

Figure 5.14: The cost of Authorisation on the scale (Scenario 2).

**Scenario 3.** A *delegation* is a common behaviour in teamwork. It occurs when a task is delegated to another employee, such as when the first one is required in another situation or the second one is asked to take over the case for *personal* reasons, such as knowing the patient.

- **DBoD:** Discharge after triage (Di_T) must be conducted by the same nurse that does the triage (Tr). This constraint considers the very beginning of the process after a patient is identified (IP). The patient is evaluated in triage (Tr) to either be discharged (Di_T) or admitted to the ER (Ad_Er) for further investigation.

- **Simulated violation:** Sara, who triaged a patient, tried to discharge him in this case. The discharge was unsuccessful as the patient needed to be identified first. Sara *delegated* the task of identifying and discharging the patient to a nurse working at the desk. Thus, the patient was first identified and then discharged. Here, Tr and Di_T were performed by different users, which violated DBoD.

- **Analysis:** Figure 5.15 presents the multi-perspective alignment of the case where the second discharge $(\neg c)(x_4, >>)$ violated both perspectives. Figure 5.16 shows that the case scored high on the scale.

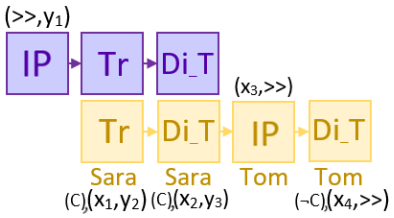



Figure 5.15: The multi-perspective alignment of checking DBoD (Scenario 3).

| Scenario 3 | $(c)_1(x_1,y_2)$ | $(c)_3(x_2,y_3)$ | $(\neg c)_5(x_4,>>)$ | High |
|---|---|---|---|---|

Figure 5.16: The cost of DBoD on the scale (Scenario 3).

**Scenario 4.** In teamwork, rectifying a problem when it is noticed, and an action taking place to solve the the issue.

- **TBoD:** Triage (Tr) must be conducted by the same nurse that grants admission to ER (Ad_Er) only if the time between these activities is more than 1 hour. This constraint considers when a patient is admitted to the ER after triage.

- **Simulated violation:** In this case, an Ad_Er was performed more than 1 hour after the performance of the Tr, and the performers were two different nurses (Sara and Tom).

- **Analysis:** The patient was identified more than 1 hour after the triage and was admitted immediately to ER by the same nurse, Tom. This action could be explained as follows. Tom *noticed the problem* that a patient was waiting after triage and was not identified, and thus was admitted for further investigation. Figure 5.17 presents the multi-perspective alignment.

  Figure 5.18 shows that this case scored medium on the scale.

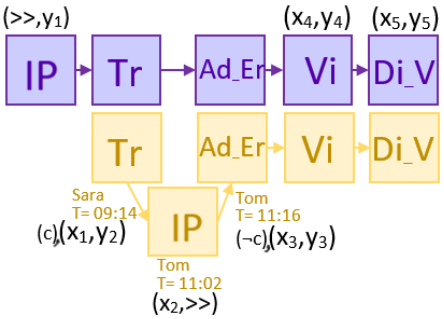



Figure 5.17: The multi-perspective alignment of checking TBoD (Scenario 4).

| Scenario 4 | (c),(x1,y2) | (¬c),(x3,y3) | Medium |
|---|---|---|---|

Figure 5.18: The cost of TBoD on the scale (Scenario 4).

### 5.5.2 Real-life event log

For the experiments with real-life logs, I used a dataset taken from incident and problem management processes supported by Volvo IT's VINST system (available in [27]). The incident and problem management process forms an essential part in every organisation. Since businesses rely heavily on IT, each issue should be dealt with as quickly as possible. The log files consisted of 65,533 incident records and 6,660 problem records. I evaluate the approach with respect to the identified RBAC constraints in the following.

**Checking DBoD.** The DBoD constraint states that in cases completed during calls, solving and closing the incident must be done by the same person.

Although data checking does not discover any violations, control flow alignment discovered deviations on the target activities. "Completed in call" means a case is solved and closed during a phone call by the same employee. If the case cannot be solved, then it must be transferred to a specialised team to be resolved and closed. However, the multi-perspectives checking discovered cases where "completed in call" occurred twice, one with legitimate users and other without. This behaviour requires further investigation. All these cases scored low on the risk scale.

**Checking SSoD.** The SSoD constraint states that when a case is transferred to a higher line from a role that is not V3-2, it must be assigned to a different role.

In the case, a transition is done to the same role, but this is followed by another transition to a different role. All of the three activities are done by the same user, the last one with a different role. This could indicate that this user is assigned to multiple roles and the first transition was to the wrong role, while the second one was successful. This can be a case of rectifying scenario with a medium risk level.

**Checking temporal constraint.** The temporal constraint TSoD states that a delayed case with more than 90 days of Wait-User time must be closed by a different person.

There is a non-compliance case with a multiple occurrences of the Wait-User event, which deviates from the expected behaviour. As the performers of these actions are different, this could be a case of delegation. However, a further investigation is required.

## 5.6 Findings

Recall the research questions:

- Q1: What are the benefits of checking RBAC using a multi-perspective approach?
- Q2: Does the multi-perspective approach improve the checking of RBAC?

Sections 5.6.1 and 5.6.2 answer Q1, while Sections 5.5 and 5.6.3 answer Q2.

### 5.6.1 Behaviour analysis

There have been many proposed technical solutions to manage access control privileges in organisations. However, Ferreira et al. [138] emphasised that managing access control is more than a technical solution, requiring consideration of the process, organisational structure, and cultural background. Thus, by considering the possible scenarios

and types of threats in healthcare systems described by Appari et al. [145], I identified six reasons behind such behaviours: personal reasons, delegation, granting permission, rectifying, potential violations, and process flaws. This section answers Q1 by providing this behaviour analysis as a result of the multi-perspective checking.

**Personal reasons:** When users abuse their privileges for curiosity or illegal purposes [145], actions under this class could be for financial gain. An example is medical fraud, involving access to protected data to conduct studies or even sell data that belongs to a celebrity, or for personal gain if the performer has a personal relationship with the patient and would like to oversee this case, as in the first scenario. Although this action would not be particularly harmful, the behaviour should be identified.

**Delegation:** Campbell et al. [146] emphasised the importance of having sufficient teamwork, delegation, and communication between nurses, as three-quarters of deaths in hospitals are related to breakdowns in teamwork and communication. Scenario 3 shows a possible case where a delegation could sabotage the required constraints.

**Granting permission:** Actions under this type could occur for different reasons. For example, it is common in hospitals to grant permission with the "break the glass" (BTG) policy [147]. BTG refers to a case when a person who does not have permission for specific information gains permission when necessary. A drawback of this policy is that users can misuse the granted privilege [145]. Another case of granting permission can occur due to a lack of staff, and an employee has the required qualifications to perform the task [36]. Scenario 2 is an example of this class.

**Rectifying:** When users unintentionally breach security [145] as the healthcare system and defined roles change, ER nurses play a significant role in trauma triage. Awwad et al. [148] showed that nurses can help reduce medical errors for patients entering the hospital by providing triage correctly. A tool to rate team performance and deliver a constructive debrief for clinical settings was developed by [149]. Their main

categories were leadership, teamwork, and task management. Although there can be efforts to rectify these errors, some corrections could lead to deviant behaviours, as in Scenario 4.

**Potential violation:** Managing the hospital, and A&E in particular, is an important and sensitive task. Accordingly, securing access control is necessary. In hospital environments, it can be complex to ensure that only authorised, legitimate users can access specific resources to which they are entitled [150]. Thus, cases where violations are not clearly defined should be identified and diagnosed.

**Process flaws:** Appari et al. [145] explained that faulty system design features could significantly threaten security enforcement. The cardinality constraint differs from other constraints that are not related to a specific activity, but to an entire process. Thus, a *robust model* is required. An interesting direction to improve the application of such constraints is considering the standard four quality criteria: fitness, precision, generalisation, and simplicity [3, 4].

### 5.6.2 Risk scale

This section answers **Q1** by showing another benefit of the multi-perspective approach that relates to the levels of the detected cases on the multi-perspective scale. Notably, the score only considers events related to the addressed constraint. Other events are not considered in the counting as it is not in the scope of this case study.

**Low level:** Although cases on this level do not include deviations from the data perspective, they include at least one deviation from the control flow perspective concerning an event related to the addressed constraint, for example, on a case, where DSoD was applied to the Ev and CS. In this case, a lab result was evaluated (Ev) but was missing the second signature (CS). DSoD was not violated, yet the deviation of the Ev could indicate a possible threat.

