Swansea University
School of Mathematics and Computer Science
MSc by Research in Computer Science

Theorising Monitoring: General Models and Access Control

Fatima Alhijji ([redacted])

Supervisor:	Prof. John Tucker
Second assessor:	Dr. Edwin Beggs
Date of submission:	10 April 2025

Declaration

This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed: [redacted]
Date: 10/04/2025

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed: [redacted]
Date: 10/04/2025

I hereby give consent for my thesis, if accepted, to be available for electronic sharing.

Signed: [redacted]
Date: 10/04/2025

The University's ethical procedures have been followed and, where appropriate, that ethical approval has been granted.

Signed: [redacted]
Date: 10/04/2025

Abstract

This thesis investigates algebraic models of monitoring that cover the essential aspects of data gathering, storing, and analysis in various contexts. The observation and recording of the behavior of a system or an entity over time is known as monitoring. Monitoring has many uses, including security, regulation compliance, and performance optimization. This thesis contributes to the development of a formal framework intended for methodically examining the attributes of monitoring systems.

This framework is based on a general model that isolates the fundamental concepts and procedures of monitoring (Johnson et al., 2017). We review the basic model with particular attention to: (i) a system of notifications that makes it easier to create and distribute alerts based on monitoring data; and (ii) interventions that enable the prevention or modification of particular behaviors based on monitoring data. We introduce a new feature namely timestamps that enhance the models treatment of time.
We apply this revised model to case studies in access control. First, we analyse the role of monitoring in the process of login into an account. Secondly, we apply it to model the role of monitoring in getting access to physical buildings. these two case studies require us to model formally passwords, keys, registration and conformance.

Contents

Chapter 1

Introduction

This dissertation explores the concept of monitoring objects and individuals. It adopts a general abstract but informal analysis of monitoring based on (Wang and Tucker, 2023), which is about 'People Watching.' In addition, it re-evaluates previous mathematical formalisations of monitoring systems presented in (Johnson et al., 2017). It enhances them by incorporating a novel analysis of the reactive capabilities of monitoring systems, a topic that had not been fully explored before. The primary contribution to the general model of monitoring lies in our investigation of the significance of timing, various types of notifications, and interventions derived from monitoring records. The revised abstract formal monitoring model is then applied to new case studies in access control systems. The research delves into creating detailed mathematical models for two seemingly straightforward scenarios: using usernames and passwords for system login and using digital keys to access secure buildings. These case studies serve as valuable tests of our newly developed general model of a monitoring system.

First, briefly contemplate some motivations for studying monitoring, and then the thesis's objectives and structure are explained. This dissertation aims to contribute to understanding monitoring systems and their role in various contexts, particularly in access control resources.

1.1 Data Monitoring and Surveillance

In an increasingly digital and interconnected world, the concepts of *monitoring* and *surveillance* have gained prominence, particularly in discussions on privacy, security, and data governance. While these terms are often used interchangeably, they have distinct meanings and implications, necessitating a clear definition and exploration of their relationship.

Monitoring refers to the systematic collection, observation, and analysis of data over time to assess performance, detect anomalies, or enhance decision-making. It is commonly employed in various fields, including environmental science, healthcare, cybersecurity, and business intelligence. The primary objective of monitoring is to gather relevant data to improve systems, ensure compliance, and facilitate informed decision-making. Monitoring can be passive or active, with the collected data used to recognize trends, optimize processes, and enhance security.

Surveillance, on the other hand, is a subset of monitoring that is often associated with control, oversight, and power asymmetry. It is defined as the systematic and focused observation of individuals, groups, or activities, typically conducted by governments, corporations, or authoritative bodies to influence, manage, protect, or regulate behavior (Lyon, 2007). Unlike general monitoring, surveillance carries implicit ethical and privacy concerns, as it involves tracking individuals, often without their explicit consent, to exert control or maintain security (Lyon, 2006).

While monitoring and surveillance share similarities in their reliance on data collection and analysis, their key distinction lies in intent and ethical implications. Monitoring is generally employed to improve efficiency, detect issues, and optimize outcomes in a broad range of contexts, whereas surveillance is characterized by an imbalance of power, where an entity exerts oversight over individuals or groups. Surveillance can be classified into different types, including government surveillance for national security, corporate surveillance for market analysis, and self-surveillance through digital platforms.

The rapid advancement of digital technologies has increasingly blurred the boundaries between monitoring and surveillance. In modern society, digital platforms, smart devices, and artificial intelligence-driven systems collect vast amounts of data for both monitoring and surveillance purposes. While some forms of monitoring, such as website analytics and predictive maintenance, are perceived as beneficial, concerns arise when monitoring transitions into intrusive surveillance, affecting personal privacy and civil liberties (Van Dijck, 2014; Graham and Wood, 2003).

(Lukovenkov, 2020) discusses how advancements in technology have made surveillance more accessible, shifting away from traditional hierarchical structures. By analyzing Jeremy Bentham's panoptic model, key elements of surveillance are identified as "visibility" and "participation." Surveillance is not only imposed by authorities but can also be voluntarily embraced through social media, where individuals contribute to self-surveillance and peer monitoring.
While surveillance is often depicted in fiction as an infringement on personal privacy, two other forms are prevalent in contemporary society: mutual surveillance within peer groups and self-surveillance. The rise of social media and ubiquitous connectivity has led to a *surveillance culture*, in which individuals participate in monitoring and exposing their own and others' activities. This has resulted in an information environment where privacy concerns are often secondary to the perceived benefits of data sharing.

The widespread implementation of monitoring in various domains has also raised ethical concerns regarding the scope, consent, and security of collected data (Wang and Tucker, 2023). For instance, companies use cookies and tracking technologies to collect behavioral data, offering personalized services while also raising concerns about the potential misuse of personal information. Similarly, employee monitoring in workplaces raises questions about autonomy and trust, as excessive oversight can create an environment of control rather than productivity.

Given these considerations, it is essential to establish ethical frameworks that ensure a balance between monitoring for legitimate purposes and the protection of individual rights. Transparency, accountability, and data governance play critical roles in preventing monitoring from becoming intrusive surveillance (Culnan and Williams, 2009). (Johnson et al., 2017) proposed an abstract theoretical framework to analyze monitoring systematically, allowing for a structured approach to its implementation while addressing ethical concerns.

There is a substantial body of literature examining the implications of surveillance, particularly in specific sectors such as law enforcement, corporate governance, and social behavior. For instance, (Parsell, 2015) explored the complex dynamics of surveillance, highlighting its dual impact on individuals and communities. Understanding these complexities provides valuable insights into the ethical dilemmas, security challenges, and social implications of data collection in contemporary society.

1.2 Problem Statement

Without a doubt, data collection and computation have made it possible for individuals and organisations to monitor our daily activities online across many sectors, such as e-commerce (Blazquez et al., 2019), health (Vincent et al., 2014), social life (Ampofo, 2011), etc. According to (Dwivedi et al., 2021), almost every member of society has become used to being monitored commercially. Although, many raise no objections even when they know they are being monitored, probably because of the positive benefits or because they are unaware of what these organisations do with the data. The truth is that people's activities, in both the physical and virtual worlds, are captured by monitoring technologies, and these activities have been reduced to digital data (Johnson et al., 2017). Clearly, the data collected by the monitoring technologies have enhanced the performances of different sectors, such as logistics, retailing, and security, and have shaped their social interaction and profit-making (Porter and Heppelmann, 2014).

While monitoring procedures are now common, little analytical attention has been given to monitoring as a concept. Despite the prevalence of monitoring practices across many fields, few theoretical tools are available to comprehend and record these practices correctly. Both (Wang and Tucker, 2023) and (Johnson et al., 2017) highlighted the scant analytical attention paid to the idea of monitoring in general. Exploring the nature of monitoring made possible by software, the authors stress the need to investigate the nature of monitoring, the fundamental structure of monitoring processes, and the classification of monitoring systems. Figure 1.1 illustrates the structure of a monitoring system as described in (Johnson et al., 2017). It is designed to combine abstract data and allows monitoring and intervention to employ different kinds of technology and practice, making it easy to monitor the activities of people online.

Furthermore, it becomes difficult to address concerns about surveillance, privacy, and the ethical ramifications of monitoring without a comprehensive knowledge of these factors. (Johnson et al., 2017) claimed that monitoring practises are characterised as the systematic collection of data regarding the behaviours of people and objects. They use this simple notion to make an abstract

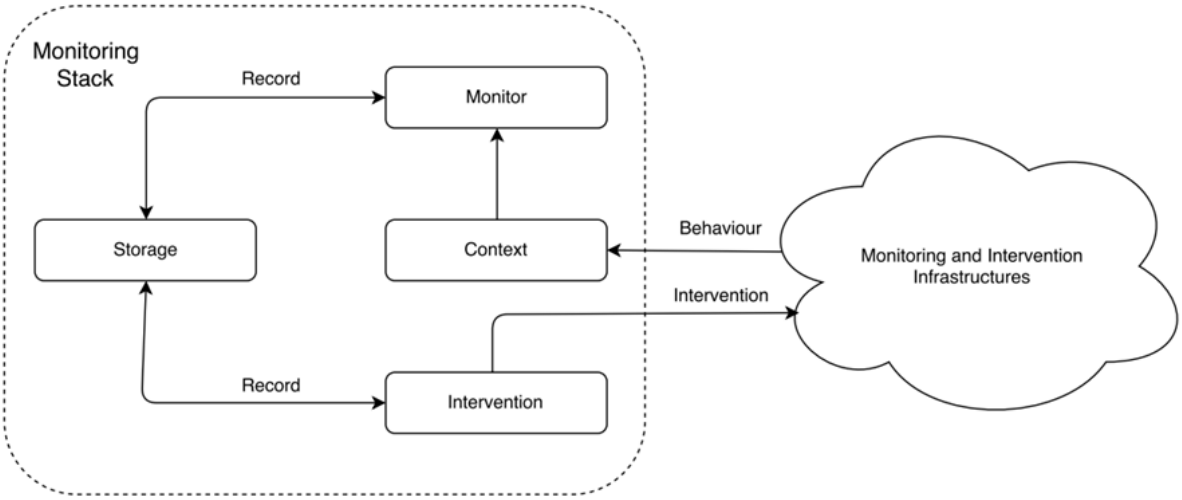

Figure 1.1: Stack architecture of monitoring and intervention (Johnson et al., 2017)

definition of monitoring that can function as a theoretical tool for examining diverse monitoring implementations. Building on this work, (Wang and Tucker, 2023) developed this idea further by outlining five different forms of monitoring, offering systems for identifying events needing interventions and behaviour changes.

These authors' contributions to the area of monitoring research and their desire for a thorough theoretical framework are the starting point for this study. Thus, the problem description is in line with their observations and underlines the importance of filling up knowledge gaps regarding monitoring procedures and their consequences for privacy, surveillance, and ethical issues.

1.3 Aim and Objectives

The first aim of this research is to extend a formal theory of monitoring, with an enhanced account of responses to monitoring data, using flags, messages, and activating interventions. The second aim is to test the new models in some security scenarios, specifically access control. More concretely, the objectives include the following:

1. To explore the commonalities and shared characteristics of diverse monitoring systems and to theorise monitoring practices, and to examine the role of monitoring in surveillance and privacy.

2. To develop new general mathematical models for monitoring processes that capture the essence of observing the behaviour of individuals and objects within a given context in which time is important.

3. To model access control methods for digital services, emphasising digital identities and protocols as integral components of monitoring systems.

In Johnson et al. (2017) certain basic concepts that characterised abstractly a monitoring system were defined and formalised mathematically using the basic methods of data type theory. Our

Objective 2 is to strengthen the conceptual discussion and models by revisiting certain concepts that are mentioned but not analysed in Johnson et al. (2017) (e.g., notification). In addition, this objective is to extend the mathematical model with real-world time as represented by timestamps. The original model was not able to adequately treat time-dependent monitoring systems. Our Objective 3 is to apply our new models to security contexts based on access controls where real-world time plays an essential role.

1.4 Significance of the Study

This study adds to the understanding of the concept of monitoring by developing further a framework for comparison and analysis, which makes it easier to recognize and examine monitoring procedures by finding similar traits and commonalities.

The universal themes of security and safety are entirely dependent on monitoring. Hence, this study adds to the development of the framework by establishing its applicability to simple canonical security scenarios focusing on identity, namely access control.

Furthermore, we consider issues brought up by widespread monitoring and surveillance, commenting on the effects of monitoring practices on people and society at large surrounding privacy and surveillance. Knowledge of monitoring can help guide policy discussions and debates on privacy protection and the moral use of surveillance technology.

1.5 Thesis Structure

This thesis consists of seven chapters, each progressively building upon the previous to develop a comprehensive understanding of monitoring systems, their algebraic models, and applications in access control. Below is an overview of the thesis structure:

- **Chapter 2: Monitoring and Surveillance**
  This chapter establishes the foundational concepts of monitoring and surveillance, distinguishing between the two and examining their evolution across different fields. It also discusses ethical and privacy concerns related to surveillance and introduces logs and logjects as essential elements in modern monitoring.

- **Chapter 3: Theoretical and Algebraic Models of Monitoring**
  This chapter presents the theoretical foundation for monitoring models, focusing on algebraic formulations. It introduces key concepts such as contexts, entities, observations, judgments, storage mechanisms, and intervention processes.

- **Chapter 4: Advanced Models: Timestamps and Notifications**
  Building on the previous chapter, which began with introducing the basic model of a monitoring system, here the model is extended by incorporating the new notions of time, timestamps and notifications. It explains how time influences monitoring, the role of timestamps in data storage, and the use of notifications for automated interventions.

- **Chapter 5: Virtual Access Control and Monitoring Systems**
  This chapter applies the newly developed monitoring framework to virtual access control scenarios. It models login processes, password authentication, user identity registration, and interventions triggered by failed login attempts. The role of notifications in managing access control events is also discussed.

- **Chapter 6: Physical Access Control and Monitoring Systems**
  This chapter examines physical access control, focusing on digital locks, biometric verification, and time-based access policies. It formulates a model for building access control and discusses security implications in real-world applications.

- **Chapter 7: Conclusion and Future Work**
  The final chapter summarizes the key contributions of the thesis, highlighting theoretical and practical implications. It discusses the limitations of the study and proposes directions for future research in monitoring systems and access control.

Chapter 2

Examples of Monitoring in Practice

2.1 Introduction

This chapter will examine two critical aspects: monitoring the primary source of data, the structure of monitoring systems, and the typologies of monitoring processes. The first aspect will consider where the data comes from in contemporary life and the relevance of monitoring in contemporary concerns about surveillance. The second aspect will provide examples of monitoring structures related to science, manufacturing, environment, finance, management, surveillance, and the Internet of Things.
This chapter is structured into seven sections. Section 2.1 deals with the introduction, which lays the foundation for the subject matter. Section 2.2 will contain the historical background of monitoring. Section 2.3 deals with monitoring in organisations, which enables organisations to achieve their aims and helps improve their performances significantly. It shows why monitoring is perceived as the primary data source and how monitoring plays a significant role in surveillance. Section 2.4 is about monitoring in homes, which is essential for collecting the data needed to enhance health outcomes and children's school performance. Section 2.5 is about logs and logjects. It will help to see how monitoring enhances the development of modern software and operations, which in turn provides a full understanding of how systems behave under production using logs and logjects, which are objects that record data. Section 2.6 is about surveillance and identity. It will reveal the role of monitoring in the information society. Section 2.7 is about domestic IoT, which will show how a network of connection cables connecting to the internet can enhance communication between digital devices and sensors.