**Medium level:** The cases on this level contain an incompliant move and a possible compliant missing move. It takes only one occurrence of an incompliant move to raise the risk level to medium, because the RBAC enforces restrictions on data, not control flow [136]. The incompliant move shows that a constraint is explicitly violated. If the case also contains a compliant missing move, the risk score increases but remains medium.

**High level:** One occurrence of an incompliant missing move is enough to raise the risk to a high level because the event is not expected from the model and violates a specific constraint. This occurrence could indicate an intentional act to breach private information.

**Severe level:** This level is the highest on the scale, indicating violations from the data perspective on more than one event. When checking RBAC constraints, data violations are considered an extreme threat and must be checked.

Hence, cases with deviations in the control flow only are low level, while data deviations are at least medium level because the checking is mainly on RBAC, so deviations in data increase risks. Cases with deviations on both perspectives are given higher priority as they indicate a problem that causes the violation.

### 5.6.3 Holistic analysis

This chapter uses the conformance checking of data to reach the goal of detecting RBAC violations. Furthermore, it uses conformance checking on the control flow to enhance the detection by understanding the context of such deviations. Combining these perspectives helped reach the goal as efficiently and precisely as possible.

This section answers **Q2** by discussing the quality of the results. The results show that some cases could not be discovered by considering a single perspective. Different reasons could cause a violation, so different explanations and solutions are provided.

**Considering the entire scenario:** By combining both perspectives, the checking can consider the entire scenario, including activities, players, times, and context. The activities and data are not checked in isolation of the circumstances related to their occurrence. Thus, data are not checked separately from the expected model, nor is a model checked without considering actual behaviours. Successful access control management involves all circumstances, such as processes, restrictions, and organisation [138]. The results of these considerations portray the strength and particularity of process mining.

**Providing more robust detection:** All the cases discovered by data checking are also discovered by the multi-perspectives approach. Besides the ability to check RBAC constraints, other flaws are revealed in areas such as teamwork and control flow. Understanding the behaviour resulting in a violation shows where and how it went wrong. In addition, teamwork behaviour must be maintained, such as granting permissions and delegating tasks, since control flow allowing behaviour beyond the expectation (generalisation) will likely allow more violations at the data level.

## 5.7 Related Work

Most conformance checking methods focus on a single perspective, usually the control flow perspective [38]. However, some constraints, such as RBAC, enforce restrictions on data attributes rather than activities. To check these constraints, conformance checking methods from the data perspective are required [136]. Linking these perspectives is still underdeveloped in the literature, so here, I discuss the work considering the conformance of processes from multiple perspectives.

**Rule-based checking.** Rule-based checking assesses certain constraints on a process model or an event log. In [151], Caron et al. proposed a comprehensive rule-based checking approach that considered the control flow with organisational and

data perspectives. Their approach verified conformance efficiently. The nature of the rule-based approach allowed the checking of certain constraints rather than the whole behaviour. Thus, in contrast to the work in this chapter, the conformance of the event log with a given process model was not determined.

**Data-aware approach.** This approach aligns control flow and checks for deviations at the data level [152, 153, 154, 155]. In [152], Leoni et al. proposed an approach to align the control flow first and then to check for deviations at the data level. This work was extended by Mannhardt et al. [153], who proposed a balanced consideration of data and control flow perspectives. Alizadeh et al. [103] introduced an auditing approach to reconciling the data and process perspectives for identifying deviations by identifying the data's purpose and context. However, none of these approaches checked compliance with enforced rules.

In contrast to these techniques, I propose an approach to check the enforced rules by considering multi-perspectives and aligning the functional side. Thus, interpreting the results is not hindered, but can be enhanced by the information brought from other perspectives.

To my knowledge, the work in this chapter is the first designed to address RBAC constraints while considering contextual information.

## 5.8 Conclusion

Many process mining methods have been proposed recently to analyse the observed behaviour recorded in event logs, typically focusing on the control flow or data perspective. However, focusing on a single perspective may be insufficient to detect security violations, and some violations may remain undetected or misdiagnosed. Therefore, I proposed a process mining approach to analyse RBAC constraints on event logs from the

data and control flow perspectives in this chapter. I defined a multi-perspective alignment and a group of combined moves as the basis for this analysis. Then, I introduced a colour-coded multi-perspective risk scale to assign costs to alignments to document and visualise risks, followed by conducting experiments to show that the proposed approach could detect RBAC violations accurately while providing valuable insights into deviations. Finally, I discussed the results by considering the process and organisational structure in possible scenarios and threats.

This chapter proposed a novel approach integrating two types of conformance checking: rule-based checking and event log and process model alignment. The approach employed rule-based checking to verify the compliance of RBAC constraints, using the alignment to identify the context of the detected violation. This integration strengthened the rule-based approach of detecting nonconformity while overcoming its limitations by considering the entire behaviour. An interesting research direction for future work would be to develop techniques to automatically integrate the rule-based checking and event logs and process models' alignment approaches.

This approach can check access control beyond the technical side, as it allows the combination of the process perspective, organisational structure, and cultural background. An interesting direction for future work would be to develop techniques to provide analysis and explanations of this knowledge.

# Chapter 6

# CONCLUSIONS AND FUTURE WORK

This chapter summarises my main contributions and links them to the main goals of conducting this thesis in Section 6.1. I discuss the limitations of this thesis and present some interesting directions for future work in Section 6.2.

## 6.1 Conclusions

In Chapter 1, I introduced the two main goals of this thesis as follows:

- **Goal 1:** To investigate the analysis of RBAC properties in organisations by using process mining methods, and
- **Goal 2:** To propose improvements in how process mining methods are used to analyse RBAC.

Goal 1 (objectives 1–3) was achieved in Chapter 3, which provided a systematic literature review of the usage of process mining for access control analysis to allow a

thorough understanding of the most recent work in the field. The systematic literature review analysed 27 publications that discussed 40 different approaches. The review shows that although process mining is a hot topic nowadays, and much effort has been devoted to analysing security issues, investigating RBAC is still far from expected. There is a significant lack of considered ways to use process mining in the context of RBAC. Thus, possible limitations and potential opportunities remain unexplored.

Goal 1 (objectives 4–5) was achieved in Chapter 4, which provided an exploratory case study on employing process mining methods, particularly rule-based checking to check RBAC constraints. Chapter 4 aimed to recognise the characteristics of the checking approaches, identify the limitations, and uncover the challenges of the process. The exploratory case study discussed a robust conformance checking method to detect RBAC violations while providing insights into possible ways to adapt this method for better results. Furthermore, this case study uncovered challenges from the process's first steps to the final step of explaining the results. Thus, Chapter 4 introduced a life-cycle for the conformance checking of RBAC.

Goal 2 (objectives 6–7) was achieved in Chapter 5, which proposed a multi-perspective checking approach of RBAC constraints to improve the process. The proposed approach combined two types of checking, rule-based and alignment, to consider the purpose of the checking and the corresponding contextual information. In contrast to the exploratory case study in Chapter 4, the multi-perspective approach provided another dimension to the analysis in considering the circumstances and resources contributing to the bigger picture and more accurate analysis. For validation, Chapter 5 applied the approach to an accident and emergency department (A&E) process and real-life log of incident and problem management processes, supported by Volvo IT's VINST system [27] process, where violations were detected and their risks were measured.

Compared with the state of the art, this comprehensive work is the first focused

on using process mining for RBAC checking. Overall, this thesis determined that *the input type and quality reflected the analysis's type and quality*. Organisations that use detailed logs allow more detailed analysis, as they provide more information that reflects reality. Likewise, organisations that use up-to-date process models allow better results, as the recorded information is compared to models that reflect the expectations. The same principle applies to RBAC constraints, since organisations that enforce specific restrictions have greater advantages in detecting potential violations.

Furthermore, this thesis showed that *the analysis was multi-dimensional*, including knowledge from process science, such as process modelling; knowledge from data science, such as security and database; and knowledge from social science, such as teamwork behaviour. The more perspectives were considered, the more accurate the results generated were one perspective in isolation might have provided inaccurate results or even failed to achieve the acquired target.

To conclude, I used the interactive model of research design proposed by Maxwell [26] as the theoretical framework for the design of this thesis. I attempted multiple iterations to ensure that the knowledge I gained from one part was used and reflected in the other parts.

## 6.2 Limitations and Future Work

This section discusses the limitations of my work, and suggests promising future research directions on the topics covered by this thesis.