2.2 Historical Background of Monitoring

2.2.1 Defining Monitoring and Surveillance

Monitoring refers to the systematic process of collecting, analyzing, and storing data related to the behavior of systems, individuals, or objects (Johnson et al., 2017). It serves various functions, including operational efficiency, security, and regulatory compliance. For example, in healthcare, patient monitoring systems continuously track vital signs to detect abnormalities, while in environmental science, air quality monitoring ensures compliance with pollution stand-

ards. It serves various functions, including ensuring operational efficiency, security, compliance with regulations, and performance optimization. For example, monitoring is commonly used in industrial processes to oversee production systems, in healthcare to track patient vitals, and in financial institutions to detect fraudulent transactions.

Surveillance, on the other hand, is a more targeted form of observation that typically involves the continuous or systematic collection of data on individuals or groups (Lyon, 2007). While monitoring can be neutral or beneficial, surveillance often carries implications related to privacy, control, and power dynamics. Examples of surveillance include CCTV cameras used in public spaces, workplace monitoring of employees, and data tracking by social media platforms.

2.2.2 Evolution of Monitoring in Different Fields

Efficient monitoring is necessary for the proper running of advanced technical systems, including water, gas, oil, power, transportation, and information technology infrastructures (Shabalov et al., 2021). Monitoring is essential for observing these systems to assess their progress and ascertain whether they are achieving their operational objectives. In 1868, Clerk Maxwell analyzed the steam engine governor, marking the inception of the control systems theory. This theory has since led to remarkable advancements in engineering, particularly in the development of control and regulatory systems for missiles, vehicles, and airplanes (Phuyal et al., 2020). The advent of control systems has led to the creation of sensors and processors that gather and analyze data, serving as the foundation for monitoring in the fields of science and engineering. Monitoring is a crucial element of science and engineering, particularly in fields that involve the use of laboratory equipment (James G. Speight, 2011).

Monitoring is not limited to science and engineering; it is also useful in other fields, including financial and social sciences. In the field of finance and accounting, monitoring plays a crucial role and has evolved ideas and methodologies to effectively oversee financial services, persons, and risks (Greenwood and Tao, 2020), (Kuzmenko et al., 2023). Richard Price (1723 - 1791) encountered challenges while utilizing inadequate demographic data and probability theories to advance the field of actuarial science throughout the 1770s. Nevertheless, individuals engage in the provision of financial services through the collection and analysis of social and economic data (Debnath and Basu, 2015), (Ozili, 2018). The incorporation of social and economic data has played a significant role in the collection, analysis, and advancement of data for the population census, mathematical models, and computational techniques and devices (Nations, 2017), (Zheng and Liu, 2022). Charles Babbage (1791 - 1871) was one of the early adopters who made use of social and economic data (Lewis, 2007).

Other developments, such as the expansion of banking, insurance, and accounting services for the general public, led to the establishment of big offices, which in turn led to the need for large-scale data processing (OECD, 2017). The presence of these offices necessitates the development of novel architectural structures, which in turn establishes a connection between data and data processing and the field of architecture (Liu et al., 2022). Hence, the various types of data obtained in such offices are closely monitored to ensure that they are properly preserved and utilized for their intended purpose.

2.2.3 The Role of Surveillance in Society

Surveillance is a notion strongly linked to monitoring, which involves the ongoing observation of individuals and their actions (Alexander, 2008). The essay Samuelsson et al. (2023) discusses the pervasive surveillance culture that impacts all aspects of human existence, including attitudes, motives, and behaviors. According to (Lyon, 2017), surveillance culture is now primarily shaped by digital technologies, which are undergoing quick and influential changes. The prevalence of surveillance culture enhances social regulation and governs individuals' contemplation on the nature of things and their impact on daily routines. For instance, social media companies monitor user activity to provide personalized advertisements, while government agencies implement surveillance systems to track and prevent criminal activities. Additionally, retail stores use facial recognition to detect shoplifters, while smart city initiatives employ surveillance networks to regulate traffic and enhance public safety. Hence, it is imperative to give due consideration to monitoring and surveillance in information systems, as this will effectively tackle societal concerns associated with the utilization of information and communication technologies (Olan et al., 2022).

The notion of the information society is relevant to the act of monitoring. As (Nath, 2009) stated, the information society effectively utilizes information technology in innovative and efficient ways to fulfill globally competitive requirements. Therefore, it is a culture that extensively employs the knowledge and information in computer technology (Agbo, 2015). With each passing day, the world is increasingly transitioning into a digital realm, transforming the global community into an information society. Given the growing need for data in today's society, it is imperative to closely supervise its usage to verify that it aligns with its intended purpose. People are increasingly worried about the methods used to gather data and the surveillance of their personal and work lives (Kapoor et al., 2018). Data is gathered comprehensively from several facets of human existence, encompassing people, group members, organizations or societies, workplaces, apps, and numerous other areas of daily human life (Kapoor et al., 2018).

Furthermore, software and computer programs are employed to oversee many activities, including collecting and monitoring data in management systems, workplace operations, internet purchasing, and healthcare settings (Dash et al., 2019). Technological tools are utilized for monitoring residential areas, public spaces, workplaces, and online platforms to enhance the level of security experienced by persons (Cascio and Montealegre, 2016). These instances demonstrate a significant and rapid growth in the global need for monitoring.

2.3 Monitoring in Organization

Monitoring is very essential in organisations, as it enables the organisation to achieve its aims and helps improve performances, usually significantly. For over a century, companies and organisations have met their markets' competitive demands by monitoring the performance of their workers. According to da Silva and Borsato (2017), many organisations use key performance indicators (KPIs) to monitor the performance of their staff to help them know the performance level at which the organisation or company expects them to achieve. Companies need to have a relevant data collection process that matches the indicators to ensure effective monitoring of the

organisation da Silva and Borsato (2017). According to Almulaiki (2023), assessment and monitoring are significant for individuals and organisations to effectively enhance their performance effectively. In organisations, the management is responsible for monitoring the performance of its staff, which is usually carried out using performance assessment tools. The study by Almulaiki (2023) identified that the use of performance appraisal is beneficial to improving employee performance compared with performance planning.

Considering schools as an organisation, Vianney et al. (2020) used a descriptive survey to determine the impact of monitoring and evaluation on the performance of primary school pupils in their school activities and in English Language courses. The study's findings showed a significant positive correlation between project monitoring and evaluation of school activities and academic performance in English (Vianney et al., 2020). Therefore, the above study shows that monitoring in the educational organisation is significant to understanding the progress of activities and determining whether the scope and objectives are achieved.

Therefore, da Silva and Borsato (2017), Vianney et al. (2020) and Almulaiki (2023) studies indicated that monitoring plays a significant role in organisations in collecting data that will be relevant in determining the performances of the organisations.

2.4 Monitoring in Home

Monitoring in the home has been identified to be essential in collecting data that will help improve health outcomes and improve the performance of those children in the schools and affect their behaviours. A report by Mathew et al. (2023) showed the beneficial role of monitoring in managing diabetic patients. The report indicated that blood glucose monitoring is beneficial to understanding the pattern of the blood glucose level of the patients by providing and recording data, which will help determine the effect of the diet intake on the patient's blood glucose. Also, monitoring is vital to understanding the impact of exercise, drug intake, and the health conditions of the individual who suffers from diabetes mellitus (Mathew et al., 2023). Therefore, monitoring plays an integral role in managing data in homes, as it helps determine the data that would be used to assess the patient's health condition and know the next approach to take to ensure that the individual is properly managed.

The study by Naite (2021) provided relevant information that indicates the relevance of monitoring of children by their parents in their homes contributes to their academic performances. The study findings showed that children who were adequately cared for in their homes had significant and better performance in their academics compared to those who were not adequately cared for in their homes (Naite, 2021). Therefore, the study suggested that parental control and guidance were relevant in improving academic performance. Thus, it was suggested that parents should be significantly involved in their children's lives at home, as that will help improve their performance significantly.

Laird et al. (2018) study showed that monitoring in homes does not affect the health and well-being of adolescent children in their homes. The study aimed to discover whether there are negative consequences of increased monitoring by parents on the well-being of their children.

The findings of this study showed that less monitoring was related to the negative behaviours of the adolescents. However, increased monitoring improved the adolescent's social behaviour, as well as have no relation to mental or psychological well-being, such as depressive symptoms (Laird et al., 2018).

Therefore, the three above articles conducted by Mathew et al. (2023), Naite (2021) and Laird et al. (2018) showed that monitoring at home is beneficial in collecting data that would be helpful in improving health condition, improving the performance of children in schools and have a positive influence on the behaviour of those children.

2.5 Logs and Logjects

Monitoring helps the development of modern software and operations to understand the behaviour of systems under production by generating data called logs Cândido et al. (2021). A log in software engineering refers to a part of the primary data source in log files that are observable by a network (Cândido et al., 2021). A log file is computer-generated data showing information concerning the usage patterns, activities, and operations in a particular device, such as an operating system, application or server (Sarker, 2021). Logs can come in various types, such as Proxy logs, Application logs, Endpoint logs, Windows event logs, Perimeter device logs and IoT logs (ManageEngine, 2023). So, the computer-generated information (log) occurs due to the presence of monitoring in the devices.

A logject is an object that records data about itself; typical examples are objects containing processes and software that generate logs. For example, network servers, smartphones, web applications, and security cameras, among others (**?**). Logjects are naturally monitoring their behaviour, as they have been described to be devices that have "self-awareness", and the concept was introduced by Rob Dodge and Martin Kitchin (Dodge and Kitchin, 2009). Logjects are created either because of a desire to observe an entity's performance or by deliberately adding software features to an object, which then creates logs to record and store them for future purposes automatically (El-Masri et al., 2020).

Logjects are significant tools in the modern age, as they allow for the storage of everyday data humans need to perform their daily activities (El-Masri et al., 2020). The help of logjects is beneficial to store phone contacts of individuals on mobile devices, such as Android phones and tablets, thus; easing the stress of memorising phone numbers by humans. Just as Logs discussed above have different types, logjects have two major types: permeable logjects and networked logjects. Therefore, logs and logjects are essential tools in monitoring, which help collect data that can be used to understand the behaviour of systems.

2.6 Surveillance and Identity

Surveillance and identity play a significant role in monitoring, and understanding the concepts is integral for the topic under consideration, which is monitoring in the information society. An ordinary definition of surveillance is that it is a continuous monitoring process that involves the monitoring of behaviours of individuals and objects virtually or in reality (Wang and Tucker,

2017). Several technologies nowadays adopt surveillance in their systems that record information about humans' activities in their physical and virtual environments (Yamin, 2019). Some surveillance system involves using digital cameras to monitor our environment to ensure the safety and secure individuals and their properties (Wang and Tucker, 2017). Also, surveillance software on the internet collects data from individuals using internet-enabled devices. When surfing the internet, such information may be used for different purposes. However, the data collected through surveillance systems may invade individuals' privacy or freedom (Wang and Tucker, 2017). Individuals or objects are regarded as entities in software engineering with regard to surveillance systems. Therefore, a surveillance system observes behaviours and identifies entities with their attributes. Thus, it comprises four major methods: entity, observable behaviour, attribute and identity. However, the major focus in this section is on surveillance and identity.

As surveillance has been defined, identity, which is a component of surveillance, refers to a method that labels an entity after recognising the entity (Wang and Tucker, 2017). To illustrate, a string that specifically labels a tweet made on Twitter is referred to as an identifier. Therefore, surveillance and identity are particular aspects that could benefit the understanding of monitoring in the information society.

2.7 Domestic IOT

The concept of domestic IoT is relevant to the topic under consideration. IoT refers to the Internet of Things, which is a particular network capable of connecting anything with the Internet that facilitates communication between digital devices and sensors that help ease individuals' lives (Kumar et al., 2019). The aim of IoT is to facilitate the lives of individuals by ensuring absolute simplicity in different spheres of life, whether at home, at work, in industries and others, which help improve the quality of life (Nižetić et al., 2020). Domestically, IoT has been beneficial in facilitating lives by ensuring smart homes, pollution control, energy saving and many others (Kumar et al., 2019). Therefore, the world has become an information society, and the data obtained through the IoT makes considering domestic IoT important as an aspect of monitoring in the information society.

Chapter 3

General Algebraic Models of Monitoring

3.1 Introduction

This chapter is structured into five sections. Section 3.2 presents the fundamental principles of monitoring, offering a conceptual understanding of how entities, characteristics, and behaviors are observed and analyzed. Section 3.3 delves into the theoretical foundations of monitoring systems, discussing the formal definitions and mathematical models that underpin monitoring frameworks. Section 3.4 explores different types of monitoring, including individual, interval-based, and instant monitoring, with a focus on their applications and implications. This section 3.4 also examines the role of storage in monitoring systems, addressing how records are collected, retained, and processed to facilitate analysis and decision-making. Finally, Section 3.5 discusses intervention mechanisms, detailing how monitoring data informs actions and decisions through different intervention strategies, including basic interventions, time-dependent interventions, and user-involved interventions. This structured approach provides a comprehensive understanding of monitoring as a fundamental concept in data observation and analysis.

3.2 What is Monitoring?

According to Johnson et al. (2017), monitoring is a process of observing the properties of people and object behaviours in a context. Context encompasses entities and their characteristics and behaviours. Thus, monitoring a context starts by choosing the needed attributes for observation. The contexts are defined as representations of abstract data and observation attributes by logical languages. Thus, Johnson et al. (2017) confine their definition of monitoring to collecting, evaluating and recording observational data; the result of a monitoring system is basically a record.

The following definition is taken from Johnson et al. (2017) and Wang and Tucker (2023). This informal definition is the basis of the formal model we study in this thesis.

Definition: Abstractly, a monitoring system comprises the following components and methods:

1. Entity: The concept of an entity refers to individuals, objects, or physical phenomena that show behaviour in a particular location and at a particular period.

2. Identity: The concept of identity refers to the monitoring method by which the data generated identifies an entity in a particular location and time.

3. Observable behaviour: This involves a method used in generating data representing the behaviour of entities at a particular location in a given period of time.

4. Attributes: Attributes are used to describe and recognise behavioural properties.

5. Processing: Processing refers to the storing and analysing properties of the state of behaviours of the individuals and objects.

6. Recording and reviewing: The aspects of recording and reviewing in monitoring refer to data storing and displaying the properties and outputs for further review.

Therefore, monitoring is important in this contemporary society as the number of datasets is significantly transforming social and economic lives (Tuomi, 2018). Hence, investigating monitoring in various aspects of human life and determining the role of monitoring in the surveillance system are needed.

3.3 The concept of monitoring

Monitoring is a process that observes the behaviour of people and objects in a specific scenario. Monitoring involves choosing data to represent entities and behaviours, properties to observe by testing the data and a form of storage to record the results. In general, monitoring is confined to the collection, evaluation and recording of observational data regarding the behaviour of entities.

A *context* consists of

- entities
- characteristics, i.e., certain information about the entities
- behaviours

Monitoring a context begins by choosing specific attributes of the data to be observed and a means of making judgements about the attributes. We informally define *Monitoring infrastructure* to be the technological and human systems that obtain and send the data representing the behaviour of entities to a monitoring system; and *Intervention infrastructure* to be the systems that receive the information from a monitoring system with interventions and initiate various responses and actions on the entities.

Here are the primary components of any monitoring system as proposed in Johnson et al. (2017):

Context: Monitoring occurs in a context. A context is made up of entities. The entities have associated or contain characteristics which give information that is pertinent to the entities inside

the context. In addition, entities have behaviours which may be observed. The characteristics of an entity can determine the behaviour of the entity. Thus, characteristics are a parameter pertaining to an entity's behaviour.

Observation: Behaviours possess attributes which may be observed. During observation, a behaviour of an entity is tested in order to decide whether an attribute exists. The outcome is evaluated, which results in a judgement. Usually an attribute is appraised using a series of values on a scale. The appraisal will be a labelling, grading, measure, or probability. An instance could include metaphorically labelled bands, giving a qualitative evaluation as a judgement, such as the following well-known three-valued traffic light descriptors:

$$\{\textit{green, amber, red}\}.$$

Another instance is that judgements based on bands of numerical values are the outcomes of physical measurements with error margins. In social surveys and questionnaires, five-valued assessment are employed, namely:

$$\{\textit{strongly agree}, \textit{agree}, \textit{neutral}, \textit{disagree}, \text{and } \textit{strongly disagree}\}.$$

Monitoring and Records: The primary function of monitoring is to produce an observation, and a record relevant to this observation. A record must encompass the following: the details of the entity, the characteristics relevant to the entity, the observed attributes, as well as the judgements.