### 6.2.1 Work in progress

As a next step, I designed a conceptual framework for employing process mining for RBAC analysis. The framework considered all the surrounding elements impacting the

process to draw a picture as close to reality as possible. Thus, the framework consisted of four main concepts: 1) functional perspective, 2) data perspective, 3) RBAC perspective, and 4) organisational background. The framework can be used for any desired task, such as mining models or enriching resources.

To support the framework, I developed a plug-in that enriches event logs with organisational information, which works as an element that facilitates the analysis.

I applied the framework to a model designed to check RBAC constraints, which can be used as a guideline when designing a conformance checking approach. A user study should be conducted to evaluate the effectiveness of this framework.

### 6.2.2 Future work

This study focuses on analysing RBAC constraints, since analysing RBAC models was beyond the scope of this thesis. I assumed the RBAC model was correctly applied in this work. Investigating constraints only is a complex task that involves addressing different dimensions, from formalising constraints and violations to simulating behaviours, checking constraints, and interpreting the results. Thus, it would be interesting to analyse the current RBAC model. The current model can be examined using one of the process mining techniques discussed in this thesis. Moreover, the conformity between a designed RBAC model and the current one could be checked and measured. Then, the designed model can be updated if necessary, or the current model can be enhanced or improved.

Some other directions for interesting future work including the following.

**Absolute process mining:** There are three main types of process mining: discovery, checking, and enriching. It would be interesting to explore the power of utilising these types collectively to analyse RBAC. For instance, process discovery can be used to derive the RBAC model from an organisation to provide a view of

the actual RBAC model of that organisation, followed by conformance checking relating to the discovered and modelled behaviours. In addition to detecting violations, the results can contribute to discovering deviations or flaws in one of the two behaviours, which can be corrected and used to enhance the current situations and models.

**Multi-perspective approaches:** The number of publications in process mining is increasing rapidly, and there are significant efforts from different perspectives, such as organisational mining, control flow mining, and conformance checking. Nonetheless, they have rarely interacted, including their lack of consideration of an RBAC perspective. Thus, the work in this thesis opens the door for cooperation and unifying such efforts, clarifying that such cooperation can improve the checking of RBAC by using process mining. Furthermore, applying a multi-perspective approach in general while exploring their potential and limitations can improve the state of the art and open new dimensions for researchers.

**Human behaviour and role:** Past discussions have mainly focused on technical issues, whereas this thesis showed the impact of human behaviour on the analysis. To the best of my knowledge, the employment of process mining for RBAC and the research on human behaviour have not yet interacted. Hence, conducting research that considers social science, the psychological side, and the technical side is a promising direction for future work, since people's cultures and backgrounds direct their behaviours as team players and influence their actions towards following explicit and implied rules. Finally, it is important to involve experts in the analysis and educate people on the skills behind these different perspectives.

# Bibliography


[1] Numminen L. Top process mining software 2023, 2023. URL `https://www.workfellow.ai/blog/top-process-mining-software`. Accessed on 01-03-2023.

[2] ProM tools, 2022. URL `https://promtools.org/`. Accessed on 01-03-2023.

[3] Aalst W. *Process Mining: Data Science in Action*. Springer Publishing Company, Incorporated, 2nd edition, 2016. ISBN 3662498502, 9783662498507.

[4] Carmona J, van Dongen B, Solti A, and Weidlich M. *Conformance Checking*. Springer Publishing Company, Incorporated, 1st edition, 2018. ISBN 978-3-319-99414-7.

[5] Peffers K, Tuunanen T, Rothenberger MA, and Chatterjee S. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[6] Ferraiolo DF, Kuhn DR, and Chandramouli R. *Role-Based Access Control*. Artech House, Inc., USA, 2nd edition, 2007. ISBN 1596931132.

[7] Ferraiolo DF, Sandhu R, Gavrila S, Kuhn DR, and Chandramouli R. Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):224–274, 2001.

[8] Tunggal AT. What is role-based access control (RBAC)? examples, benefits, and more, 2022. URL `https://www.upguard.com/blog/rbac`. Accessed on 01-03-2023.

[9] Ferraiolo DF, Gilbert DM, and Lynch N. An examination of federal and commercial access control policy needs. In *National Computer Security Conference, 1993 (16th) Proceedings: Information Systems Security: User Choices*, page 107. DIANE Publishing, 1995.

[10] Sandhu R and Samarati P. Access control: principle and practice. *IEEE communications magazine*, 32(9):40–48, 1994.

[11] Vincent C Hu, Rick Kuhn, and Dylan Yaga. Verification and test methods for access control policies/models. *NIST Special Publication*, 800:192, 2017.

[12] National Institute of Standards and Technology. Access control policy tool (ACPT), 2021. URL `https://www.nist.gov/programs-projects/accesscontrol-policy-tool-acpt`. Accessed on 16-06-2023.

[13] Aalst W, Reijers HA, and Song M. Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, 14(6):549–593, 2005.

[14] Ly LT, Rinderle S, Dadam P, and Reichert M. Mining staff assignment rules from event-based data. In Bussler J and Haller A, editors, *Business Process Management Workshops*, pages 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32596-3.

[15] Jin T, Wang J, and Wen L. Organizational modeling from event logs. In *Sixth International Conference on Grid and Cooperative Computing (GCC 2007)*, pages 670–675. IEEE, 2007.

[16] Aalst W. Process mining: The missing link. In *Process Mining*, pages 25–52. Springer, 2016.

[17] Aalst W, De Beer HT, and van Dongen B. Process mining and verification of properties: An approach based on temporal logic. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 130–147. Springer, 2005.

[18] Aalst W, Adriansyah A, Medeiros A, Arcieri F, Baier T, Blickle T, Bose JC, Brand P, Brandtjen R, Buijs J, Burattin A, Carmona J, Castellanos M, Claes J, Cook J, Costantini N, Curbera F, Damiani E, de Leoni M, and Wynn M. Process mining manifesto. volume 99, pages 169–194, 08 2011. ISBN 978-3-642-28107-5. doi: 10.1007/978-3-642-28108-2_19.

[19] Remyga V. Global process mining survey 2021, 2021. URL `https://www2.deloitte.com/kz/en/pages/risk/articles/global-process-mining-survey-2021.html`. Accessed on 01-03-2023.

[20] Workfellow tool, 2022. URL `https://www.workfellow.ai/guides/process-mining-101`. Accessed on 01-03-2023.

[21] Celonis, 2022. URL `https://www.celonis.com/process-mining/what-is-process-mining/`. Accessed on 01-03-2023.

[22] UIPath process mining, 2022. URL `https://www.uipath.com/product/process-mining`. Accessed on 01-03-2023.

[23] IBM process mining, 2022. URL `https://www.ibm.com/uk-en/products/process-mining`. Accessed on 01-03-2023.

[24] Aalst W. Process mining: a 360 degree overview. In *Process Mining Handbook*, pages 3–34. Springer, 2022.

[25] Kothari CR. *Research Methodology Methods and Techniques*. New Age International P Ltd., Publishers, New Delhi, 2nd rev. ed. edition, 2004.

[26] Maxwell J. Qualitative research design : An interactive approach / j.a. maxwell. 01 2012.

[27] 4TU. BPI challenge 2013 closed problems, 2013. URL `https://data.4tu.nl/articles/dataset/BPI_Challenge_2013_closed_problems/12714476`. Accessed on 01-03-2023.

[28] Alrahili R. Towards employing process mining for role-based access control analysis: a systematic literature review. In *Proceedings of the Future Technologies Conference*, pages 904–927. Springer, 2021.

[29] Alrahili R. On the usage of process mining for access control analysis. The 2019 NMS PGR Poster Competition, Poster presented at King's College London, London, UK, .

[30] Alrahili R. A multi perspective conformance checking approach of role-based access control. .

[31] Song M and Aalst W. Towards comprehensive support for organizational mining. *Decision Support Systems*, 46(1):300–317, 2008.

[32] Aalst W. *Process-Aware Information Systems: Lessons to Be Learned from Process Mining*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[33] Verbeek HMW, Buijs J, van Dongen B, and Aalst W. Xes, xesame, and ProM 6.

In *International conference on advanced information systems engineering*, pages 60–75. Springer, 2010.

[34] Günther C and Verbeek E. Standard definition. *Fluxicon Process Laboratories, XES Version*, 1, 2012.

[35] IEEE task force on process mining: XES standard definition, 2016. URL `http://www.xes-standard.org/`. Accessed on 01-03-2023.

[36] Rinderle-Ma S and Aalst W. Life-cycle support for staff assignment rules in process-aware information systems. 2007.