Interventions: In order to utilize the monitoring data, the records must comprise specific features that should be communicated to the *intervention infrastructures*. These infrastructures are external to the monitoring system. The communications with external infrastructures are referred to as *notifications*. The notification could trigger various virtual or physical activities that could alter the characteristics of an entity as well as its behaviour.

Judgements influence interventions: *Trigger conditions* detect the observations that necessitate actions. *Judgement* is the input of a trigger. Its outcome is a Boolean value which could, in turn, trigger a notification for action. In other words, this action should only be taken if there is a trigger; otherwise, no action should be taken. The triggers determine the line of action to be taken. Actions alter the information regarding the characteristic of an entity.
The combination of all the above perspectives constitutes the underlying informal conceptual framework for monitoring which can be formalised using multiple technical approaches and schemes. There are diverse logics in formalising judgements as well as attributes, diverse semantic modelling for behaviour, and diverse computational models in order to analyse monitoring and interventions.

3.4 A Basic Model of Monitoring

The following is a detailed explanation of the key theoretical elements of the basic model of monitoring created in Johnson et al. (2017).

3.4.1 Time

The selection of a model of time T is crucial and we will refer to it as the *monitoring clock*. To start we choose the time to be discrete and set T as follows:

$$T = \{0, 1, 2, ...\}. \tag{3.1}$$

Some algebraic structures will be added later to our models of time.

3.4.2 Behaviour and Contexts

Behaviour: Behaviour is defined by a number of types of data - usually video, audio, images, text or quantitative measurements. The streams are perceived as sequences in *discrete time* which are infinite and well-defined.
Hence, the data *stream* is described as a *total function* $a : T \to A$ mapping time points in T to data in A. The set $[T \to A]$ of all streams describes the space of all behaviours.
We assume that the behaviour corresponding to entities occurs based on time. In this thesis, an entity's behaviour is modelled over time by a data *stream* from A as shown:

$$a(0), a(1), a(2), ..., a(t), ... \in A \,\text{for}\, t \in T. \tag{3.2}$$

where T characterizes a set of data which represents points in time and A is a set of elements. There exist various forms of stream behaviour, as both time and data can be viewed as continuous or discrete. Besides, they both can be arranged through topologies as well as orderings.

Entities and Characteristics: Let E be the set of entities and C be the set of characteristics. We describe the *behaviour* relevant to an entity as a data stream engendered by the behaviour map

$$[[-, -]] : E \times C \to [T \to A] \tag{3.3}$$

so that for entity $e \in E$, with characteristics $\chi \in C$, at time $t \in T$,

$$[[e, \chi]](t) = \text{data for behaviour of an entity } e \text{ with characteristics } \chi \text{ at time } t.$$

3.4.3 Observation

Attributes and Judgements: Behaviours own attributes that may be observed over time. Let $Attr$ be the set of behavioural attributes, and J be the set of judgements; usually, J has a finite number of elements.
Despite the fact that there are several forms that an observation can take, the action of observing an entity's behaviour, and making an assessment, begins with a map

$$Obs_0 : Attr \times [T \to A] \to J. \tag{3.4}$$

There are two types of observations, which we will define below: individual observation and observation over intervals and at instants.

Individual Observation: Assume an attribute of interest varies in accordance with the entity as well as its characteristics and its choice is defined as follows:

$$P : E \times C \to Attr. \tag{3.5}$$

Then, when we substitute into mapping 3.4, the observation map will become:

$$Obs : E \times C \times [T \to A] \to J. \tag{3.6}$$

which, for a given entity e with characteristics χ, calculates the degree that $P(e, \chi)$ is a property of $a \in [T \to A]$:

$$Obs(e, \chi, a) = Obs_0(P(e, \chi), a). \tag{3.7}$$

Interval Observation (over $[t_1, t_2]$): Most likely, time is involved in attributes as behaviour is based on time. Firstly, we observe the behaviour over an interval $[t_1, t_2] \subset T$. Hence, this leads to further amendments of mapping 3.4 as follows:

$$Obs_0 : Attr \times [T \to A] \times T \times T \to J. \tag{3.8}$$

This function evaluates the history or behavior of the system over an interval.

Path Observation ([0,t]): The interval $[0, t]$ is selected in order to perceive the entity's entire history up to time t. It captures changes and behavior across time, considering all behaviors between 0 and t.

Instant Observation (at t): The interval $[t_1, t_2]$ is utilized as $[t, t]$ in order to perceive behavior at an instance of time t, which looks only at a single time point and returns the behavior of the system at that specific instant.

$$Obs_0 : Attr \times [T \to A] \times T \to J. \tag{3.9}$$

In cases that depend upon time, the attribute to be observed can be dependent on the entity and its characteristics (as stated earlier with the choice function P), and also on the inspection time. Thus, the choice function can have the form:

$$P : E \times C \times T \times T \to Attr. \tag{3.10}$$

3.4.4 Monitoring

This thesis will implement monitoring by letting the set R of records be

$$R = E \times C \times Attr \times J. \tag{3.11}$$

There are three major types of monitoring which are individual monitoring and monitoring over intervals and at instants. And the idea of monitoring can be expressed as the following simple

mapping

$$Monitor : E \times C \to R. \tag{3.12}$$

Individual Monitoring: Suppose that entity e and characteristic χ determine the attribute to be observed, then the attribute to be checked will be $P(e, \chi)$.
When you apply e as an entity, χ as the characteristics, then $P(e, \chi)$ is the attribute to be checked and it is defined by:

$$Monitor(e, \chi) = (e, \chi, P(e, \chi), Obs(e, \chi, [[e, \chi]])). \tag{3.13}$$

Monitoring over intervals and at instants: In this type of monitoring, the monitoring is performed within a particular time interval. The suggested monitoring mapping for time interval t_1 and t_2 using mappings 3.3 and 3.8 will become:

$$Monitor : E \times C \times T \times T \to R. \tag{3.14}$$

Therefore, the monitoring is defined by

$$Monitor(e, \chi, t_1, t_2) = (e, \chi, P(e, \chi), Obs(e, \chi, [[e, \chi]], t_1, t_2)). \tag{3.15}$$

Monitoring stream: When monitoring is done at a specific time t then the $t_1 = t_2 = t$, we assume that the observation starts and ends at the same moment. However, because the function now produces results over time, it effectively forms a stream of records as follows:

$$monitor : E \times C \to [T \to R]. \tag{3.16}$$

is defined by

$$monitor(e, \chi) = (e, \chi, P(e, \chi), Obs(e, \chi, [[e, \chi]])). \tag{3.17}$$

3.4.5 Storage

The purpose of the storage element is collecting, as well as retaining the entire, or some aspects of the records generated by monitoring. Consequently, we suppose that, at any given time, the storage involves a finite list of records:

$$r_0, r_1, ..., r_n \in R \tag{3.18}$$

For ease and shortness, we will often assume that the entire records produced by the monitoring are stored. The index n denotes the time calculated by the monitoring clock. The output relevant to the monitoring element together with the input of the storage element is a stream described by map 3.16. A storage element can be designed in several ways, including the database theory. In this thesis, we will assume that there is an abstract data type for storage, which includes functionalities like input, arrange, and output the records, as well as support programs capable of analysing the stored data.

3.5 Interventions

Intervention, in the context of monitoring, refers to the actions or measures taken based on the information gathered from the monitoring process. These interventions can involve external systems and activities that may influence or alter the behavior or characteristics of the entities being monitored.

An intervention may or may not inform an entity via a notification. We consider different types of intervention, such as basic intervention, intervention with notifications, intervention depending on time, and intervention with notification and user action.

In this section, we present the algebraic model for basic interventions as introduced by Johnson et al. (2017), and other types of intervention: intervention with notifications, intervention depending on time, and intervention with notification and user action. We will explain each type by defining them, providing examples, and discussing their complexity and scope. The choice of intervention type depends on specific requirements including security policies, desired outcomes, goals of the monitoring system, and the entities being monitored.

3.5.1 Basic Intervention

Definition: Basic intervention involves taking actions based on the monitoring data without incorporating notifications or time dependencies.

Example: If the monitoring system detects a security breach in a computer network, a basic intervention could involve automatically blocking the suspicious IP address to prevent further access.

Consideration: Basic interventions are suitable for some straightforward security responses without the need for notifications or user interaction.

Intervention for streams: By the conceptual framework, interventions are derived from judgments and are not directly related to behaviors; they are independent of the streams.

Triggers: As an input, trigger conditions take a judgement value $j \in J$ that comes from the observations made by the function Obs and sends out a truth value $tc(j) \in \mathbb{B}$ as an output. Therefore, we use $Trig = [J \to \mathbb{B}]$ to display the set of all trigger functions.

Actions: An update $act(\chi)$ is executed on the information $\chi \in C$ by an action function $act : C \to C$. Let Act be the set of all functions that can be executed information in C.

Interventions: We use *triggers* and *action* functions to specify the intervention that occurs as a result of an entity's behaviour being observed. Using both of these functions, we define an intervention $tc \to act$, where $(tc, act) \in Intv = Trig \times Act$; $tc \to act$ donates the idea when tc

returns t on a judgement the operator act is applied to C. Thus:

$$Int : R \times Intv \rightarrow E \times C, \tag{3.19}$$

defined as,

$$Int((e, \chi, P(e, \chi), j), tc \rightarrow act) = \left\{ \begin{array}{l} (e, act(\chi)\ if\ tc(j) \\ (e, act(\chi)\ if\ \neg\ tc(j). \end{array} \right\} \tag{3.20}$$

3.5.2 Other Interventions

We now briefly introduce the three other types of intervention which built on basic intervention.

1 - Intervention with Notifications
Definition: This type of monitoring system not only triggers interventions but also sends notifications to external systems or users to inform them about the detected event.
Example: In a smart home security system, if unauthorized access is detected, the system could both lock the doors and send a notification to the homeowner's mobile device.
Consideration: Interventions with notifications provide a way to keep relevant stakeholders informed and enable them to take further actions if needed.

We will see this type of intervention in Chapter 5 which will explain and model how intervention with notifications works in the login scenario.

2 - Intervention Depends on Time
Definition: This type of intervention considers the timing of events in the monitoring process. Different interventions may be triggered based on specific time intervals or patterns.
Example: A manufacturing plant monitoring system may schedule routine maintenance tasks during non-production hours to minimize disruption.
Consideration: Time-dependent interventions are useful for scenarios where security policies vary based on the time of day.

We will see this type of intervention in Chapter 6 which will explain and model how intervention depending on time works in the building access control scenario.

3 - Intervention with Notification and User Action
Definition: This involves notifying users and, in addition, allowing users to take actions or make decisions based on the received notifications.
Example: In a health monitoring application, if abnormal vital signs are detected, the system may notify both the user and emergency services. The user, upon receiving the notification, can choose to call for help or dismiss the alert if it's a false alarm.
Consideration: In situations where human judgment or user input is valuable, interventions with user action allow for more flexible and user-involved responses.

This type will not be seen in the models of Chapter 5 and Chapter 6 but it can be considered as an enhancement of the models.

In conclusion, interventions can vary in complexity and scope. Basic interventions act without communication, while interventions with notifications involve informing entities. Time-dependent interventions consider the temporal aspect of monitoring data, and interventions with user action allow users to respond to notifications actively. The choice of intervention type depends on the the level of automation desired, the criticality of security events, and the level of user involvement in decision-making, a combination of these intervention types may be implemented to create a comprehensive and adaptive security system.

3.6 Summary

This chapter began by defining monitoring and discussing key related concepts, including context, observation, monitoring records, interventions, and judgments that influence interventions. It then introduced the fundamental principles that form the basis of the mathematical model of monitoring, as presented in Johnson et al. (2017). These principles include streams, monitoring streams, time, behavior, different types of observations, types of monitoring, and types of interventions. Additionally, the chapter explored the role of storage in retaining monitoring records. While this model provides a structured approach to monitoring, it assumes a discrete-time framework and abstracts certain real-world complexities. Future research may focus on refining the model to accommodate real-time processing, dynamic behavioral changes, and more adaptive intervention strategies.

Chapter 4

General Algebraic Models of Monitoring with Timestamps and Notifications

4.1 Introduction

In computing systems, time is usually measured using a system clock(Mohay, 2003). Therefore, time can be viewed as the clock that counts an event and describes the input and output of the event. On the other hand, a timestamp refers to the sequence of characters and encoded information that identifies the time a certain event has occurred, mostly in terms of the date and time (Mohay, 2003). In some computing systems, the accuracy of the time in a timestamp can be given by a small fraction of a second.
A timestamp can also be defined as the digital information on date and time that is usually attached to digital data. Timestamps are important in recording and monitoring data in computer systems (Mohay, 2003). Precisely, timestamps provide detailed information on when a particular event has occurred, For example, the time a digital file was created or last modified, the time the file was accessed, as well as the most recent status change of the file (Chen and Hong, 2004).
This section 4.1 explores the role of time and timestamps in computing systems, the different timestamp formats, the structure of timestamps in an algebraic model, as well as how computer systems generate timestamps. Then section 4.2 revise the basic model of monitoring by introducing timestamps as an important new component of the general model. In section 4.3, we introduce the notion of notifications by a monitoring system, and, in section 4.4, we revise one more time the general model to include notifications. In both cases, timestamps and notifications, we survey their various forms and roles in computing systems to motivate and guide their formalisation.

4.1.1 Time

Time in computer systems is measured using a system clock. The clock is basically a microchip whose function is to regulate the timing and speed of the various functions of the computer

(Mohay, 2003). The microchip usually consists of a crystal that vibrates at a given frequency upon the application of an electrical signal. The number of crystal vibrations per unit time – clock speed – determines the processor speed of the computer. For instance, a processor with a speed of 2 GHz means that the system clock can make two billion vibrations/cycles in one second. Because the system clock is a continuous pulse, it enables the computer to keep the correct time. The system clock keeps a record of the number of seconds elapsed since an arbitrary starting time called an epoch (Chen and Hong, 2004). The system clock then uses this information to accurately determine the actual date and time.
Maintaining the correct time is crucial in a computer system. First, time is deemed very important in the performance of scheduled tasks (Mohay, 2003). Failure of the system clock to keep track of the correct time can prevent the computer system from performing the right task at the right time. Second, time is also essential in creating timestamps for files (Mohay, 2003). The computer system employs the system clock to apply the timestamps on every created or modified file. Therefore, accurate system time ensures that the timestamps on the data files are reliable. Third, accurate system time is also essential in ensuring accurate and reliable software and access authorization (Mohay, 2003). Inaccurate system time can be very problematic for the use of software that relies on web-based services and web-based authorization schemes that require access to computer information.

So, the role of time in computing systems is to count the number of events occurring over time. In addition to the virtual clock T that enumerates events we now add real-world times for events using *timestamp*.

4.1.2 Timestamp

A timestamp defines the time a computer records an event, and not necessarily the time the event took place. Of course, in most cases, the difference between the actual time of the event and the time at which the event is recorded by timestamp is trivial. The timestamp that is usually entered into a log file should, as much as practically possible, be close to the time of the event that the log records (Mohay, 2003). Timestamps are used for logging events or a sequence of events. Therefore, the timestamps must utilise a consistent format to allow for comparison between records to enable easy tracking of progress over time. Since a great majority of operating systems support POSIX stat, timestamp usually involves the time of the last access to a file, the time the file was last modified, and the most recent change of file status (Chen and Hong, 2004).
However, it is worth noting that while timestamps are a standard way of storing data and are universal, there is no universal timestamp format. Different operating systems use different ways to store timestamps (Chen and Hong, 2004). For example, while Windows uses the ANSI standard for creating timestamps, other operating systems such as Linux may use a different standard. In Windows, timestamps are stored as the number of seconds since 1st January, 1601 while in Unix, timestamps are stored as the number of seconds since 1st January, 1970 (Mohay, 2003). Additionally, programming languages may store timestamps using methods that are different from those used in a database. Because of the diverse array of timestamp formats available,

modern programming languages and computer applications have built-in timestamp conversion functions (Chen and Hong, 2004). Timestamp functions are tasked with calculating times and manipulating time information to synchronize computer operations within a computing system.