[37] Russell N, Aalst W, Ter Hofstede AHM, and Edmond D. Workflow resource patterns: Identification, representation and tool support. In *International conference on advanced information systems engineering*, pages 216–232. Springer, 2005.

[38] Carmona J, van Dongen B, and Weidlich M. Conformance checking: foundations, milestones and challenges. In *Process Mining Handbook*, pages 155–190. Springer, 2022.

[39] CPN tools, 2022. URL `https://cpntools.org/`. Accessed on 01-03-2023.

[40] Sandhu R, Coyne EJ, Feinstein HL, and Youman CE. Role-based access control models. *Computer*, 29(2):38–47, 1996.

[41] Osborn S, Sandhu R, and Munawer Q. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(2):85–106, 2000.

[42] Sandhu R, Ferraiolo DF, and Kuhn DR. The NIST model for role-based access control: towards a unified standard. In *ACM workshop on Role-based access control*, volume 10, 2000.

[43] Ferraiolo DF, Cugini J, and Kuhn DR. Role-based access control (RBAC): Features and motivations. In *Proceedings of 11th annual computer security application conference*, pages 241–48, 1995.

[44] Bertino E, Bonatti PA, and Ferrari E. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC)*, 4(3):191–233, 2001.

[45] Joshi J, Bertino E, Latif U, and Ghafoor A. A generalized temporal role-based access control model. *IEEE transactions on knowledge and data engineering*, 17 (1):4–23, 2005.

[46] Crampton J. Specifying and enforcing constraints in role-based access control. In *Proceedings of the eighth ACM symposium on Access control models and technologies*, pages 43–50, 2003.

[47] Ahn GJ and Sandhu R. Role-based authorization constraints specification. *ACM Transactions on Information and System Security (TISSEC)*, 3(4):207–226, 2000.

[48] Botha RA and Eloff JHP. Separation of duties for access control enforcement in workflow environments. *IBM Systems Journal*, 40(3):666–682, 2001.

[49] Chadwick DW, Xu W, Otenko S, Laborde R, and Nasser B. Multi-session separation of duties (MSoD) for RBAC. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*, pages 744–753. IEEE, 2007.

[50] Esna-Ashari M, Rabiee HR, and Mirian-Hosseinabadi S. Reliability of separation of duty in ANSI standard role-based access control. *Scientia Iranica*, 18(6): 1416–1424, 2011.

[51] Lu J, Li R, Hu J, and Xu D. Static enforcement of static separation-of-duty policies in usage control authorization models. *IEICE transactions on communications*, 95(5):1508–1518, 2012.

[52] Kuhn DR. Mutual exclusion of roles as a means of implementing separation of duty in role-based access control systems. In *Proceedings of the second ACM workshop on Role-based access control*, pages 23–30, 1997.

[53] Li N, Tripunitara MV, and Bizri Z. On mutually exclusive roles and separation-of-duty. *ACM Transactions on Information and System Security (TISSEC)*, 10 (2):5–es, 2007.

[54] Sandhu R. Separation of duties in computerized information systems. In *DBSec*, pages 179–190. Citeseer, 1990.

[55] Simon RT and Zurko ME. Separation of duty in role-based environments. In *Proceedings 10th Computer Security Foundations Workshop*, pages 183–194. IEEE, 1997.

[56] Ultra JD and Pancho-Festin S. A simple model of separation of duty for access control models. *Computers & Security*, 68:69–80, 2017.

[57] Sandhu R. Separation of duties in computerized information systems. In *DBSec*, pages 179–190. Citeseer, 1990.

[58] Nash MJ and Poland KR. Some conundrums concerning separation of duty. In *IEEE Symposium on Security and Privacy*, pages 201–209. Citeseer, 1990.

[59] Schefer S, Strembeck M, and Mendling J. Checking satisfiability aspects of binding constraints in a business process context. In *International Conference on Business Process Management*, pages 465–470. Springer, 2011.

[60] Strembeck M and Mendling J. Generic algorithms for consistency checking of mutual-exclusion and binding constraints in a business process context. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 204–221. Springer, 2010.

[61] Tan K, Crampton J, and Gunter CA. The consistency of task-based authorization constraints in workflow. In *Proceedings. 17th IEEE Computer Security Foundations Workshop, 2004.*, pages 155–169. IEEE, 2004.

[62] Casati F, Castano S, and Fugini M. Managing workflow authorization constraints through active database technology. *Information Systems Frontiers*, 3(3):319–338, 2001.

[63] Wainer J, Barthelmess P, and Kumar A. W-RBAC—A workflow security model incorporating controlled overriding of constraints. *International Journal of Cooperative Information Systems*, 12(04):455–485, 2003.

[64] Muhammad Aqib and Riaz Ahmed Shaikh. Analysis and comparison of access control policies validation mechanisms. *International Journal of Computer Network and Information Security*, 7(1):54–69, 2015.

[65] Vincent Hu. Machine learning for access control policy verification. Technical report, Technical Report, 2021.

[66] Margrave. The margrave policy analyzer, 2021. URL `http://www.margrave-tool.org/`. Accessed on 16-06-2023.

[67] National Institute of Standards and Technology. Access control rule logic circuit simulation (ACRLCS), 2021. URL `https://csrc.nist.gov/Projects/Access-Control-Policy-Tool/`

access-control-rule-logic-circuit-simulation. Accessed on 16-06-2023.

[68] NuSMV. NuSMV: A new symbolic model checker., 2021. URL https://nusmv.fbk.eu/. Accessed on 16-06-2023.

[69] Chao Huang, Jianling Sun, Xinyu Wang, and Yuanjie Si. Inconsistency management of role base access control policy. In *2009 International Conference on E-Business and Information System Security*, pages 1–5. IEEE, 2009.

[70] Bei Wu, Xing-yuan Chen, Yong-fui Zhang, and Xiang-dong Dai. An extensible intra access control policy conflict detection algorithm. In *2009 International Conference on Computational Intelligence and Security*, volume 1, pages 483–488. IEEE, 2009.

[71] XACML. Oasis extensible access control markup language (XACML) TC, 2021. URL https://www.oasis-open.org/committees/tc_home.phpwg_abbrev=xacml. Accessed on 16-06-2023.

[72] Masahiro Fujita, Patrick C. McGeer, and JC-Y Yang. Multi-terminal binary decision diagrams: An efficient data structure for matrix representation. *Formal methods in system design*, 10:149–169, 1997.

[73] Kathi Fisler, Shriram Krishnamurthi, Leo A Meyerovich, and Michael Carl Tschantz. Verification and change-impact analysis of access-control policies. In *Proceedings of the 27th international conference on Software engineering*, pages 196–205, 2005.

[74] Salman Saghafi, Tim Nelson, and Daniel J Dougherty. Geometric logic for policy analysis. In *International Workshop on Automated Reasoning in Security and Software Verification (ARSEC 2013)*, pages 12–20, 2013.

[75] Mohammad Nur Nobi, Maanak Gupta, Lopamudra Praharaj, Mahmoud Abdelsalam, Ram Krishnan, and Ravi Sandhu. Machine learning in access control: A taxonomy and survey. *arXiv preprint arXiv:2207.01739*, 2022.

[76] John Heaps, Ram Krishnan, Yufei Huang, Jianwei Niu, and Ravi Sandhu. Access control policy generation from user stories using machine learning. In *Data and Applications Security and Privacy XXXV: 35th Annual IFIP WG 11.3 Conference, DBSec 2021, Calgary, Canada, July 19–20, 2021, Proceedings 35*, pages 171–188. Springer, 2021.

[77] Ravi Mukkamala, Vishnu Kamisetty, and Pawankumar Yedugani. Detecting and resolving misconfigurations in role-based access control (short paper). In *Information Systems Security: 5th International Conference, ICISS 2009 Kolkata, India, December 14-18, 2009 Proceedings 5*, pages 318–325. Springer, 2009.

[78] Lujo Bauer, Scott Garriss, and Michael K Reiter. Detecting and resolving policy misconfigurations in access-control systems. *ACM Transactions on Information and System Security (TISSEC)*, 14(1):1–28, 2011.

[79] Dianxiang Xu, Lijo Thomas, Michael Kent, Tejeddine Mouelhi, and Yves Le Traon. A model-based approach to automated testing of access control policies. In *Proceedings of the 17th ACM symposium on Access Control Models and Technologies*, pages 209–218, 2012.

[80] Nyame G and Qin Z. Precursors of role-based access control design in kms: A conceptual framework. *Information*, 11(6):334, 2020.

[81] Dan Lin, Prathima Rao, Elisa Bertino, Ninghui Li, and Jorge Lobo. EXAM: a comprehensive environment for the analysis of access control policies. *International Journal of Information Security*, 9:253–273, 2010.

[82] Kitchenham B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 2004.

[83] Mariano DC, Leite C, Santos LH, Rocha RE, and de Melo-Minardi RC. A guide to performing systematic literature reviews in bioinformatics. *arXiv preprint arXiv:1707.05813*, 2017.

[84] Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, and Linkman S. Systematic literature reviews in software engineering–a systematic literature review. *Information and software technology*, 51(1):7–15, 2009.

[85] Moher D, Liberati A, Tetzlaff J, Altman DG, and PRISMA Group*. Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *Annals of internal medicine*, 151(4):264–269, 2009.

[86] Baumgrass A, Baier T, Mendling J, and Strembeck M. Conformance checking of RBAC policies in process-aware information systems. In *International Conference on Business Process Management*, pages 435–446. Springer, 2011.

[87] Jans M, Van Der Werf JM, Lybaert N, and Vanhoof K. A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications*, 38(10):13351–13359, 2011.

[88] Sellami R, Gaaloul W, and Moalla S. An ontology for workflow organizational model mining. In *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pages 199–204. IEEE, 2012.

[89] Hanachi C, Gaaloul W, and Mondi R. Performative-based mining of workflow organizational structures. In *International Conference on Electronic Commerce and Web Technologies*, pages 63–75. Springer, 2012.

[90] Accorsi R and Stocker T. On the exploitation of process mining for security audits: the conformance checking case. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 1709–1716. ACM, 2012.

[91] Baumgrass A, Schefer-Wenzl S, and Strembeck M. Deriving process-related RBAC models from process execution histories. In *2012 IEEE 36th Annual Computer Software and Applications Conference Workshops*, pages 421–426, July 2012. doi: 10.1109/COMPSACW.2012.80.

[92] Leitner M, Baumgrass A, Schefer-Wenzl S, Rinderle-Ma S, and Strembeck M. A case study on the suitability of process mining to produce current-state RBAC models. In La Rosa M and Soffer P, editors, *Business Process Management Workshops*, pages 719–724, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-36285-9.

[93] Leitner M. Delta analysis of role-based access control models. In *International Conference on Computer Aided Systems Theory*, pages 507–514. Springer, 2013.

[94] Burattin A, Sperduti A, and Veluscek M. Business models enhancement through discovery of roles. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 103–110. IEEE, 2013.

[95] Accorsi R, Stocker T, and Müller G. On the exploitation of process mining for security audits: the process discovery case. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 1462–1468. ACM, 2013.

[96] Zhao W and Zhao X. Process mining from the organizational perspective. In *Foundations of Intelligent Systems*, pages 701–708. Springer, 2014.

[97] Schönig S, Cabanillas Macias C, Di Ciccio C, Jablonski S, and Mendling J.

Mining resource assignments and teamwork compositions from process logs. *Softwaretechnik-Trends*, 36(4):1–6, 2016.

[98] Zahoransky R, Holderer J, Lange A, and Brenig C. Process analysis as first step towards automated business security. 2016.

[99] Salnitri M, Alizadeh M, Giovanella D, Zannone N, and Giorgini P. From security-by-design to the identification of security-critical deviations in process executions. In Mendling J and Mouratidis H, editors, *Information Systems in the Big Data Era*, pages 218–234, Cham, 2018. Springer International Publishing. ISBN 978-3-319-92901-9.

[100] Ye J, Li Z, Yi K, and Al-Ahmari A. Mining resource community and resource role network from event logs. *IEEE Access*, 6:77685–77694, 2018. doi: 10.1109/ACCESS.2018.2883774.

[101] Cabanillas C, Schönig S, Sturm C, and Mendling J. Mining expressive and executable resource-aware imperative process models. In *Enterprise, Business-Process and Information Systems Modeling*, pages 3–18. Springer, 2018.

[102] Schönig S, Cabanillas C, Di Ciccio C, Jablonski S, and Mendling J. Mining team compositions for collaborative work in business processes. *Software & Systems Modeling*, 17(2):675–693, 2018.

[103] Alizadeh M, Lu X, Fahland D, Zannone N, and Aalst W. Linking data and process perspectives for conformance analysis. *Computers and Security*, 73: 172 – 193, 2018. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2017.10.010. URL `http://www.sciencedirect.com/science/article/pii/S0167404817302262`.

[104] Havur G and Cabanillas C. History-aware dynamic process fragmentation for risk-aware resource allocation. In Panetto H, Debruyne C, Hepp M, Lewis D, Ardagna C, and Meersman R, editors, *On the Move to Meaningful Internet Systems: OTM 2019 Conferences*, pages 533–551, Cham, 2019. Springer International Publishing.

[105] Cabanillas C, Ackermann L, Schönig S, Sturm C, and Mendling J. The RALph miner for automated discovery and verification of resource-aware process models. *Software and Systems Modeling*, pages 1–27, 2020.

[106] Asare E, Wang L, and Fang X. Conformance checking: Workflow of hospitals and workflow of open-source EMRs. *IEEE Access*, 8:139546–139566, 2020.

[107] McGee S and Greer D. Towards an understanding of the causes and effects of software requirements change: two case studies. *Requirements Engineering*, 17 (2):133–155, 2012.

[108] Petersen K, Vakkalanka S, and Kuzniarz L. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.

[109] Leitner M and Rinderle-Ma S. A systematic review on security in process-aware information systems–constitution, challenges, and future directions. *Information and Software Technology*, 56(3):273–293, 2014.

[110] Kelemen R. Systematic review on process mining and security. In *Central and Eastern European e| Dem and e| Gov Days 2017*, 2017.

[111] Arias M, Saavedra R, Marques MR, Munoz-Gama J, and Sepúlveda M. Human resource allocation in business process management and process mining. *Management Decision*, 2018.

[112] Garcia C, Meincheim A, Junior ER, Dallagassa MR, Sato DM, Carvalho DR, Santos EA, and Scalabrin EE. Process mining techniques and applications – a systematic mapping study. *Expert Systems with Applications*, 133:260 – 295, 2019. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2019.05.003. URL `http://www.sciencedirect.com/science/article/pii/S0957417419303161`.

[113] Runeson P and Höst M. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131, 2009.

[114] Yin RK. *Case Study Research: Design and Methods*, volume 5. sage, 2009.

[115] Erdogan TG and Tarhan A. Systematic mapping of process mining studies in healthcare. *IEEE Access*, 6:24543–24567, 2018.

[116] Batista E and Solanas A. Process mining in healthcare: A systematic review. In *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–6, 2018.

[117] Martin N, Wittig N, and Munoz-Gama J. Using process mining in healthcare. In *Process Mining Handbook*, pages 416–444. Springer International Publishing Cham, 2022.

[118] Sun R, Karaca Z, and Wong HS. Trends in hospital emergency department visits by age and payer, 2006-2015: statistical brief# 238. *Healthcare Cost and Utilization Project (HCUP) Statistical Briefs. Rockville, MD: Agency for Healthcare Research and Quality*, 2018.

[119] Ivan Shugurov and Alexey Mitsyuk. Generation of a set of event logs with noise. 01 2014. doi: 10.15514/SYRCOSE-2014-8-13.

[120] BPM 2013. 9th international workshop on business process intelligence 2013, 2013. URL https://www.win.tue.nl/bpi/2013/challenge.html. Accessed on 16-07-2023.

[121] Arjel Bautista, Syed Akbar, Anthony Alvarez, Tom Metzger, and Marshall Reaves. Process mining in information technology incident management: A case study at volvo belgium. In *BPIC@ BPM*, 2013.

[122] Peter Van den Spiegel, Leen Dieltjens, and Liese Blevi. Bpi challenge 2013-applied process mining techniques for incident and problem management. *Proceedings of the 3rd Business Process Intelligence Challenge (BPIC 2013). http://ceur-ws. org*, 1052, 2013.

[123] Chang Jae Kang, Young Sik Kang, Yeong Shin Lee, Seonkyu Noh, Hyeong Cheol Kim, Woo Cheol Lim, Juhee Kim, and Regina Hong. Process mining-based understanding and analysis of volvo it's incident and problem management processes. *BPIC@ BPM*, 85, 2013.

[124] Minstry of Justice. It scurity inceient management policy, 2023. URL https://security-guidance.service.justice.gov.uk/it-security-incident-management-policy. Accessed on 21-07-2023.

[125] Maggi FM, Westergaard M, Montali M, and Aalst W. Runtime verification of LTL-based declarative process models. In *International Conference on Runtime Verification*, pages 131–146. Springer, 2011.