We want to define a timestamp in the following format:

$$[YYYY : MM : DD : HH : MM : SS]. \tag{4.1}$$

e.g. Half past 3pm on the 7^{th} June 2022 would have the form:

$$[2022 : 06 : 07 : 15 : 30 : 00]. \tag{4.2}$$

In order to do so, first, we must define $Date$ and $Time$.
Let,

$$Year = \{0, 1, 2, ..., 9\}^4 \tag{4.3}$$

be the set of all possible combinations of 4 digits, representing the year.
Let,

$$Month = \{1, 2, ..., 12\} \tag{4.4}$$

where each number $n > 0$ represents the n^{th} month of the year.
Let,

$$Day = \{1, 2, ..., 31\} \tag{4.5}$$

where each number $n > 0$ represents the n^{th} day of the month.
This allows us to define the date to be:

$$Date \subset Year \times Month \times Day. \tag{4.6}$$

$Date$ is a proper subset because the number of days in the months vary, e.g., in the case of February there is no 30^{th} day. For Time, let

$$24Time = \mathbb{Z}_{24} \times \mathbb{Z}_{60} \times \mathbb{Z}_{60} \tag{4.7}$$

where $\mathbb{Z}_{24}$ gives the hour, and each copy of $\mathbb{Z}_{60}$ gives the minutes and seconds respectively.
Now, for our model we can define timestamp as follows:

$$Timestamp = Date \times 24Time. \tag{4.8}$$

Note that the set of timestamps is a finite set. Primarily this is because the set of dates is bounded by the year 9999. Thus, we might expect that the validity of our models using timestamps is limited to finite durations.

4.1.3 Timestamp in alternative formats

Although different timestamp formats are available, timestamps are usually stored as integers. The utilization of integers to store timestamps is very efficient since integers necessitate minimal storage space (Mohay, 2003). However, the integers are usually converted to legible time formats by the built-in timestamp functions when displayed. Apart from the difference in format, timestamps also have different specificities and resolutions (Chen and Hong, 2004). For instance, some timestamps have a resolution in the range of seconds, while others require milliseconds or even nanoseconds.

In order to express the timestamp in alternative formats, First, let the set $AmPmTime$ represents the 12-hour format for time, typically used in clock systems that distinguish between "AM" (Ante Meridiem) and "PM" (Post Meridiem). This set would contain tuples representing hours, minutes, and seconds in the 12-hour format. Specifically, the set $AmPmTime$ would be defined as:

$$AmPmTime = \{(h, m, s) \mid h \in \{1, 2, \ldots, 12\}, m \in \{0, 1, 2, \ldots, 59\}, s \in \{0, 1, 2, \ldots, 59\}\} \times \{AM, PM\} \tag{4.9}$$

where:

- h represents the hour, which can take values from 1 to 12 (for a 12-hour clock).
- m represents the minutes, which range from 0 to 59.
- s represents the seconds, also ranging from 0 to 59.

Now we can define the timestamp in alternative formats,

$$\phi : 24Time \to AmPmTime \tag{4.10}$$

be the function defined as follows:

$$\phi(h, m, s) = \begin{cases} (12, m, s) \text{ AM}, & \text{if } h = 0, \\ (h, m, s) \text{ AM}, & \text{if } 1 \leq h < 12, \\ (12, m, s) \text{ PM}, & \text{if } h = 12, \\ (h - 12, m, s) \text{ PM}, & \text{if } h > 12. \end{cases} \tag{4.11}$$

where $(h, m, s) \in 24Time$.

Next, let

$$\phi : Date \to Date_2 \tag{4.12}$$

The set $Date_2$ consists of tuples of the form $(d, \psi(m), y)$, where:
d is the day of the month, $\psi(m)$ is the abbreviation of the month, and y is the year.

Now, ϕ is defined by

$$\phi(y, m, d) = (d, \psi(m), y) \tag{4.13}$$

where

$$\psi(m) = \begin{cases} Jan & \text{if } m = 1 \\ Feb & \text{if } m = 2 \\ Mar & \text{if } m = 3 \\ April & \text{if } m = 4 \\ May & \text{if } m = 5 \\ Jun & \text{if } m = 6 \\ July & \text{if } m = 7 \\ Aug & \text{if } m = 8 \\ Sep & \text{if } m = 9 \\ Oct & \text{if } m = 10 \\ Nov & \text{if } m = 11 \\ Dec & \text{if } m = 12 \end{cases} \tag{4.14}$$

we can use ϕ and ψ to print the timestamp in a more human readable format. e.g. $[2022 : 06 : 07 : 15 : 30 : 00]$ becomes $3 : 30 : 00pm, 7Jun\ 2022$.

4.2 A Model of Monitoring with Timestamp

We began our study of the theory of monitoring in 3.3 by introducing the Conceptual Framework for Monitoring, namely *Context, Entities, Characteristics, Behaviour, Observation, Attributes, Judgements, Monitoring, Records, and Interventions.*

We utilized the aforementioned concepts in order to develop theoretical ideas about monitoring systems. In section 4.1, we demonstrated how to model the fundamental conceptual components of monitoring systems using algebraic models. In section 4.1, we introduced both time and timestamp and how they are important when monitoring occurs. We defined time, timestamp, and timestamp in different formats using the algebraic model.

We will extend our basic model of monitoring by adding the timestamp to the general model which will result in a new modeling of observation and record functions.
Recall the mappings 3.3 - 3.4, explained in detail in section 3.4, for an introduction to the monitoring streams.
Now, we will modify the definition of a monitoring observation (refer to mapping 3.5 in section 3.4) to include a timestamp. We defined timestamp using the set:

$$Timestamp = Date \times 24Time \tag{4.15}$$

which refers to the sequence of characters and encoded information which identify the time a certain event has occurred, mostly in terms of the date and time of a day.

Hence, the modified observation map can be expressed as such:

$$Obs_0 : Attr \times [T \to A] \to J \times Timestamp. \tag{4.16}$$

At this point we do not know if timestamps are being 'correctly' used: for example, we might seek correctness conditions based on the relationship between *timestamps* and virtual clock T.
Individual Observation: Recall mapping 3.4 in subsection 3.4.3. Upon substitution into mapping 4.16, the observation map will be:

$$Obs : E \times C \times [T \to A] \to J \times Timestamp \tag{4.17}$$

which, for a given entity e with characteristics χ, computes the degree that $P(e, \chi)$ is a property of $a \in [T \to A]$ and timestamps the judgement:

$$Obs(e, \chi, a) = Obs_0(P(e, \chi), a) \tag{4.18}$$

Observation over intervals and at instants: Recall mapping 3.8 in subsection 3.4.3 for more details. Upon insertion of timestamp, the corresponding observation map can be defined as follows:

$$Obs_0 : Attr \times [T \to A] \times T \times T \to J \times Timestamp. \tag{4.19}$$

When incorporating the timestamp into mapping 3.10, the observation map can be rewritten as such:

$$Obs_0 : Attr \times [T \to A] \times T \to J \times Timestamp. \tag{4.20}$$

Let $R = E \times C \times Attr \times J \times Timestamp$ be the set of records.
Individual Monitoring: Suppose that entity e and characteristic χ determined the attribute P to be observed, then the attribute to be checked will be $P(e, \chi)$. Then

$$Monitor : E \times C \to R \tag{4.21}$$

When you apply e is an entity, χ is the characteristics, then $P(e, \chi)$ is the attribute to be checked defined by:

$$Monitor(e, \chi) = (e, \chi, P(e, \chi), Obs(e, \chi, [[e, \chi]])) \tag{4.22}$$

Monitoring over intervals and at instants: In this the monitoring is done in a specific time interval. The suggested monitoring mapping for time interval T and T using mappings 3.3 and 3.8 will become:

$$Monitor : E \times C \times T \times T \to R \tag{4.23}$$

Therefore, the monitoring is defined by

$$Monitor(e, \chi, t_1, t_2) = (e, \chi, P(e, \chi), Obs(e, \chi, [[e, \chi]], t_1, t_2)) \tag{4.24}$$

Monitoring stream: When the monitoring is done for a specific time t then the $t_1 = t_2 = t$

which gives rise to the stream of records as follows:

$$monitor : E \times C \to [T \to R]. \tag{4.25}$$

which is defined by

$$monitor(e, \chi) = (e, \chi, P(e, \chi), Obs(e, \chi, [[e, \chi]])). \tag{4.26}$$

4.2.1 Behaviors with Timestamps

So far, we have introduced timestamps into the general model of Johnson et al. (2017)Johnson and Tucker (2013) as summarized in chapter 3. We did this by timestamping observations; in this way timestamps appear in records.
However, having made distinctions between a number of notions of time, in particular introducing the general model with real-world time in the form of timestamps, we can create a new and other variant of the general model as follows.
Recalling from section 3.4.2 behaviors are represented in the general model as pure data streams

$$a : T \to A.$$

However, now we can refine behaviors by modeling them by a stream of data with timestamps, which have the form

$$a : T \to A \times Timestamp.$$

This formal behavior opens up the model to new examples, we will not rewrite the general model again with this new form of behaviors rather we will meet examples in the chapter followed where access to resources are timestamped and these timestamps are a key part of the observation process.
Actually, it is easy to see three naturally occurring roles for time: there is clock T counting events and behaviors which may be timestamped by real-world time $Timestamp$; these behaviors are subsequently observed by a monitoring process at another real-time and recorded with its own timestamp. Such a situation combines both approaches we have discussed.

4.3 Notifications

4.3.1 Introduction

A notification within a system is a set of protocols and procedures that incorporate both human, and computer components to timely generate and send messages to a person or group of persons (Alfarsi and Juma, 2018); it is the process of alerting an individual of an occurrence of a particular event. Notification in the system could be simple, complex, and emergency notification system (Hansen and Atkins, 1993). A simple notification system uses a single means of communication, like email, while complex notification is designed to send out crucial information and would typically use other methods of communication. It could also include human elements

to make each person involved alerted. Emergency notification may use of the phone system, television broadcast, and a host of other communication methods.

In this section, we will explain the types, forms, and purpose of notifications, how notifications need action from the user, the priorities of notifications, and define notifications in algebraic models.

4.3.2 Types, Forms, and Purpose of Notifications

There are several types of notifications used by the system to interact with the user, some of which are discussed below:

- Error Message Notification: Notification that is mostly used to display an error to the user (Hansen and Vaudreuil, 2004). For example, in the login scenario, the error message notification is sent to the user when the login fails due to incorrect credentials.
- Alert Message Notification: Notification used to inform the user that there are some information which he/she needs to be aware of is known as alert (Gusev et al., 2014). For example, in the login scenario, the alert message notification is sent to the user when a login attempt occurs through untrusted devices and locations.
- Exception Message Notification: This type of notification is mainly used to inform the user about the faults and functional limitations of the system (Liu Wenyin et al., 2002). For example, in the login scenario, the exception message notification can be used to let the user know that the mobile application login is not working. Where else, the user can login from the web browser.
- Success Message Notification: This type of notification is mainly used to inform the user that the specific activities conducted by the user are successful (Simonis and Dibner, 2007). For example, the success message notification is sent to the user when the login is successful in the login scenario.
- Confirmation Message Notification: This type of notification is used to get confirmation from the user for a specific action that occurred in the system to ensure security (Mooloo and Fowdur, 2013). For example, in the login scenario, if the user tries to login into the system using a new device, the system will send the Confirmation Message Notification to get confirmation from the user whether he or she is the one who is trying to login to the system using a new device.
- Warning Message Notification: This notification type is used to provide warnings with respect to user actions on the system (Zhang et al., 2017). For example, in the login scenario, the warning message notification is sent to the user when he or she continuously fails the login.
- Informational Message Notification: This notification type is used to provide information regarding the system changes that occurred to the user (Sposaro and Tyson, 2009). For instance, in the login scenario, system information regarding the login security approach is sent to the user and also advice to not share their password with others.

- Badges Message Notification: This notification type is used by the system to provide achievements and milestone of the user in the system (Santos et al., 2013). For example, the system can send a badge to the user for using the system for 100 continuous days or any specific milestone.
- Status Indicators Notification: This notification type is used by the system to inform the user about the current status of his or her activity or profit (Lombardo et al., 2003). For example, the system can send a notification that you are login to the system or logout of the system.
- Acknowledgement Message Notification: This Notification type is used to inform the user about the feedback for the specific user action (Lee and Oh, 2011). For example, the system can use the acknowledgement notification to inform that the user placed the order successfully.

4.3.3 Action from the user

It is not all notifications require user action. It is important to know that the notification should clearly indicate to the user when a certain action is required. For example, there should be a clear icon showing that the action is required as well as the notification should continue reminding until the input is received from the user. For example, if the user tries to login into the system using a new browser, then the notification asks the user whether he or she trusts the browser for future usage or not. This action from the user is very important for the security system. Therefore, until the user selects 'Yes' or 'No', the system should remind the user with repeated notifications.

4.3.4 Priorities of Notifications

There are priorities that can be given for the notification to catch the user's attention. The key priorities are Low, Medium, and High context (Streefkerk et al., 2007). The low-priority notifications are just system information to the user (more like a log of activities). The medium-priority notifications are system information about user actions. These types of notifications will require some actions from the user most of the time. The high-priority notifications are more of an alarming situation that requires immediate attention from the user without fail.
The types of notifications and the priorities are interlinked. Depending on the types of notification, the priority of the notification will be determined. The table below provides the link between the priorities and the types of notification with examples of login scenario.

Priority	Types of Notifications	Notification Scenario in Login
Low	Informational Message Notification, Success Message Notification, Badges Message Notification and Status Indicators Notifications	When successful user login to the system
Medium	Error Message Notification, Exception Message Notification and Acknowledgments Message Notification	When the user enter credential that is wrong
High	Warning Message Notification, Confirmation Message Notification and Alert Message Notification	When the user encounter wrong login try for n number of times and the system blocks for k minutes

4.3.5 Notifications: Algebraic Models

Basically, notification messages are responses from the system which depend on the judgment of observation to interact with the user to either notify the user, notify and require user action or it could be no response.

Let $Resp$ be a set of messages including the element $noResponse$.
Let,

$$resp : J \to Resp \tag{4.27}$$

Where $resp$ accept a judgement value $j \in J$ as input, and returns a $Resp$.

The current model provides a simple way to handle notifications, where the system decides whether to notify the user, request an action, or give no response. While this works for basic cases, real-world systems often need more detail.

One way to improve the model is by adding source and destination details. This means identifying where the notification comes from (e.g., a system component) and who should receive it (e.g., a user or another system). This helps direct messages to the right place.

Another useful addition is priority levels. Some notifications are more urgent than others. For example, an error that needs immediate attention should be treated differently from a general update. Assigning priorities helps the system respond appropriately.

Finally, adding extra details like timestamps or reasons for the notification can make the system more efficient. This helps in tracking messages, improving responses, and automating actions.

By including these features, the notification system becomes more organized and useful, ensuring that important messages are delivered and handled correctly.