[126] Aalst W, De Beer H, and van Dongen B. Process mining and verification of properties: An approach based on temporal logic. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 130–147. Springer, 2005.

[127] De Beer H and Van den Brand H. The LTL checker plugins a (reference) manual eindhoven university of technology. 8. 2004.

[128] Aalst W. Extracting event data from databases to unleash process mining. *BPM-Driving innovation in a digital world*, pages 105–128, 2015.

[129] Weerdt J and Wynn MT. Foundations of process event data. *Process Mining Handbook. LNBIP*, 448:193–211, 2022.

[130] M Weske. Business process management–concepts, languages, architectures, verlag. *Berlin*, 2007.

[131] Bozkaya M, Gabriels J, and van der Werf JM. Process diagnostics: a method based on process mining. In *2009 International Conference on Information, Process, and Knowledge Management*, pages 22–27. IEEE, 2009.

[132] Yang J, Ouyang C, Aalst W, Hofstede A, and Yu Y. Ordinor: A framework for discovering, evaluating, and analyzing organizational models using event logs. *Decision Support Systems*, 158:113771, 2022.

[133] Baumgrass A. Deriving current state RBAC models from event logs. In *2011 Sixth International Conference on Availability, Reliability and Security*, pages 667–672. IEEE, 2011.

[134] Aalst W. Business alignment: using process mining as a tool for delta analysis and conformance testing. *Requirements engineering*, 10(3):198–211, 2005.

[135] Aalst W, Hee K, Van der Werf JM, and Verdonk M. Auditing 2.0: Using process mining to support tomorrow's auditor. *Computer*, 43:90 – 93, 04 2010. doi: 10.1109/MC.2010.61.

[136] Taghiabadi E, Gromov V, Fahland D, and Aalst W. Compliance checking of data-aware and resource-aware compliance requirements. In *OTM Confederated International Conferences*, pages 237–257. Springer, 2014.

[137] Mannhardt F. Multi-perspective process mining. In *BPM (Dissertation/Demos/Industry)*, pages 41–45, 2018.

[138] Ferreira A, Cruz-Correia R, Antunes L, Farinha P, Oliveira-Palhares E, Chadwick DW, and Costa-Pereira A. How to break access control in a controlled manner. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, pages 847–854. IEEE, 2006.

[139] Aalst W, Adriansyah A, and van Dongen B. Replaying history on process models for conformance checking and performance analysis. *Wiley Int. Rev. Data Min. and Knowl. Disc.*, 2(2):182–192, March 2012. ISSN 1942-4787. doi: 10.1002/widm.1045. URL `http://dx.doi.org/10.1002/widm.1045`.

[140] Garvey PR and Lansdowne ZF. Risk matrix: an approach for identifying, assessing, and ranking program risks. *Air Force Journal of Logistics*, 22(1):18–21, 1998.

[141] Ni H, Chen A, and Chen N. Some extensions on risk matrix approach. *Safety Science*, 48(10):1269–1278, 2010.

[142] Bernsmed K, Frøystad C, Meland PH, Nesheim DA, and Rødseth OJ. Visualizing cyber security risks with bow-tie diagrams. In *International Workshop on Graphical Models for Security*, pages 38–56. Springer, 2017.

[143] Arya Adriansyah, Boudewijn F van Dongen, and Wil MP van der Aalst. Conformance checking using cost-based fitness analysis. In *2011 ieee 15th international enterprise distributed object computing conference*, pages 55–64. IEEE, 2011.

[144] Xixi Lu, Dirk Fahland, and Wil MP van der Aalst. Conformance checking based on partially ordered event data. In *Business Process Management Workshops: BPM 2014 International Workshops, Eindhoven, The Netherlands, September 7-8, 2014, Revised Papers 12*, pages 75–88. Springer, 2015.

[145] Appari A and Johnson ME. Information security and privacy in healthcare: current state of research. *International journal of Internet and enterprise management*, 6(4):279–314, 2010.

[146] Campbell AR, Layne D, Scott E, and Wei H. Interventions to promote teamwork, delegation and communication among registered nurses and nursing assistants: An integrative review. *Journal of Nursing Management*, 28(7):1465–1472, 2020.

[147] Maw HA, Xiao H, Christianson B, and Malcolm JA. BTG: Break-the-glass access control model for medical data in wireless sensor networks. *IEEE journal of biomedical and health informatics*, 20(3):763–774, 2015.

[148] Awwad K, Ng YG, Lee K, Lim PY, and Rawajbeh B. Determination of the triage skill and knowledge levels of prehospital emergency medical staff: A cross-sectional study. *International Emergency Nursing*, 64:101203, 2022.

[149] Cooper S, Cant R, Porter J, Sellick K, Somers G, Kinsman L, and Nestel D. Rating medical emergency teamwork performance: development of the team emergency assessment measure (TEAM). *Resuscitation*, 81(4):446–452, 2010.

[150] Boadu EO and Armah GK. Role-based access control (RBAC) based in hospital management. *Int. J. Softw. Eng. Knowl. Eng*, 3:53–67, 2014.

[151] Caron F, Vanthienen J, and Baesens B. Comprehensive rule-based compliance checking and risk management with process mining. *Decision Support Systems*, 54(3):1357–1369, 2013.

[152] Leoni M and Aalst W. Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming. In *Business process management*, pages 113–129. Springer, 2013.

[153] Mannhardt F, De Leoni M, Reijers HA, and Aalst W. Balanced multi-perspective checking of process conformance. *Computing*, 98(4):407–437, 2016.

[154] Leoni M, Aalst W, and van Dongen B. Data-and resource-aware conformance checking of business processes. In *International Conference on Business Information Systems*, pages 48–59. Springer, 2012.

[155] Felli P, Gianola A, Montali M, Rivkin A, and Winkler S. Cocomot: Conformance checking of multi-perspective processes via smt. In *International Conference on Business Process Management*, pages 217–234. Springer, 2021.

# Appendices

# Appendix A

# SLR tables

## A.1 PRIMA checklist

Table A.1 shows the PRIMA check-list of the systematic literature review in Chapter 3.

| Topic | Num | Check-list item | Page |
|---|---|---|---|
| | | TITLE | |
| Title | 1 | Identify the report as a systematic review. | 36 |
| | | ABSTRACT | |
| Abstract | 2 | Provide a structured summary including, as applicable: background; objectives; data sources; study eligibility criteria, participants, and interventions; study appraisal and synthesis methods; results; limitations; conclusions and implications of key findings; systematic review registration number. | 38 |
| | | INTRODUCTION | |

Table A.1: PRIMA check-list

| Topic | Num | Check-list item | Page |
|---|---|---|---|
| Rationale | 3 | Describe the rationale for the review in the context of existing knowledge. | 38 |
| Objectives | 4 | Provide an explicit statement of the objective(s) or question(s) the review addresses. | 41 |
| | | METHODS | |
| Protocol and registration | 5 | Indicate if a review protocol exists, if and where it can be accessed (e.g., Web address), and, if available, provide registration information including registration number. | 39 |
| Eligibility criteria | 6 | Specify study characteristics (e.g., PICOS, length of followup) and report characteristics (e.g., years considered, language, publication status) used as criteria for eligibility, giving rationale. | 39 |
| Information sources | 7 | Describe all information sources (e.g., databases with dates of coverage, contact with study authors to identify additional studies) in the search and date last searched. | - |
| Search | 8 | Present full electronic search strategy for at least one database, including any limits used, such that it could be repeated. | - |
| Selection selection | 9 | State the process for selecting studies (i.e., screening, eligibility, included in systematic review, and, if applicable, included in the meta-analysis). | 39 |

Table A.1: PRIMA check-list

| Topic | Num | Check-list item | Page |
|---|---|---|---|
| Data collection process | 10 | Describe method of data extraction from reports (e.g., piloted forms, independently, in duplicate) and any processes for obtaining and confirming data from investigators. | 39 |
| Data items | 11 | List and define all variables for which data were sought (e.g., PICOS, funding sources) and any assumptions and simplifications made. | - |
| Risk of bias in individual studies | 12 | Describe methods used for assessing risk of bias of individual studies (including specification of whether this was done at the study or outcome level), and how this information is to be used in any data synthesis. | 64 |
| Summary measures | 13 | State the principal summary measures (e.g., risk ratio, difference in means). | 64 |
| Synthesis of results | 14 | Describe the methods of handling data and combining results of studies, if done, including measures of consistency (e.g., I2 ) for each meta-analysis. | 64 |
| Risk of bias across studies | 15 | Specify any assessment of risk of bias that may affect the cumulative evidence (e.g., publication bias, selective reporting within studies). | 64 |
| Additional analyses | 16 | Describe methods of additional analyses (e.g., sensitivity or subgroup analyses, meta-regression), if done, indicating which were pre-specified. | 64 |
| RESULTS | | | |