4.4 A Model of Monitoring with Notifications

In this section, we will extend our basic model of monitoring by incorporating notification messages into the general model which will yield to a new modeling of observation as well as

record functions.
Please refer to section 3.4 for a detailed explanation of our basic model.
Now, we will adapt the definition relevant to monitoring observation to include a notification. Upon incorporating the notification into mapping 3.9, the observation map becomes

$$Obs : E \times C \times [T \to A] \to J \times Resp. \tag{4.28}$$

In the presence of notification, the set R of records is implemented as follows

$$R = E \times C \times Attr \times J \times Resp. \tag{4.29}$$

4.5 Summary

This chapter introduced the role of time and timestamps in computing systems and their significance in monitoring processes. It explored how timestamps provide temporal markers for recorded events and discussed various timestamp formats used across different operating systems and programming languages. An algebraic model for timestamps was developed, defining timestamps in both standard and alternative formats. The chapter then extended the general monitoring model by incorporating timestamps into observation and recording functions, ensuring that monitored behaviors are temporally marked for accuracy. Furthermore, it examined the concept of notifications, categorizing different types and priorities of notifications within a system. A basic algebraic model for notifications was presented, defining notifications as system responses dependent on monitoring judgments. However, this model remains simplistic, lacking details such as notification sources, destinations, and priority levels. Future work could extend this model by incorporating richer structures for notifications, including sender-receiver information, urgency levels, and response dependencies, to improve adaptability in real-world monitoring applications.

Chapter 5

Virtual Access Control and Monitoring Systems: Login

5.1 Introduction

Access to resources, physical and virtual, usually involves something resembling a 'key.' The 'key' must be accepted or rejected, and the process of validation is based on monitoring. Authentication mechanisms, including password-based and key-based systems, are essential for ensuring controlled access to secure environments (Santhosh Kumar and Sinha, 2021),(Gavilan and Martinez, 2022). This general observation suggests test case studies for our general model.

Access to computer accounts and digital resources has long been based on passwords (de Carné de Carnavalet and Mannan, 2014). Passwords can be as simple as four digits, common in ATM and credit cards, or complicated with more than one string organized hierarchically. The effectiveness of password-based authentication depends on various security factors, such as complexity, unpredictability, and storage mechanisms (Esposito et al., 2014). Some aspects of passwords are common in other authentication mechanisms, such as hashing and biometrics, which further enhance security by reducing direct password exposure (Carzaniga et al., 2000),(Vrbaski et al., 2018).

This chapter begins with defining a password. It then develops a formal model of passwords based on their structure, constraints, validity and strength. Then we present a basic model of user registration by describing three distinct actions; namely: creating a new account, changing a password of an existing account and deleting an account.

We now apply our general model of monitoring to investigate the process of login in. A login case study is introduced where entities, attributes, observation and monitoring are all described. In this case study, attributes are the properties corresponding to login behaviour; each login attempt will be either successful login or failed login. Observation involves making a judgement upon the login behaviour of a certain device. Monitoring is the collection, evaluation and recording of observational data about the behaviour of entities. The output of a monitoring system is records of observations. We will explain the failed login scenarios (eg: k successive failed login interventions) as interventions and describe their entities, characteristics, record, trigger conditions and actions. Finally, this chapter introduces notifications.

5.2 Password Structure, Validity and Strength

A password is a series of characters used to validate the identity of a user. During the authentication process, passwords are usually used together with a username. Passwords are intended to be known only to the user and allow the user to obtain access to a device, website, file, program, etc. The combination of username and password is usually unique and provides a record of access.

Passwords can range in length and a choice of letters (upper or lower case), digits and symbols. In this section, we will develop a formal model of passwords based on their structure, validity and strength.

5.2.1 Passwords and their properties

Password alphabets.
Let

$$Upper = \{A, B, C, ..., Z\} \tag{5.1}$$

$$Lower = \{a, b, c, ..., z\} \tag{5.2}$$

$$Digit = \{0, 1, 2, ..., 9\} \tag{5.3}$$

$$Symbol = \{@, /, -, \%, \&, ...\} \tag{5.4}$$

be a set of all possible characters used to create a password. So,

$$Alphanumeric = Upper \ \cup \ Lower \ \cup \ Digit \tag{5.5}$$

and the full alphabet which we will use is

$$A = Alphanumeric \ \cup \ Symbol \tag{5.6}$$

Given any symbols $a_1, a_2, ..., a_n \in A$, we can form a string

$$p = a_1 a_2 ....a_n \tag{5.7}$$

Let $String = A^*$ be the set of all strings over A including the empty string. Thus, let Pwd be a set of all acceptable passwords which is a subset of $String$

$$Pwd \subseteq String \tag{5.8}$$

Password length and password predicates
Each password may have a condition on its length. We suppose that a password is a string p with a minimum length l_{min} and maximum length l_{max}. The length of string p satisfies:

$$l_{min} \leq |p| \leq l_{max} \tag{5.9}$$

Next, we consider some predicates on passwords. These predicates check whether a password contains a type of character: Let

$$Upper(p) \equiv (\exists_i)\ [1 \leq i \leq |p|\ \wedge\ a_i \in Upper] \tag{5.10}$$

$$Lower(p) \equiv (\exists_i)\ [1 \leq i \leq |p|\ \wedge\ a_i \in Lower] \tag{5.11}$$

$$Digit(p) \equiv (\exists_i)\ [1 \leq i \leq |p|\ \wedge\ a_i \in Digit] \tag{5.12}$$

$$Symbol(p) \equiv (\exists_i)\ [1 \leq i \leq |p|\ \wedge\ a_i \in Symbol] \tag{5.13}$$

These predicates check if a password contains only one type of character: Let

$$Upper_{all}(p) \equiv (\forall_i)\ [1 \leq i \leq |p|\ \wedge\ a_i \in Upper] \tag{5.14}$$

$$Lower_{all}(p) \equiv (\forall_i)\ [1 \leq i \leq |p|\ \wedge\ a_i \in Lower] \tag{5.15}$$

$$Digit_{all}(p) \equiv (\forall_i)\ [1 \leq i \leq |p|\ \wedge\ a_i \in Digit] \tag{5.16}$$

$$Symbol_{all}(p) \equiv (\forall_i)\ [1 \leq i \leq |p|\ \wedge\ a_i \in Symbol] \tag{5.17}$$

5.2.2 Examples of password conditions

Common Services. Conditions on passwords vary enormously. Table 5.1 tabulates familiar examples of services and their corresponding password conditions as of date 2014 (de Carné de Carnavalet and Mannan, 2014).

Table 5.1: Examples of different services with password conditions

Service	Min Length	Max Length	Charsets Required
Twitter	6	999	-
Apple	8	32	1 Upper, 1 Lower and 1 Digit
Paypal	8	20	2 Charsets
Skype	6	20	2 Charsets or Upper Only
Microsoft	1	-	-
Google	8	100	-
Yahoo!	6	32	-
eBay	6	20	2 Charsets

Note that PayPal counts uppercase and lowercase letters as a single char set.

Formalising password conditions. The formulas of password validation for a service have the following form:

$$AppleCond(p) \equiv Upper(p) \wedge Lower(p) \wedge Digit(p) \tag{5.18}$$

$$\begin{aligned} PayPalCond(p) \equiv & (Upper(p) \wedge Lower(p)) \vee (Upper(p) \wedge Digit(p)) \vee \\ & (Upper(p) \wedge Symbol(p)) \vee (Lower(p) \wedge Digit(p)) \vee \\ & (Lower(p) \wedge Symbol(p)) \vee (Digit(p) \wedge Symbol(p)) \end{aligned} \tag{5.19}$$

$$\begin{aligned} SkypeCond(p) \equiv & ((Upper(p) \wedge Lower(p)) \vee (Upper(p) \wedge Digit(p)) \vee \\ & (Upper(p) \wedge Symbol(p)) \vee (Lower(p) \wedge Digit(p)) \\ & \vee (Lower(p) \wedge Symbol(p)) \vee (Digit(p) \wedge Symbol(p)) \\ & \vee Upper_{all}(p)) \end{aligned} \tag{5.20}$$

$$\begin{aligned} eBayCond(p) \equiv & (Upper(p) \wedge Lower(p)) \vee (Upper(p) \wedge Digit(p)) \vee \\ & (Upper(p) \wedge Symbol(p)) \vee (Lower(p) \wedge Digit(p)) \vee \\ & (Lower(p) \wedge Symbol(p)) \vee (Digit(p) \wedge Symbol(p)) \end{aligned} \tag{5.21}$$

Sets of acceptable passwords. The set of all possible passwords for each service are given by:

$$TwitterPwd = \{p | p \in String, 6 \leq |p| \leq 999\} \tag{5.22}$$

$$ApplePwd = \{p | p \in String, 8 \leq |p| \leq 32 \wedge AppleCond(p)\} \tag{5.23}$$

$$PaypalPwd = \{p | p \in String, 8 \leq |p| \leq 20 \wedge PaypalCond(p)\} \tag{5.24}$$

$$SkypePwd = \{p | p \in String, 6 \leq |p| \leq 20 \wedge SkypeCond(p)\} \tag{5.25}$$

$$MicrosoftPwd = \{p | p \in String, 1 \leq |p| \leq l_{Max}\} \tag{5.26}$$

$$GooglePwd = \{p | p \in String, 8 \leq |p| \leq 100\} \tag{5.27}$$

$$YahooPwd = \{p | p \in String, 6 \leq |p| \leq 32\} \tag{5.28}$$

$$eBayPwd = \{p | p \in String, 6 \leq |p| \leq 20 \wedge eBayCond(p)\} \tag{5.29}$$

Therefore, by creating formulae from such predicates, we can define for an abstract service $Serv$, the set

$$ServPwd = \{p | p \in String, l_{min} \leq |p| \leq l_{max} \wedge ServCond(p)\} \tag{5.30}$$

of all acceptable passwords corresponding to this service.

The importance and Purpose of the General Predicate $ServCond(p)$ provides a generalized way to express password validation conditions across different services. Instead of defining separate rules for each service individually, we abstract the password condition into a single predicate function, making the model more scalable and reusable. Moreover, it adapts to any service by adjusting the predicate $ServCond(p)$, it also ensures all passwords for a service meet the predefined rules and Provides a high-level, reusable definition for acceptable passwords, simplifying password validation across services.

5.2.3 Validity and Strength

In order to check the validity of the password whether it is valid or not valid, we define the following:

$$validate : String \times \mathcal{P}(String) \to \{valid, notvalid\} \tag{5.31}$$

$$validate(p, ServPwd) = \begin{cases} valid & \text{if } p \in ServPwd \\ notvalid & \text{if } p \notin ServPwd \end{cases} \tag{5.32}$$

Suppose we have a simple strength classification of passwords in which the three sets are disjoint:

$$Pwd = Weak \cup Medium \cup Strong \tag{5.33}$$

In practice, interpretations of weak, medium and strong (and other) categories vary considerably with different services; usually, they can be partially found out by re-engineering (de Carné de Carnavalet and Mannan, 2014).
Strong passwords, commonly allow the four types of characters and prefer a longer password using a combination of at least one from each character type.

Strength typically has to do with the size of the search space for brute algorithms: more symbols to choose from and longer passwords expand the search space. Thus, the length of the password is a key factor in evaluating its strength. The longer the password, the stronger it is.

5.3 Registration Mode

Before modeling the login process, we need to have a model of a store for usernames and passwords, which we call a *registry*. To create a basic model of the registry, we have to explain the following three actions:

1. Creating a new account.
2. Changing a password of an existing account.
3. Deleting an account.

For simplicity we are not considering the importance of password recovery; could be added.

5.3.1 Defining the registry

First, let's define the *basic data*. Let
$String$ = set of all finite strings of symbols that are available to make usernames and passwords, including the empty string
$User \subseteq String$ = a set of all acceptable usernames
$Pwd \subseteq String$ = a set of all acceptable passwords
$Reg \subseteq User \times Pwd$ = a set of registered usernames and associated passwords
An attempt at a login is a pair $(x, y) \in String \times String$ and need not be accepted by the system. In particular, Reg is a finite subset, i.e.,

$$Reg \in P_{fin}(User \times Pwd) \tag{5.34}$$

A valid username and password form a pair $(u, p) \in Reg$.

5.3.2 Creating a new account

In order to create a new account, the proposed username x and password y should be valid and the username must not have been previously used.
The validity of a username x and password y is simply:

$$(x, y) \in User \times Pwd \tag{5.35}$$

To check if the username x is taken (or not) in a registry Reg, we need a predicate.
First, we compute the set of all usernames in a registry by

$$username : P_{fin}(User \times Pwd) \to P_{fin}(User) \tag{5.36}$$

defined for $Reg \in P_{fin}(User \times Pwd)$,

$$username(Reg) = \{x : (x, y) \in Reg\} \tag{5.37}$$

Second, we can define a Boolean function

$$notTaken : String \times P_{fin}(User \times Pwd) \to \mathbb{B} \tag{5.38}$$

Given $x \in String$ and $Reg \in P_{fin}(User \times Pwd)$ we can check if the username is taken or not using

$$notTaken(x, Reg) = \begin{cases} t & \text{if } x \notin username(Reg) \\ f & \text{if } x \in username(Reg) \end{cases} \tag{5.39}$$

So, creating a new account is defined by

$$add : String \times String \times P_{fin}(User \times Pwd) \to P_{fin}(User \times Pwd) \tag{5.40}$$

$$add(x, y, Reg) = \begin{cases} Reg \cup \{(x, y)\} & \text{if } x \in User \wedge notTaken(x, Reg) \wedge y \in Pwd \\ Reg & \text{otherwise} \end{cases} \tag{5.41}$$

This process will likely need notifications, see section 5.6.

5.3.3 Deleting an account and updating password of an existing account

Having described how a user registers and accesses an account, in this subsection we model how a user can delete their account.

We define a function

$$delete : String \times String \times P_{fin}(User \times Pwd) \to P_{fin}(User \times Pwd) \tag{5.42}$$

Thus, deleting an account is defined by

$$delete(x, y, Reg) = \begin{cases} Reg - \{(x, y)\} & \text{if } (x, y) \in Reg \\ Reg & \text{if } (x, y) \notin Reg \end{cases} \tag{5.43}$$

To change a password, the user must be signed into the account.
The new password should be a valid password where $p \in Pwd$.

$$newPwd : String^2 \times String^2 \times P_{fin}(User \times Pwd) \to P_{fin}(User \times Pwd) \tag{5.44}$$

which is defined by

$$newPwd((x, y_1), (x, y_2), Reg) = \begin{cases} (Reg - \{(x, y_1)\}) \cup \{(x, y_2)\} & \text{if } (x, y_1) \in Reg \\ & \wedge\ y_2 \in Pwd \\ Reg & \text{otherwise} \end{cases} \tag{5.45}$$

Here we do not require y_2 to be new, i.e. $y_1 = y_2$ is acceptable.

5.4 Basic Model of Login

A login is a process of controlling the access of a user by checking the login details, where login details contain a username and password. If a pair of username and password is valid then it is a successful login, otherwise it is a failed login. For each successful login the registry changes.

5.4.1 Changing Registries

Recall from section 5.3.1 our basic data used to define the registry.
In particular, *Reg* is a finite subset and the state of the registry changes over time. We introduce discrete time, defined as counting clock cycles. Let

$$T = \{0, 1, 2, ..., t, ...\} \tag{5.46}$$

We introduce a mapping from discrete time to the finite powerset $P_{fin}(User \times Pwd)$. Changing notation for the set (5.34), Reg now becomes:

$$Reg : T \to P_{fin}(User \times Pwd) \tag{5.47}$$

If at time t, username u has password p in the registry, then:

$$(u, p) \in Reg(t) \tag{5.48}$$

5.4.2 Entities

In monitoring, the entity to be observed is a device, and its behaviour is its attempts to login to the service.

Let $Device$ be the set of all devices that can be used to login to the system, and $Char$ be the set of characteristics of all these devices.
We define the behaviour of an entity as a stream of data.

$$b : \mathbb{N} \to String \times String \times T \tag{5.49}$$

which models a possible sequence of attempts at login from a device in discrete time measured by T.

$$b(0), b(1), ..., b(n), ... \tag{5.50}$$

Specifically, the n^{th} attempt at login by a device is:

$$b(n) = (u(n), p(n), t(n)) \tag{5.51}$$

where, $t(n)$ is a *time stamp*.
The behaviour of the entities is now represented by the map

$$[[-, -]] : Device \times Char \to [\mathbb{N} \to String \times String \times T] \tag{5.52}$$

such that for device $d \in Device$, with characteristics $c \in Char$ for n^{th} attempts $n \in \mathbb{N}$ belongs to the stream

$$[[d, c]] : \mathbb{N} \to String \times String \times T \tag{5.53}$$

and $[[d, c]](n) = n^{th}$ attempt at login by device d with characteristics c.

It is natural to assume that the timestamps increase monotonically i.e.

$$n, m \in \mathbb{N}, n < m \implies t(n) < t(m) \tag{5.54}$$

5.4.3 Attributes

Attributes are properties of login behaviour: each login attempt will be compared with the information in the registry which will be either succeed or fail.
Thus, the set of judgements is

$$Judge = \{accept, tryagain\} \tag{5.55}$$

To check on whether or not an attempt login works, we use the following map

$$Match : String \times String \times T \times [T \to P_{fin}(User \times Pwd)] \to Judge \tag{5.56}$$

Given $x \in String$ and $y \in String$ at certain time $t \in T$ and $Reg \in [T \to P_{fin}(User \times Pwd)]$ which will compute a judgement

$$Match(x, y, t, Reg) = \begin{cases} accept & \text{if } (x, y) \in Reg(t) \\ tryagain & \text{if } (x, y) \notin Reg(t) \end{cases} \tag{5.57}$$

we think of Reg as a parameter of the attribute (as it changes as users come and go). For simplicity, we can fix this parameter and rewrite $match$:

$$match_{Reg} : String \times String \times T \to Judge \tag{5.58}$$

Thus, we take $Attr = \{match_{Reg}\}$.

5.4.4 Observation

Observation involves making a query about, or testing, the login behaviour of a device. The attribute is a match of a login attempt with the registry. In order to make this judgement, we define the following map

$$Obs_0 : Attr \times [\mathbb{N} \to String \times String \times T] \times \mathbb{N} \to Judge \tag{5.59}$$

which, given an attribute $match_{Reg}$ and series of logins $b \in [\mathbb{N} \to String \times String \times T]$ for n^{th} attempts $n \in \mathbb{N}$:

$$Obs_0(match_{Reg}, b, n) = match_{Reg}(b(n)) \tag{5.60}$$

5.4.5 Monitoring

Monitoring is confined to the collection, evaluation and recording of observational data about the behaviour of entities. The outputs of a monitoring system are simply records of observations. First, let

$$Record = Device \times Char \times Attr \times Judge \tag{5.61}$$

be the set of records.

We observe *Device* d and *Char* c then $match_{Reg}$ is the attribute to be checked using mappings 5.52 and 5.59, the monitor function becomes

$$Monitor : Device \times Char \to [\mathbb{N} \to Record] \tag{5.62}$$

defined for $d \in Device$ and $c \in Char$ by

$$Monitor(d, c)(n) = (d, c, match_{Reg}, Obs_0(match_{Reg}, [[d, c]], n)) \tag{5.63}$$

This concludes the application of our general monitoring model of Chapter 3 to the essentials of the login process.

5.5 Failed logins as interventions

In section 5.4, the model of login discussed was either a successful login or failed login based on the validity of the login details. The fail login model considered in section 5.4 was basic, with no interventions to try again. However, in this section, we will explain how intervention can behave when the fail login occurs with different types of conditions and notifications. We now consider the following login protocol: if there are k successive failed login attempts in a short period Δt, the user will be temporarily blocked for a period of β time units.

5.5.1 Entities and characteristics

The entity to be monitored is a physical device which gives access to an account, and the behaviour to be monitored is login attempts. The characteristic is the status or state of the device in relation to login attempts, these states are used to classify the device's mode of operation and determine the appropriate intervention or response based on observed behavior.

Let C be the set of characteristics of the device. These characteristics label the modes of the device:

$$C = \{normal, tempblock\} \tag{5.64}$$

Here *normal* represents normal mode of operation where the device is free to make login attempts; and *tempblock* represents a temporarily blocked mode of operation of the device where login attempts failed more than k times.

5.5.2 Record

The record is an output of the monitoring component containing the evaluated attributes of observed data. A record r has the form:

$$r = (d, c, match, j) \in Record = Device \times Char \times Attr \times Judge \tag{5.65}$$

where,
d is the device used to login
c is the characteristics of the device

match is the attribute observed
j is the judgement
In general, each time the user attempts to login to the system, the record will be generated. The streams $\mathbb{N} \to Record$ of records might be obtained using mapping equation 5.62.

5.5.3 Useful timing functions

The following functions will be utilized to analyze the behaviour of the login within time bounds. Let Bd be the set of available time bounds and let $\Delta t \in Bd$.
First, we define the last login (x, y, t) that took place at time t.

The number of the most recent login is represented by $m(t)$ and it is defined as follows:

$$m(t) = max\{n \in \mathbb{N} : \pi_t(b(n)) \leq t\} \tag{5.66}$$

where $b(n)$ represents the n^{th} login attempt by a device and π_t is the projection onto the time component of a login tuple.
We are particularly interested in blocks of k consecutive logins

$$length(b, t, k) = \pi_t b(m(t)) - \pi_t b(m(t) + (1 - k)) \tag{5.67}$$

where $length(b, t, k)$ will calculate the length of period of the last k successive logins and *checkspeed* will check the speed of the last k successive logins if it is less than or equal to Δt

$$checkspeed : Block \times T \times \mathbb{N} \times Bd \to \{t, f\} \tag{5.68}$$

$$checkspeed(b, t, k, \Delta t) = \begin{cases} t & \text{if } length(b, t, k) \leq \Delta t \\ f & \text{if } otherwise \end{cases} \tag{5.69}$$

The parameter Δt can take on quite different values if one is trying to detect whether the user is a human or a robot. In case of human, Δt could take the value of 3 minutes, allowing him/her to check whether their passcode is a successful match or not. However, in case of cyber attack (robot), the timing might be a matter of k failures where k is large and Δt is small.

5.5.4 k successive failed login intervention

A monitoring intervention is to be triggered when k successive failed login attempts are made when the device is in normal mode within a fixed time frame Δt.
T represents clock cycles that define the time stamp $t \in T$ of a device that is monitored.
We consider the behaviour of the device to be a map from number of a login attempts to the tuple $(username, password, timestamp)$

$$match(x, y, t, Reg) = \begin{cases} accept & \text{if } (x, y) \in Reg(t) \\ tryagain & \text{if } (x, y) \notin Reg(t) \end{cases} \tag{5.70}$$

Computes an outcome of inputting a password where $Reg(t)$ can be thought of as an attribute.

Now, set $\pi : [\mathbb{N} \to String \times String \times T] \to [\mathbb{N} \to \{1, 0\}]$

$$\pi(b)(n) = \phi_{tryagain}(Match_{Reg}(b(n)) \tag{5.71}$$

where $\phi_{tryagain}$ returns 1 if the login attempt is failed, otherwise will return 0.

For example, the aggregation operation $agg : [\mathbb{N} \to String \times String \times T] \to \{0, .., k\}$ is defined by:

$$agg(b, n) = \sum_{n=m(t)+1-k}^{m(t)} \pi(b)(n) \tag{5.72}$$

where this function will compute for time t the number of failed logins in the most recent k logins.

Now we

1. look for blocks of k failed logins, and
2. check the total period of time they take against the fixed time frame Δt to trigger an intervention.

5.5.5 Trigger conditions and actions

Let $tc : J \to \mathbb{B}$ map from a sequence of judgements to a Boolean designed by setting: $tc(exceptional) = t$ and $tc(nonexceptional) = f$.

Let, $act : C \to C$ on characteristics is defined by $act(normal) = tempblock$ and $act(tempblock) = tempblock$
The *tempblock* will automatically expire after a set period of time and will become: $act(normal) = tempblock$ and $act(tempblock) = normal$.

Let β be the period of block time, $\pi_t b(m(t))$ be the time at which the block period started, corresponding to the time of the k failed login attempt and $\max(0, \cdot)$ be the function that ensures the remaining block time never becomes negative, making it zero when the block period has expired. so the remaining block time is defined as follows:

$$\text{remaining}(b, k, t, \beta) = \max(0, \pi_t b(m(t)) + \beta - t) \tag{5.73}$$

$$\text{reset}(b, t, k, \beta) = \begin{cases} t, & \text{if remaining}(b, k, t, \beta) = 0 \\ f, & \text{otherwise} \end{cases} \tag{5.74}$$

5.6 Notifications

Notification is a key aspect of the login process (Santhosh Kumar and Sinha, 2021). Gavilan and Martinez (2022) confirmed that the notification is key to enhancing the overall user experience. Notification, in general used to create an effective communication protocol between the computer

components and humans (Esposito et al., 2014) by alerting the user with respect to certain login attempts or notifying about the results of the login attempts. Notifications in a login system can be grouped according to their complexity, which are simple (Carzaniga et al., 2000), complex (Vrbaski et al., 2018) and emergency (Malizia et al., 2010) notifications. A brief discussion about the use of different notifications with respect to their complexity in a login scenario can be seen below:

- **Simple Notification System:** It is used in the login scenario to allow the system to communicate with the user. For example, the system sends an email to the user to confirm a particular login action.
- **Complex Notification System:** This approach is used in advanced login systems where the notification is designed to communicate critical information to the users. The notification process can use various communication methods like email, SMS, and push-up messages.
- **Emergency Notification System:** This approach is used whether there is a threat to the security of the system. This notification process could use all the possible mediums to communicate to the user about login-related breaches or incidents

5.6.1 Notifications for our login models:

Even in simple login scenarios, there is a wide choice of useful messages. For example, in the case of a success or fail situation the following set of messages is appropriate:

Messages = {'You have successfully logged in.', 'Your login failed. Try again.'}

In the case where there is a bound on the number of login attempts the following set of messages is appropriate:

Messages = {'You have successfully logged in.', 'Your login failed. Try again.','Your account is temporarily blocked due to many failed logins. Try again later.'}

Of course, the notification could be a little more helpful in this case if the user it is warned that there is a bound:

Messages = {'You have successfully logged in.', 'Your login failed. You have k-1 further attempts.', 'Your account is temporarily blocked due to many failed logins. Try again later.'}

5.6.2 Classification of notifications on login processes:

For a set

$$Messages = \{m_1, ..., m_n\} \tag{5.75}$$

of messages belonging to a notification, the different messages can perform different functions. Chapter 3 identified 12 types of notifications; Some of these scenarios can be seen below:

- **Error Message Notification:** This notification is activated when the user enters an incorrect credential to log in to the system. In this scenario, the system sends an error message informing the user that the login failed due to wrong credentials.

- **Alert Message Notification:** This notification is activated when a user tries to log in to the system using an untrusted device or location. In this scenario, the system sends an alert message to the user warning that the login attempt is occurring from an untrusted source and asks them to verify their identity.

- **Exception Message Notification:** This notification is activated when the mobile application login is not working and the system directs the user to use a web-based login. In this scenario, the system sends an exception message to the user explaining the mobile application issue and suggests an alternative method to log in, such as providing the link to the web login.

- **Success Message Notification:** This notification is activated when the user successfully login to the application with the correct credentials. In this scenario, the system sends a success message confirming that the login was successful.

- **Confirmation Message Notification:** This notification is activated when the user tries to log in to the system using a new device. In this scenario, the system sends a confirmation message to the user to confirm whether he or she is trying to log in from the specific new device. This notification is mainly utilised to enhance the security of the system.

- **Warning Message Notification:** This notification is activated when the user fails to log in to the system repeatedly within a short period of time. In this scenario, the system sends a warning message to the user that there have been several failed attempts to log in to the system.

- **Informational Message Notification:** This notification is activated when the system updates the security features of the login process. In this scenario, the system sends an informational message informing the user of detailed information regarding the security enhancements.

- **Badges Message Notification:** This notification is activated when the user achieves a specific milestone in the system. In this scenario, the system sends a badge message to the user congratulating achieving a specific milestone. For example, if the milestone is to log in to the system 100 times within 6 months and if the user achieves it, the badge message notification will be sent to the respective user.

- **Status Indicators Notification:** This notification is activated when the user logs in and out of the system. In this scenario, the system sends a status message to the user informing the login information, such as time, device name and location, also the log-out information, such as time.

- **Acknowledgement Message Notification:** This notification is activated when the user successfully login to the system. In this scenario, the system sends an acknowledgment message to the user to confirm whether the successful login was from him or her. This type of messaging can be used for confirmation, feedback, user action confirmation, transactional completion, and user engagement.

5.7 Summary

This chapter explores access control mechanisms, focusing on the roles of passwords, monitoring, and interventions in login systems. It begins by formally defining passwords, modeling their structure, constraints, validity, and strength. A registry model is introduced to manage user accounts, detailing actions such as account creation, password changes, and account deletion. The chapter then examines a login case study, where monitoring tracks login attempts, distinguishing between successful and failed logins. Failed logins are modeled as interventions, incorporating trigger conditions that temporarily block access after multiple failed attempts within a specified time frame. The concept of notifications is also introduced, highlighting different types of messages—such as alerts, warnings, and confirmations—that enhance security and user awareness. By integrating timestamps, this chapter extends and applies the general monitoring framework of chapter 3 and 4 to the login process, providing a formalized approach to access control and user authentication management. limitations include assumptions about password security policies and the fixed intervention criteria. Future work could extend the model by integrating adaptive security measures and multi-factor authentication for users; and dynamic risk assessment and AI-driven anomaly detection for implementations.

Chapter 6

Physical Access Control and Monitoring Systems: Buildings

6.1 Introduction

In today's world where technology is highly developed, it is important to have systems in place that control and monitor who can enter buildings and facilities. These systems make sure that only authorized individuals are allowed entry to different parts of a building, ensuring security. This chapter will take a closer look at the various components of these physical access control and monitoring systems, focusing on concepts like Identity, Space, Time, and User Registry to explain their relationship. By understanding these systems, we can gain insight into how they function in real-life situations. This case study further tests and explores the general model of monitoring with its new components in Chapter 3.

6.2 Digital Locks and Keys

6.2.1 The General Concept

In the field of physical access control and monitoring systems, identity plays a crucial role. It refers to the unique traits or characteristics that distinguish one person or entity from another. Identity verification, especially in the context of granting access, can involve several methods, including biometric data (such as fingerprints or retina scans), facial recognition, and access cards. These methods are fundamental in helping to verify the identity of individuals requesting access. While biometric data is often considered more secure than traditional methods (such as passwords or access cards), it's important to note that no system is entirely foolproof. Some biometric techniques, like retina scans or facial recognition, can be vulnerable to sophisticated attempts to replicate or spoof the data. As a result, while biometric verification is generally a reliable security measure, combining it with other forms of security (a layered approach) can enhance its effectiveness and address potential vulnerabilities (Wang et al., 2016).

A registry is a data management system that plays a crucial role in managing identity information. It acts as a centralized repository that stores and organizes data related to individual identities and access privileges. This registry holds information about authorized personnel,

their access rights, and any associated restrictions. Access to this registry is tightly controlled to prevent unauthorized changes or breaches, ensuring that only authorized individuals can modify or view sensitive data.

For example, in a commercial office building utilizing a digital access control system, employees are issued access cards that are registered in the system. Each access card is linked to the individual's identity, including their name, photo, and specific access permissions. The registry securely maintains this information and ensures that only authorized individuals can enter designated areas of the building, based on the access rights associated with their credentials (Olayiwola et al., 2021).