Table A.1: PRIMA check-list

| Topic | Num | Check-list item | Page |
|---|---|---|---|
| Study selection | 17 | Give numbers of studies screened, assessed for eligibility, and included in the review, with reasons for exclusions at each stage, ideally with a flow diagram. | 42 |
| Study characteristics | 18 | For each study, present characteristics for which data were extracted (e.g., study size, PICOS, follow up period) and provide the citations. | 42 |
| Risk of bias in studies | 19 | Present data on risk of bias of each study and, if available, any outcome- level assessment (see item 12). | 45 |
| Results of individual studies | 20 | For all outcomes considered (benefits or harms), present, for each study: (a) simple summary data for each intervention group (b) effect estimates and confidence intervals, ideally with a forest plot. | 39 |
| Synthesis of results | 21 | Present results of each meta-analysis done, including confidence intervals and measures of consistency. | 39 |
| Risk of bias across studies | 22 | Present results of any assessment of risk of bias across studies (see Item 15). | 64 |
| Additional analysis | 23 | Give results of additional analyses, if done (e.g., sensitivity or subgroup analyses, meta-regression [see Item 16]). | 64 |
| DISCUSSION | | | |

Table A.1: PRIMA check-list

| Topic | Num | Check-list item | Page |
|---|---|---|---|
| Summary of evidence | 24 | Summarize the main findings including the strength of evidence for each main outcome; consider their relevance to key groups (e.g., healthcare providers, users, and policy makers). | 58 |
| Limitations | 25 | Discuss limitations at study and outcome level (e.g., risk of bias), and at review-level (e.g., incomplete retrieval of identified research, reporting bias). | 58 |
| Conclusions | 26 | Provide a general interpretation of the results in the context of other evidence, and implications for future research. | 66 |
| | | FUNDING | |
| Funding | 27 | Describe sources of funding for the systematic review and other support (e.g., supply of data); role of funders for the systematic review. | - |

Table A.1: PRIMA check-list

## A.2 The extended classifications

Table A.2 shows the classification of the approaches that are included in the works. Note that (M: methods, PD: process discovery, E: enhancement, CC: conformance checking, Org:organisational model).

| ID | Ref | Input | Output | M | Tool |
|---|---|---|---|---|---|
| | | | Organisational mining | | |
| S6 | [31] | Log | Org | PD | Org miner (ProM) |
| S6 | [31] | Log | Org | E | replacement filter (ProM) |
| S8 | [87] | Log | Org | PD | FSM, Fuzzy miner (ProM) |
| S9 | [88] | Log | hierarchical relationships | PD | 3 population plug ins (ProM) |
| S10 | [89] | enriched Log | Org | PD | Plugins (ProM) |
| S12 | [91] | Log, role extension | current RBAC model | PD | role derivation |
| S13 | [92] | Log | Org | PD | organizational mining (ProM) |
| S16 | [95] | Log | activity transitions | PD | Fuzzy miner |
| S16 | [95] | Log | Org | PD | Heuristic-based algorithms |
| S16 | [95] | Log | Org | PD | search-based algorithms |
| | | | Social network mining | | |
| S1 | [13] | Log | social network | PD | MiSoN |

Table A.2: The classification of the approaches

| ID | Ref | Input | Output | M | Tool |
|---|---|---|---|---|---|
| S6 | [31] | Log | social network | PD | social network miner (ProM) |
| S6 | [31] | social , Org | social network | PD | grouping plug in (ProM) |
| S6 | [31] | Log, process model | social network measures | E | social network analysis (ProM) |
| Staff Assignment Rules (SAR) mining | | | | | |
| S3 | [14] | Log, Org | SAR | PD | staff assignment mining |
| S5 | [36] | Log, Org | SAR, decision tree | PD | a number of plugins (ProM) |
| S13 | [92] | Log, Org | SAR | PD | staff assignment mining (ProM) |
| S18 | [97] | Log, Org | SAR | PD | DPIL |
| S22 | [101] | Log, org info | Org | PD | RALph |
| S23 | [102] | Log, Org | SAR | PD | DpilMiner |
| S25 | [104] | Log, model, org info | SAR | PD | ASP |
| S26 | [105] | Log | SAR | PD | RALph |
| Role mining | | | | | |
| S4 | [15] | Log | Org | PD | - |
| S13 | [92] | Log | related RBAC models | PD | role derivation, xoRET |
| S13 | [92] | Log | hierarchical structure | PD | role hierarchy miner (ProM) |

Table A.2: The classification of the approaches

| ID | Ref | Input | Output | M | Tool |
|---|---|---|---|---|---|
| S15 | [94] | Log, process model | activities in roles | E | plug in (ProM) |
| S21 | [100] | Log, social network | resource role networks | PD | 3 algorithms |
| | | | Check properties | | |
| S2 | [17] | Log, Org | incompliant traces | CC | LTL checker (ProM) |
| S5 | [36] | Log | incompliant traces | CC | LTL checker (ProM) |
| S7 | [86] | Log, RBAC model | RBAC inconsistency | CC | LTL checker (ProM) |
| S8 | [87] | Log | incompliant traces | CC | LTL checker (ProM) |
| S11 | [90] | Log | a tasks/originator matrix | CC | Originator by Task matrix (ProM) |
| S11 | [90] | Log | incompliant traces | CC | LTL checker (ProM) |
| S11 | [90] | Log | incompliant traces | CC | SCIFF checker (ProM) |
| S14 | [93] | two RBAC models | Verification | CC | |
| S19 | [98] | Log | incompliant traces | CC | SCIFF (SWAT) |

Table A.2: The classification of the approaches

| ID | Ref | Input | Output | M | Tool |
|---|---|---|---|---|---|
| S20 | [99] | Log, model, policies | incompliant traces | CC | STS Tool |
| S24 | [103] | Log, model, CRUD matrix | incompliant traces | CC | Plug in (ProM) |
| S26 | [105] | Log, model | Verification | CC | RALph |
| S27 | [106] | Log, model, PN-Result | alignment and replay | CC | Plugin(ProM) |

Table A.2: The classification of the approaches

# Appendix B

# One R Algorithm

Here is the OneR algorithm code that I used in Chapter 4.

```
import sys
import csv
import pandas as pd
import numpy as np
import math
import random


DEBUGGING = True
DATA_FILE = 'log1.csv'


# MAIN
#-get data from a file
try:
```

```
    df1 = pd.read_csv( DATA_FILE, na_filter=False )
except IOError as iox:
    print('there was an I/O error trying to open the
    data file: ' + str( iox ))
    sys.exit()
#-get size of raw data set
N = len( df1.columns )
M = len( df1.values )

#-print columns
if DEBUGGING:
    print('INPUT FILE = ' + DATA_FILE)
    print('number of attributes = ' + str( N ))
    print('number of instances = ' + str( M ))
    for ( i, c, t ) in zip( range( N ), df1.columns,
    df1.dtypes ):
        print('{} - {} ({})'.format( i, c, t ))

df = df1.drop(['startTime'], axis=1)
df = df.drop(['completeTime'], axis=1)
df = df.drop(['case'], axis=1)
df = df.drop(['event1'], axis=1)



#-get size of raw data set
```

```
N = len( df.columns )
M = len( df.values )

#-print columns
if DEBUGGING:
    print('INPUT FILE = ' + DATA_FILE)
    print('number of attributes = ' + str( N ))
    print('number of instances = ' + str( M ))
    for ( i, c, t ) in zip( range( N ), df.columns,
    df.dtypes ):
        print('{} - {} ({})'.format( i, c, t ))

y = df.event
df = df.drop(['event'], axis=1)
X = df.values

#-get size of raw data set
N = len( df.columns )
M = len( df.values )

#if ( DEBUGGING ):
#    for i in range( M ):
#    print(X[i], y[i])

# get unique list of class values
```

```
classes = sorted( set( y ))
num_classes = len( classes )
if ( DEBUGGING ):
    print('classes = ', classes)

# get unique lists of values for each attribute
attr_values = []
for i in range( N ):
    attr_values.append( sorted( set( X[:,i] )))
if ( DEBUGGING ):
    print('attribute values =')
    for i in range( N ):
        print(df.columns[i], attr_values[i])

attr_dict = [[ dict() for k in range
(len( attr_values[i] ))] for i in range( N )]
for i in range( N ):
    for k in range( len( attr_values[i] )):
        for c in range( num_classes ):
            attr_dict[i][k][c] = 0

# for each attribute value, count the number of
#occurrences of each class
for j in range( M ): # loop through all instances
    c = y[j] # save the class for this instance
```

```
    for i in range( N ):
        # find index of attribute value X[j,i] in
        attributes[i]
        k = attr_values[i].index( X[j][i] )
        if ( k < 0 ):
            print('ERROR! (attribute, value) not found:
            (', i, X[j][i], ')')
        else:
            attr_dict[i][k][c] += 1

# for each attribute value, find the most frequent class
#(0, 1 or 2)
attr_value_class = [[ -1 for k in range
(len(attr_values[i])) ] for i in range( N ) ]
for i in range( N ):
    for k in range( len( attr_values[i] )):
        most_freq = 0
        for c in range( num_classes ):
            if ( attr_dict[i][k][c] > attr_dict[i][k]
            [most_freq] ):
                most_freq = c
        attr_value_class[i][k] = most_freq

# display table of counts and most frequent class for
#each single attribute
```

```
if ( DEBUGGING ):
    for i in range( N ):
        for k in range( len( attr_values[i] )):
            print('attribute ', df.columns[i])
            print(' value ', attr_values[i][k])
            print(' count = (' )
            for c in range( num_classes ):
                print(attr_dict[i][k][c])
            print(')  most frequent=',
            attr_value_class[i][k])

# build 1R set of rules using attributes 0 and 1
rules = [[ -1 for i0 in range( len( attr_values[0] ))]
for i1 in range( len( attr_values[1] )) ]
for k1 in range( len( attr_values[1] )):
    for k0 in range( len( attr_values[0] )):
        most_freq_0 = attr_value_class[0][k0]
        most_freq_1 = attr_value_class[1][k1]
        if ( most_freq_0 == most_freq_1 ):
            # the same class is the most frequent for
            #both attribute values
            rule_class = most_freq_0
        elif ( attr_dict[0][k0][most_freq_0] >
        attr_dict[1][k1][most_freq_1] ):
            # the k0-th attribute value occurs more
```

```
            #frequently than the k1-th attribute
            #value, so select the k0-th class
            rule_class = most_freq_0
        elif ( attr_dict[1][k1][most_freq_1] >
        attr_dict[0][k0][most_freq_0] ):
            # the k1-th attribute value occurs more
            #frequently than the k0-th attribute value,
            #so select the k1-th class
            rule_class = most_freq_1
        else:
            # the k0-th and k1-th attributes occur with
            #the same frequency, to randomly select
            #between them to decide which class
            if ( random.random() < 0.5 ):
                rule_class = most_freq_0
            else:
                rule_class = most_freq_1
        rules[k1][k0] = rule_class

# display the rules
print('\nAnd the RULES are...')
if ( DEBUGGING ):
    for k1 in range( len( attr_values[1] )):
        for k0 in range( len( attr_values[0] )):
            print('({},{})=({},{}) -> {} ({})'.format
```

```
            (df.columns[1], df.columns[0],
            attr_values[1][k1],
            attr_values[0][k0], rules[k1][k0],
            classes[rules[k1][k0]] ))

# score the rules
count = 0.0
# loop through instances
for j in range( M ):
    # find index in attribute values lists for the
    #attribute values in this instance
    k0 = attr_values[0].index( X[j,0] )
    if ( k0 < 0 ):
        print('ERROR finding k0-th attribute: ', X[j,0])
        sys.exit()
    k1 = attr_values[1].index( X[j,1] )
    if ( k1 < 0 ):
        print('ERROR finding k1-th attribute: ', X[j,1])
        sys.exit()
    if ( y[j] == rules[k1][k0] ):
        count += 1
score = ( count / M )
print('number of correct predictions = {} out of {} =
{}'.format (count, M, score ))
```

# Appendix C

# LTL Formula

The LTL file containing the RBAC violations formulas.

```
##### Defining the attributes
set ate.EventType;
set ate.Originator;
set ate.role;
date ate.Timestamp := "yyyy-MM-dd";
set ate.WorkflowModelElement;
number pi.numSimilarInstances;

##### Renaming the attributes
rename ate.EventType as eventtype;
rename ate.EventType as event;
rename ate.Originator as originator;
rename ate.Originator as person;
rename ate.Timestamp as timestamp;
```

```
rename ate.Timestamp as time;
rename ate.role as role;
rename ate.WorkflowModelElement as modelelement;
rename ate.WorkflowModelElement as activity;
rename ate.WorkflowModelElement as place;
rename ate.WorkflowModelElement as element;
rename pi.numSimilarInstances as freq;

######### Subformulas #########
subformula R_does_A-A_not_B( R: role, A: activity,
B: activity ) :=
{<h2>Is activity A done by a member of role R? </h2>
    <p>Arguments:<br>
    <ul>
        <li><b>R</b> of type set (<i>ate.role</i>)</li>
        <li><b>A</b> of type set
        (<i>ate.WorkflowModelElement</i>)</li>
    </ul>
    </p>}
<>((activity == A /\ (activity != B /\ role == R)));


subformula P_does_A-A_not_B( P: person, A: activity,
B: activity ) :=
{<h2>Is activity A done by a person P? </h2>
```

```
    <p>Arguments:<br>
    <ul>
        <li><b>P</b> of type set
        ( <i>ate.Originator</i>)</li>
        <li><b>A</b> of type set
        ( <i>ate.WorkflowModelElement</i>)</li>
    </ul>
    </p>}
<>((activity == A /\(activity != B /\ person == P)));


subformula A_done_by_P-P_not_Q( A: activity, P: person,
Q: person ) :=
{<h2>Is activity A done by a person P? </h2>
    <p>Arguments:<br>
    <ul>
        <li><b>P</b> of type set
        ( <i>ate.Originator</i>)</li>
        <li><b>A</b> of type set
        ( <i>ate.WorkflowModelElement</i>)</li>
    </ul>
    </p>}
<>((person == P /\(person != Q /\ activity == A)));
```

```
subformula eventually_person_P( P: person ) :=
{<h2>Does person <b>P</b> perform?</h2>
    <p> If person <b>P</b> is the performer?</p>
    <p> Arguments:<br>
    <ul><li><b>P</b> of type set
        (<i>ate.Originator</i>)</li></ul></p>}
    <>( person == P );

########## Formulas ##########
formula Check_SSoD( A: activity ,
B: activity ) :=
{<h2>Is there a role doing task A and B?</h2>
    <p>Arguments:<br>
    <ul><li><b>A</b> of type set
    ( <i>ate.WorkflowModelElement</i>)</li>
    <li><b>B</b> of type set
    ( <i>ate.WorkflowModelElement</i>)</li>
    </ul></p>}
    exists[ r: role |
        ( R_does_A-A_not_B( r, A, B ) /\
        R_does_A-A_not_B( r, B, A ) ) ];


formula Check_DSoD( A: activity , B: activity ) :=
{<h2>Is there a person doing task A and B?</h2>
```

```
    <p>Arguments:<br>
    <ul><li><b>A</b> of type set
    ( <i>ate.WorkflowModelElement</i>)</li>
    <li><b>B</b> of type set
    ( <i>ate.WorkflowModelElement</i>)</li>
    </ul></p>}
    exists[ p: person |
        ( P_does_A-A_not_B( p, A, B ) /\
        P_does_A-A_not_B( p, B, A ) ) ];


formula Check_DBoD(  A: activity, B: activity ) :=
{<h2>Is there a person doing task A and not B?</h2>
    <p>Arguments:<br>
    <ul><li><b>A</b> of type set
    ( <i>ate.WorkflowModelElement</i>)</li>
    <li><b>B</b> of type set
    ( <i>ate.WorkflowModelElement</i>)</li>
    </ul></p>}
        exists[ p: person |
            exists[ q: person |
                (A_done_by_P-P_not_Q(B, p, q)
                /\A_done_by_P-P_not_Q(A, q, p))
                        ]
                ];
```

```
formula eventually_person_P_and_eventually_person_Q
_and_eventually_person_S
( P: person, Q: person, S: person) :=
{<h2>This formula to check cardinality to compute
if P, Q, and S are performers </h2>
     <p> Arguments:<br>
     <ul><li><b>P</b> of type set
        (<i>ate.Originator</i>)</li></ul>
        <li><b>Q</b> of type set
        (<i>ate.Originator</i>)</li></ul>
        <li><b>S</b> of type set
        (<i>ate.Originator</i>)</li></ul></p>}
( eventually_person_P( P ) /\ (eventually_person_P(Q)
  /\ eventually_person_P(S)));
```