6.2.2 Examples

To illustrate the concepts of identity and registry further, consider a few real-world examples:

Example 1: Airport Security

Airports serve as hubs for global transportation, connecting millions of passengers to destinations worldwide. Ensuring the safety and security of both travelers and airport staff is a critical priority. To achieve this, airports utilize advanced physical access control systems that rely on cutting-edge technology and rigorous security measures. In this section, we explore in detail how these systems help safeguard airport facilities.

1. **Identity Verification:** Airport security begins with the verification of passengers' identities. This process ensures that only individuals with legitimate reasons and proper authorization can access secure areas of the airport. Identity verification can take various forms, including government-issued IDs, biometric scans, and other stringent checks (Khan and Efthymiou, 2021).

    - **Government-Issued IDs:** Passengers are generally required to present government-issued identification, such as passports or driver's licenses, to verify their identity. These documents contain key information, including the individual's name, photograph, date of birth, and nationality. The security staff at the airport validate these documents to confirm the traveler's identity.
    - **Biometric Scans:** Many modern airports have adopted biometric technologies to enhance identity verification. Biometrics involves the use of unique physical or behavioral traits, such as fingerprints, retina scans, or facial recognition, to verify an individual's identity. Biometric data is generally difficult to forge or duplicate, making it a more secure system of verification compared to traditional methods.

2. **Data Registry:** Identity verification relies on a comprehensive data registry, which stores information about passengers and their travel details. This registry is a central database that securely manages and organizes identity-related data, ensuring that only authorized individuals can access restricted areas. The data registry plays a crucial role in enhancing security and streamlining airport operations (Rubinger et al., 2023).

- **Registered Data:** The registry contains detailed information about each passenger, including their personal details, travel itinerary, and any special considerations or restrictions. This information is collected during the booking process and is securely stored for reference.
- **Watchlists:** The data registry is cross-referenced with watchlists that contain information about individuals of interest, such as those with criminal records or individuals on no-fly lists. This cross-referencing helps identify potential security threats and allows for immediate action when necessary (Koerber et al., 2020).

3. **Access Control Points:** Access control points are strategically placed throughout the airport to regulate entry into secured areas. These points are equipped with digital locks and keys, which are managed by the access control system. Access is granted or denied based on the results of identity verification and the individual's authorization (Mohamed et al., 2022).

    - **Digital Locks and Keys:** Digital locks and keys are sophisticated security mechanisms that rely on digital standards, encryption, and authentication protocols. When an individual's identity is verified and their authorization is confirmed, the access control system electronically unlocks the designated entry points, granting access. If there are any concerns about the individual's identity or authorization, access is denied (Aluri, 2020).
    - **Authorization Levels:** Different passengers and airport staff may have varying levels of authorization, granting access to specific areas depending on their roles and responsibilities. For example, airline crew members may have access to restricted areas like the tarmac and secure lounges, while passengers are limited to public areas until they clear security checkpoints.

Example 2: University Campus

A university campus consists of multiple buildings and facilities with varying access control needs. Students, faculty, and staff each have distinct identities within the access control system, granting them entry to specific classrooms, labs, and dormitories. The registry contains information about each individual's role and the spaces they are permitted to enter during specific time periods (Voon et al., 2016).

- **Card Access:** The system works by scanning access cards that contain unique information about the individual's identity. This allows the system to verify whether the person is authorized to access the building or specific areas based on the time and their credentials.

6.3 Access Spaces, Locations, and Doors

In the context of access control systems, space is a crucial element. It represents the physical areas within a building or facility where access control measures are implemented. These spaces

can range from individual rooms to entire floors or wings of a building. Locations within these spaces are further defined by specific access points, typically represented by doors.

Consider a commercial office building with multiple floors. Each floor may have its own access control policies, and within each floor, individual rooms are considered separate spaces. To manage access effectively, digital locks and keys are deployed at the doors leading to these spaces. Identity verification occurs at these access points, determining whether an individual is authorized to enter the space beyond the door.

6.3.1 Accessing a Building with Time and Conditions Considerations

Access control systems not only focus on verifying identity and regulating access to spaces but also take into account time and access conditions. Time plays a crucial role in determining when an individual is allowed to enter specific areas. The concept of "time-based access" defines the time window during which an individual's access privileges are valid.

For example, in a hospital, different staff members require access to various areas, such as patient wards, laboratories, and administrative offices. A nurse, for example, may have access to a specific ward only during their shift. Once their shift ends, their access rights to the ward are revoked, ensuring that only authorized personnel are present at all times (Woo et al., 2011).

6.4 Basic Model of Building Access

Before modeling the building access control process, the access control logic influences the monitored data by determining who is allowed or denied access during specific time intervals. we need to define authentication $Auth$ which verifies the identity of an individual and determines whether the provided credentials (access card, biometric data, etc.) are valid, and authorization $Allow$ which determines whether an authenticated individual has the right to access a specific resource (access zone) during a particular time. The results of $Auth$ and $Allow$ contribute to the monitored records, reflecting the security and operational status of the access control system.

6.4.1 Basic Model of Building Access

Our task is to formally model the key components of entity, characteristic, attribute, judgment etc, which we will do over the following subsections. We start this process by introducing the following notation. We denote the set of access cards by AC and the set of their characteristics by CC.

The characteristics include biometric data of the registered cardholder, access zone, and time intervals when access time is possible. Thus, CC contains data from the set CH of cardholders, the set BD of biometric data, the set AZ of access zones, and the set TI of time intervals, ie.,

$$CC \subset CH \cup BD \cup AZ \cup TI. \tag{6.1}$$

Recall, that for any set A, $P_{fin}(A)$ is the set of all finite subsets of A.
we formalize the authorization of people and the validity of their access as follows: Let $Auth$ be

a nonempty element of $P_{fin}(AC \times CC \times CH \times BD)$.
Let *Allow* be a nonempty element of $P_{fin}(CH \times CC \times AZ \times TI)$.

6.4.2 Time dependency of Auth and Allow

The state of the authentication and authorisation changes over time. To modify the *Auth* and *Allow* over time, we consider T as a discrete set counting clock cycles i.e., $T = \{0, 1, 2, ..., t, ...\}$.

In the following we introduce a mapping from T into $P_{fin}(AC \times CC \times CH \times BD)$, i.e.,

$$Auth : T \to P_{fin}(AC \times CC \times CH \times BD), \tag{6.2}$$

where for any $t \in T$, $Auth(t)$ is a subset from $P_{fin}(AC \times CC \times CH \times BD)$. Hence, at a given time t, for $(a, c, h, d) \in AC \times CC \times CH \times BD$ will be authenticated if

$$(a, c, h, d) \in Auth(t). \tag{6.3}$$

In the same way, we define *Allow* as follows:

$$Allow : T \to P_{fin}(CH \times CC \times AZ \times TI) \tag{6.4}$$

where at a given time t, for $(h, c, z, ti) \in CH \times CC \times AZ \times TI$ will be authorised if

$$(h, c, z, ti) \in Allow(t). \tag{6.5}$$

6.4.3 Entities

We define an entity to be a lock and we let *Lock* be the set of all locks to be monitored and let *Char* be the set of characteristics for these locks.
We define the *Behaviour of an entity* as a stream of data involved in opening a lock:

$$b : \mathbb{N} \to AC \times CC \times T, \tag{6.6}$$

where $\mathbb{N}$ is the set of natural numbers, AC is the set of all acceptable access cards, and CC is the set of all acceptable card characteristics. This model gives possible sequences of access attempts to a building in discrete time measured by T.

$$b(0), b(1), ..., b(n), ... \tag{6.7}$$

Specifically, the behaviour in the n^{th} access attempt to a building is given by:

$$b(n) = (a(n), c(n), t(n)) \tag{6.8}$$

where, $a(n)$ is the access card, $c(n)$ is the characteristic, and $t(n)$ is a time stamp in the n^{th} access attempts. Here we see a new kind of behaviour that contains its own real-world timestamps – recall our general discussion in subsection 4.2.1.

The behaviour of the entities is given by the following map:

$$[[-,-]] : Lock \times Char \to [\mathbb{N} \to AC \times CC \times T] \tag{6.9}$$

such that for lock $l \in Lock$, with characteristics $c \in Char$ for n^{th} access attempt $n \in \mathbb{N}$ belongs to the stream

$$[[l,c]] : \mathbb{N} \to AC \times CC \times T \tag{6.10}$$

and $[[l,c]](n) \; = \; n^{th}$ access attempt to unlock l with characteristics c.

6.4.4 Attributes

Attributes are properties of access behaviour where each access attempt will be compared with the data in the authenticate (authorised) lists which will be either verified (authorised) or not verified (not authorised).
In the following, we introduce the verification and authorization process by the following two functions. For $a \in AC$, $c \in CC$, $h \in CH$, $d \in BD$, $t \in T$, $z \in AZ$, $ti \in TI$, $Auth(t) \in P_{fin}(AC \times CC \times CH \times BD)$, and $Allow(t) \in P_{fin}(CH \times CC \times AZ \times TI)$:

$$Verfiy : AC \times CC \times CH \times BD \times T \times [T \to P_{fin}(AC \times CC \times CH \times BD)] \to \{verified, notverified\} \tag{6.11}$$

$$Verify(a,c,h,d,t,Auth) = \begin{cases} verified & \text{if } (a,c,h,d) \in Auth(t), \\ notverified & \text{if } (a,c,h,d) \notin Auth(t), \end{cases} \tag{6.12}$$

$$Verfiyz : CH \times CC \times AZ \times TI \times T \times [T \to P_{fin}(CH \times CC \times AZ \times TI)] \to \{authorized, notauthorized\} \tag{6.13}$$

$$Verifyz(h,c,z,ti,t,Allow) = \begin{cases} authorized & \text{if } (h,c,z,ti) \in Allow(t), \\ notauthorized & \text{if } (h,c,z,ti) \notin Allow(t). \end{cases} \tag{6.14}$$

According to the outcomes of the above two functions, the set of judgments is given as follows:

$$Judge = \{granted, denied\}, \tag{6.15}$$

where the judgment is granted if the outcome of (6.12) is $verified$ and the outcome of (6.14) is *authorized*, or *denied* otherwise. Therefore, the judgment process can be presented by the following function:

$$Access : AC \times CC \times BD \times CH \times AZ \times TI \times T \times [T \to P_{fin}(AC \times CC \times CH \times BD)] \times [T \to P_{fin}(CH \times CC \times AZ \times TI)] \to \{granted, denied\} \tag{6.16}$$

$$Access(a, c, d, h, z, ti, t, Auth, Allow) = \begin{cases} granted & \text{if } Verify(a, c, h, d, Auth) = verified \wedge \\ & Verifyz(h, c, z, ti, Allow) = authorized \\ denied & \text{if } otherwise \end{cases} \tag{6.17}$$

we think of $Auth$ and $Allow$ as parameters of attribute. Fix these parameters and rewrite $access$:

$$access_{(Auth,Allow)} : AC \times CC \times BD \times CH \times AZ \times TI \times T \to Judge \tag{6.18}$$

where:

$$access_{(Auth,Allow)}(a, c, d, h, z, ti, t) = \begin{cases} granted & \text{if } Verify(a, c, h, d, Auth) = verified \wedge \\ & Verifyz(h, c, z, ti, Allow) = authorized \\ denied & \text{if } otherwise \end{cases} \tag{6.19}$$

Thus, we take $Attr = \{access_{(Auth,Allow)}\}$.

6.4.5 Observation

From the previous section, we notice that the behaviour has an attribute that can be observed over time, which involves testing the access behaviour of opening a lock. The attribute is access attempts to a building through a lock with authentication and authorization.
To proceed, we define the following map

$$Obs_0 : Attr \times [\mathbb{N} \to AC \times CC \times BD \times CH \times AZ \times TI \times T] \times \mathbb{N} \to Judge \tag{6.20}$$

which, given an attribute $access_{(Auth,Allow)}$ and series of accessing $b \in [\mathbb{N} \to AC \times CC \times BD \times CH \times AZ \times TI \times T]$ for n^{th} attempts $n \in \mathbb{N}$:

$$Obs_0(access_{(Auth,Allow)}, b, n) = access_{(Auth,Allow)}(b(n)) \tag{6.21}$$

6.4.6 Monitoring

Monitoring is confined to the collection, evaluation, and recording of observational data about the behavior of entities. The outputs of a monitoring system are simply records of observations. First, let

$$Record = Lock \times Char \times Attr \times Judge \tag{6.22}$$

be the set of records.

We observe $Lock\ l$ and $Char\ c$ then $access_{(Auth,Allow)}$ is the attribute to be checked using

mappings 6.9 and 6.20, the monitor function becomes

$$Monitor : Lock \times Char \to [\mathbb{N} \to Record] \tag{6.23}$$

defined for $l \in Lock$ and $c \in Char$ by

$$Monitor(l, c)(n) = (l, c, access_{(Auth,Allow)}, Obs_0(access_{(Auth,Allow)}, [[l, c]], n)) \tag{6.24}$$

6.5 Summary

This chapter presented a formal model of access control systems, focusing on authentication and authorization processes by defining key components such as access cards, cardholder details, biometric data, and access time intervals. The model was applied in a case study on building access control, illustrating its practical use in monitoring entities, their attributes, and access decisions. It also discussed how digital locks and keys function in regulating access, supported by real-world examples such as airport security and university campuses. By integrating time dependency, the model captures dynamic changes in authentication and authorization states, ensuring structured access control based on predefined rules. The model, while comprehensive, is a theoretical framework that requires empirical validation through real-world testing. The case study showed its effectiveness in certain contexts, such as building access, but other domains might present unique challenges that require adaptations. The model's scalability and efficiency in large-scale implementations remain to be tested. Future work should focus on real-world applications to assess its effectiveness, alongside enhancing security measures through advanced encryption and biometric privacy protections. As biometric technologies evolve, continuous updates will be necessary to address emerging threats and ensure robust access control mechanisms.

Chapter 7

Conclusion

7.1 Summary of Work Done

This thesis has systematically explored the theoretical foundations of monitoring systems, focusing on their algebraic modeling and applications in access control. By extending existing formal models, this research introduced enhancements such as timestamps, notifications, and structured interventions. These refinements improved the theoretical framework's ability to analyze monitoring systems systematically and apply them to real-world security challenges.
The research set out with three primary objectives:

1. **To explore the commonalities and shared characteristics of diverse monitoring systems, theorize monitoring practices, and examine the role of monitoring in surveillance and privacy.** This objective was achieved by developing a general monitoring framework that formalizes key concepts such as observation, judgment, and intervention. A key contribution was the distinction between monitoring and surveillance, emphasizing ethical considerations in security applications.

2. **To develop new general mathematical models for monitoring processes that capture the essence of observing the behavior of individuals and objects within a given context in which time is important.** This was accomplished through the introduction of timestamps, allowing the model to record events in real-time. This refinement enhanced the precision of monitoring records, improving their applicability in security-sensitive environments.

3. **To model access control methods for digital services, emphasizing digital identities and protocols as integral components of monitoring systems.** The revised framework was applied to case studies in virtual and physical access control, particularly in login systems and building security. These case studies demonstrated how monitoring models could be effectively integrated into access control mechanisms, supporting authentication, authorization, and automated interventions.

7.2 Review of Aims and Objectives: Scope and Limitations

The primary aim of this research was to extend a formal theory of monitoring by enhancing response mechanisms through structured notifications, timestamps, and interventions. The numbered objectives were successfully addressed through the theoretical and applied components of the study. However, several limitations were identified:

- **Scalability and Complexity** – While the monitoring framework was successfully applied to specific security scenarios, its application to large-scale distributed systems remains an open challenge.
- **Discrete-Time Assumption** – The model assumes discrete monitoring events, which may not fully capture real-time monitoring processes in dynamic systems.
- **Balancing Security and Privacy** – The study acknowledges the trade-off between effective monitoring for security purposes and the ethical concerns related to user privacy.

7.3 Key Technical Challenges

Several technical challenges emerged throughout the research:

1. **Generalization vs. Specificity** – Ensuring that the model is broad enough to be applicable across different monitoring systems while retaining enough specificity for practical implementation.
2. **Dynamic Interventions** – Implementing interventions that adapt to varying security risks without introducing excessive complexity.
3. **Synchronization of Timestamps** – Managing time-sensitive events accurately in distributed systems.
4. **Security vs. Privacy Trade-offs** – Addressing concerns related to data collection, privacy regulations, and ethical monitoring practices.

7.4 Future Work

Building upon the outcomes of this research, several directions for future work have been identified:

- **Refining the Model for Real-Time Processing** – Extending the framework to handle real-time, continuous monitoring rather than discrete-event tracking.
- **Enhancing Security Mechanisms** – Integrating multi-factor authentication and AI-based anomaly detection into access control systems.
- **Expanding Use Cases** – Applying the model beyond access control to fields such as IoT security, cybersecurity threat detection, and healthcare monitoring.

- **Privacy-Preserving Monitoring** – Investigating techniques for privacy-aware monitoring that balance security needs with ethical concerns.

7.5 Final Remarks

This research has successfully developed an extended theoretical framework for monitoring, addressing the core objectives through formal modeling and applied case studies. While the thesis provides a structured approach to understanding monitoring mechanisms, the ever-evolving landscape of digital security necessitates further exploration. The findings lay a strong foundation for future advancements in monitoring systems, security modeling, and ethical interventions.

References

Agbo, I. S. (2015). Factors influencing the use of information and communication technology (ict) in teaching and learning computer studies in ohaukwu local government area of ebonyi state-nigeria. *Journal of Education and Practice*, 6:71–86.

Alexander, M. (2008). Survey, Surveillance, Monitoring and Recording. In *Management Planning for Nature Conservation*, pages 49–62. Springer Netherlands, Dordrecht.

Alfarsi, G. and Juma, M. (2018). Developing a Mobile Notification System for Al Buraimi University College Students. 1:10–16.

Almulaiki, W. A. (2023). The Impact of Performance Management on Employee Performance. *Saudi Journal of Business and Management Studies*, 8(02):22–27.

Aluri, D. C. (2020). Smart lock systems: An overview. *International Journal of Computer Applications*, 177:40–43.

Ampofo, L. (2011). Social Life of Real Time Monitoring. *Journal of Audience and Reception Studies*, 8(1):21–47.

Blazquez, D., Domenech, J., Gil, J. A., and Pont, A. (2019). Monitoring E-commerce Adoption from Online Data. *Knowledge and Information Systems*, 60(1):227–245.

Cândido, J., Aniche, M., and van Deursen, A. (2021). Log-based software monitoring: a systematic mapping study. *PeerJ Computer Science*, 7:e489.

Carzaniga, A., Rosenblum, D., and Wolf, A. (2000). Achieving Scalability and Expressiveness in an Internet-Scale Event Notification Service.

Cascio, W. and Montealegre, R. (2016). How Technology Is Changing Work and Organizations. *Annual Review of Organizational Psychology and Organizational Behavior*, 3:349–375.

Chen, J. and Hong, S., editors (2004). *Real-Time and Embedded Computing Systems and Applications, 9th International Conference, RTCSA 2003, Tainan, Taiwan, February 18-20, 2003. Revised Papers*, volume 2968 of *Lecture Notes in Computer Science*. Springer.

Culnan, M. and Williams, C. C. (2009). How Ethics Can Enhance Organizational Privacy: Lessons from the ChoicePoint and TJX Data Breaches. *MIS Quarterly*, 33(4):673–687.

da Silva, F. A. and Borsato, M. (2017). Organizational Performance and Indicators: Trends and Opportunities. *Procedia Manufacturing*, 11:1925–1932.

Dash, S., Shakyawar, S. K., Sharma, M., and Kaushik, S. (2019). Big data in healthcare: management, analysis and future prospects. *Journal of Big Data*, 6(1):54.

de Carné de Carnavalet, X. and Mannan, M. (2014). From very weak to very strong: Analyzing password-strength meters. In *Network and Distributed System Security Symposium*.

Debnath, L. and Basu, K. (2015). A short history of probability theory and its applications. *International Journal of Mathematical Education in Science and Technology*, 46(1):13–39.

Dodge, M. and Kitchin, R. (2009). Software, Objects, and Home Space. *Environment and Planning A*, 41:1344–1365.

Dwivedi, Y. K., Ismagilova, E., Hughes, D. L., Carlson, J., Filieri, R., Jacobson, J., Jain, V., Karjaluoto, H., Kefi, H., Krishen, A. S., Kumar, V., Rahman, M. M., Raman, R., Rauschnabel, P. A., Rowley, J., Salo, J., Tran, G. A., and Wang, Y. (2021). Setting the future of digital and social media marketing research: Perspectives and research propositions. *International Journal of Information Management*, 59:102168.

El-Masri, D., Petrillo, F., Guéhéneuc, Y.-G., Hamou-Lhadj, A., and Bouziane, A. (2020). A systematic literature review on automated log abstraction techniques. *Information and Software Technology*, 122:106276.

Esposito, C., Ciampi, M., and De Pietro, G. (2014). An event-based notification approach for the delivery of patient medical information. *Information Systems*, 39:22–44.

Gavilan, D. and Martinez, G. (2022). Exploring user's experience of push notifications: a grounded theory approach. *Qualitative Market Research: An International Journal*, 25.

Graham, S. and Wood, D. M. (2003). Digitizing surveillance: Categorization, space, inequality. *Critical Social Policy*, 23:227 – 248.

Greenwood, M. and Tao, L. (2020). Regulatory monitoring and university financial reporting quality: Agency and resource dependency perspectives. *Financial Accountability Management*, 37.

Gusev, M., Ristov, S., Velkoski, G., and Gushev, P. (2014). Alert notification as a service. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 319–324. IEEE.

Hansen, S. E. and Atkins, E. T. (1993). Automated system monitoring and notification with swatch. In *USENIX Seventh System Administration Conference (LISA 93)*, Monterey, CA. USENIX Association.

Hansen, T. and Vaudreuil, G. (2004). Rfc3798: Message disposition notification.

James G. Speight (2011). *Handbook of Offshore Oil and Gas Operations*. Gulf Professional Publishing, USA.

Johnson, K. and Tucker, J. V. (2013). The data type of spatial objects. *Formal Aspects of Computing*, 25(2):189–218.

Johnson, K., Tucker, J. V., and Wang, V. (2017). Theorising monitoring: Algebraic models of web monitoring in organisations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10644 LNCS:13–35.

Kapoor, K. K., Tamilmani, K., Rana, N. P., Patil, P., Dwivedi, Y. K., and Nerur, S. (2018). Advances in Social Media Research: Past, Present and Future. *Information Systems Frontiers*, 20(3):531–558.

Khan, N. and Efthymiou, M. (2021). The use of biometric technology at airports: The case of customs and border protection (cbp). *International Journal of Information Management Data Insights*, 1(2):100049.

Koerber, A., Starkey, J. C., Ardon-Dryer, K., Cummins, R. G., Eko, L., and Kee, K. F. (2020). A qualitative content analysis of watchlists vs safelists: How do they address the issue of predatory publishing? *The Journal of Academic Librarianship*, 46(6):102236.

Kumar, S., Tiwari, P., and Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, 6(1):111.

Kuzmenko, O., Bilan, Y., Bondarenko, E., Gavurova, B., and Yarovenko, H. (2023). Dynamic stability of the financial monitoring system: Intellectual analysis. *PLOS ONE*, 18(1):e0276533.

Laird, R. D., Zeringue, M. M., and Lambert, E. S. (2018). Negative reactions to monitoring: Do they undermine the ability of monitoring to protect adolescents? *Journal of Adolescence*, 63:75–84.

Lee, M. and Oh, S. K. (2011). Fast handover scheme using handover notification with no acknowledgement. In *2011 IEEE MTT-S International Microwave Workshop Series on Intelligent Radio for Future Personal Terminals*, pages 1–3.

Lewis, M. A. (2007). Charles babbage: Reclaiming an operations management pioneer. *Journal of Operations Management*, 25:248–259.

Liu, Z., Sampaio, P., Pishchulov, G., Mehandjiev, N., Cisneros-Cabrera, S., Schirrmann, A., Jiru, F., and Bnouhanna, N. (2022). The architectural design and implementation of a digital platform for Industry 4.0 SME collaboration. *Computers in Industry*, 138:103623.

Liu Wenyin, Weijia Jia, and Pui On Au (2002). Add exception notification mechanism to Web services. In *Fifth International Conference on Algorithms and Architectures for Parallel Processing, 2002. Proceedings.*, pages 483–488, Beijing, China. IEEE Comput. Soc.

Lombardo, J., Burkom, H., Elbert, Y., Magruder, S., Lewis, S., Loschen, W., Sari, J., Sniegoski, C., Wojcik, R., and Pavlin, J. (2003). A Systems Overview of the Electronic Surveillance System for the Early Notification of Community-Based Epidemics (ESSENCE II). *Journal of urban health : bulletin of the New York Academy of Medicine*, 80:i32–42.

Lukovenkov, S. G. (2020). "PANOPTIC VISIBILITY". SURVEILLANCE CULTURE OF 21TH CENTURY. *RSUH/RGGU Bulletin. Series Philosophy. Social Studies. Art Studies*, (2):10–19.

Lyon, D. (2006). *Theorizing Surveillance*. Willan.

Lyon, D. (2007). *Surveillance Studies: An Overview*. Polity Press.

Lyon, D. (2017). Surveillance Culture: Engagement, Exposure, and Ethics in Digital Modernity. *International Journal of Communication*, 11:1–18.

Malizia, A., Onorati, T., Diaz, P., Aedo, I., and Astorga-Paliza, F. (2010). SEMA4A: An ontology for emergency notification systems accessibility. *Expert Systems with Applications*, 37(4):3380–3391.

ManageEngine (2023). Different types of logs in SIEM and their log formats.

Mathew, T. K., Zubair, M., and Prasanna, T. (2023). *Blood Glucose Monitoring*. Treasure Island (FL): StatPearls Publishing, in: statpe edition.

Mohamed, A. K. Y. S., Auer, D., Hofer, D., and Küng, J. (2022). A systematic literature review for authorization and access control: Definitions, strategies and models. *International Journal of Web Information Systems*, 18(2/3):156–180.

Mohay, G. (2003). *Computer and Intrusion Forensics*. Artech House computer security series Computer and intrusion forensics. Artech House.

Mooloo, D. and Fowdur, T. P. (2013). An ssl-based client-oriented anti-spoofing email application. *2013 Africon*, pages 1–5.

Naite, I. (2021). Impact of Parental Involvement on Children's Academic Performance at Crescent International School, Bangkok, Thailand. *IOP Conference Series: Earth and Environmental Science*, 690(1):012064.

Nath, H. K. (2009). The Information Society. *A Journal of the SCTU*, 4:19–29.

Nations, U. (2017). *Principles and recommendations for population and housing censuses*. United Nations.

Nižetić, S., Šolić, P., López-de-Ipiña González-de-Artaza, D., and Patrono, L. (2020). Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*, 274:122877.

OECD (2017). Technology and innovation in the insurance sector.

Olan, F., Jayawickrama, U., Arakpogun, E., Suklan, J., and Liu, S. (2022). Fake news on Social Media: the Impact on Society. *Information Systems Frontiers*.

Olayiwola, A., Dare, O., Olumoye, M. Y., Ikedilo, O., Kolawole, F., Abiodun, S., Adeola, A., and Ugwu, N. (2021). Optimization of an identity access control system using biometric techniques. 27:647–653.

Ozili, P. K. (2018). Impact of digital finance on financial inclusion and stability. *Borsa Istanbul Review*, 18(4):329–340.

Parsell, C. (2015). Surveillance in supportive housing: Intrusion or autonomy? *Urban Studies*, 53.

Phuyal, S., Bista, D., and Bista, R. (2020). Challenges, Opportunities and Future Directions of Smart Manufacturing: A State of Art Review. *Sustainable Futures*, 2:100023.

Porter, M. E. and Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard Business Review*, 92:18.

Rubinger, L., Ekhtiari, S., Gazendam, A., and Bhandari, M. (2023). Registries: Big data, bigger problems? *Injury*, 54:S39–S42.

Samuelsson, L., Cocq, C., Gelfgren, S., and Enbom, J. (2023). *Everyday Life in the Culture of Surveillance*. Nordicom, University of Gothenburg.

Santhosh Kumar, B. and Sinha, S. (2021). Monitoring Login Shell Alert Notification Application. In *2021 5th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT)*, pages 235–237. IEEE.

Santos, J. L., Charleer, S., Parra, G., Klerkx, J., Duval, E., and Verbert, K. (2013). Evaluating the Use of Open Badges in an Open Learning Environment. In Hernández-Leo, D., Ley, T., Klamma, R., and Harrer, A., editors, *Scaling up Learning for Sustained Impact*, pages 314–327, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sarker, I. H. (2021). Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective. *SN Computer Science*, 2(5):377.

Shabalov, M., Zhukovskiy, Y., Buldysko, A. D., Gil, B., and Starshaia, V. V. (2021). The influence of technological changes in energy efficiency on the infrastructure deterioration in the energy sector. *Energy Reports*, 7:2664–2680.

Simonis, I. and Dibner, P. C. (2007). OpenGIS sensor planning service implementation specification. *Implementation specification OGC*, pages 1–21.

Sposaro, F. and Tyson, G. (2009). iFall: An android application for fall monitoring and response. In *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6119–6122, Minneapolis, MN, USA. IEEE.

Streefkerk, J. W., van Esch-Bussemakers, M., and Neerincx, M. (2007). Context-Aware Notification for Mobile Police Officers. In Harris, D., editor, *Engineering Psychology and Cognitive Ergonomics*, pages 436–445, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tuomi, I. (2018). The Impact of Artificial Intelligence on Learning, Teaching, and Education: Policies for the Future.

Van Dijck, J. (2014). Datafication, Dataism and Dataveillance: Big Data Between Scientific Paradigm and Ideology. *Surveillance Society*, 12:197–208.

Vianney, S., Prudence, D., and Nathan, N. (2020). Monitoring and Evaluation and Institutional Performance. *International Journal of Scientific and Research Publications (IJSRP)*, 10:367–377.

Vincent, C., Burnett, S., and Carthey, J. (2014). Safety measurement and monitoring in healthcare: A framework to guide clinical teams and healthcare organisations in maintaining safety. *BMJ quality safety*, 23(8).

Voon, M., Yeo, S., and Voon, N. (2016). Campus access control and management system. page 395–404.

Vrbaski, M., Bolic, M., and Majumdar, S. (2018). Complex Event Recognition Notification Methodology for Uncertain IoT Systems Based on Micro-Service Architecture. In *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 184–191. IEEE.

Wang, V. and Tucker, J. (2023). People Watching: Abstractions and Orthodoxies of Monitoring. *Technology in Society*.

Wang, V. and Tucker, J. V. (2017). Surveillance and identity: conceptual framework and formal models. *Journal of Cybersecurity*, 3(3):145–158.

Wang, Y., Rane, S., Draper, S. C., and Ishwar, P. (2016). Biometric security from an information-theoretical perspective. *Now Publishers*. Accessed via Now Publishers.

Woo, H., Lee, H. J., Kim, H.-C., Kang, K. J., and Seo, S. S. (2011). Hospital wireless local area network-based tracking system. *Healthcare Informatics Research*, 17(1):18–23.

Zhang, Y., Wu, C., and Wan, J. (2017). A human-in-the-loop wireless warning message notification model and its application in connected vehicle systems. *Journal of Intelligent Transportation Systems*, 21(2):148–166.

Zheng, K. and Liu, Y. (2022). Application of Mathematical Models in Economic Variable Input and Output Models under the Scientific Visualization. *Computational Intelligence and Neuroscience*, 2022:1–10.