UNIVERSITY OF NOTTINGHAM

Applying Reliability Engineering Techniques to the Process of Community Pharmacy Dispensing

by

Matthew Naybour

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Engineering
School of Civil Engineering

December 2019

Abstract

This thesis represents 3 years of research into applying engineering techniques to the community pharmacy dispensing process. The research has been undertaken with the long-term aims of improving the reliability and efficiency of community pharmacy dispensing.

A detailed Coloured Petri Net (CPN) model of the dispensing process is constructed through an iterative process of model design and improvement. The model includes coloured tokens which are used in the CPN to track the dispensing process at a high level of detail. The main novelty of the CPN model developed in this thesis is the ability to model the reliability and efficiency of a healthcare process in a single simulation-based model. Key model outputs related to pharmacy performance include the number of prescriptions dispensed, the number of dispensing errors, and the average waiting time.

Results from observations and interviews conducted at 4 UK community pharmacy sites are presented. Quantitative data was collected on the duration of individual stages of the dispensing process, and qualitative interviews about the practice were recorded with practitioners. This data collection represents a novel research contribution, to the field of pharmacy safety and efficiency, since previous work on timing individual stages of the dispensing process has not been carried out before at the same level of detail. The results of a distribution fitting analysis of the data are then used in the CPN model, when simulating a typical UK pharmacy.

A modern optimisation framework, Ant Colony Optimisation (ACO), is applied to the CPN model, to ascertain optimal pharmacy set-ups. While using the CPN within the optimisation framework, the CPN model is viewed as a discrete set-up problem, where a number of decision variables are set at discrete values to produce a single pharmacy set-up. Examples of decision variables include the number of dispensers and pharmacists to employ, the checking strategy to use, and the work patten staff should follow. The optimisation problem is to find the best values of these decision variables. The main aspects of novelty in this work are: the use of a three-stage heuristic process, and the combination of CPN and ACO frameworks to tackle a community pharmacy set-up problem. This framework can be used to aid decision makers by providing a Pareto front of non-dominated community pharmacy set-ups to choose from.

Acknowledgements

I would like to start by thanking both of my supervisors, Rasa Remenyte-Prescott and Matthew Boyd, for giving me the opportunity to study a PhD at the University of Nottingham. Special thanks go to Matthew, for his knowledge and expertice in navigating the University ethics approval process, and his facilitation of the on-site studies. Special thanks go to Rasa, for all the advice, guidance, and the occasional Monday morning grilling over the last 3 years. I couldn't have asked for a better team to work with.

Furthermore, I am greatly appreciateive towards the ESPRC for funding the project, and the University of Nottingham for providing all the equipment and resources used throughout my time at the University.

Much appreciation goes to all the pharmacy practitioners who took part in on-site studies and interviews, their contribution has allowed us to include a flavour of the real world within our model.

Thank you to everyone in the Resilience Engineering research group who made life in Nottingham a fond time to remeber.

To everyone who played our inventive version of Dirak at lunch, it was a pleasure.

Finally, this project would never have got off the ground without the support and love given to me by my parents. I love you both, and I hope to live up to the standard you have set for me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In the UK, community pharmacies are contractors of the National Health Service (NHS). They provide prescriptions to patients who have been prescribed a medicine regime by their doctor. The patient either brings a prescription form from the doctor's surgery and presents it to the community pharmacy, or an electronic version of the form can be sent directly from the doctor to the pharmacy using the Electronic Prescription Service (EPS) (Cornford et al., 2014). Once the prescription form is presented to the pharmacy, they collect the items listed on the form, print and apply adhesive labels with instructions for use to all items included in the prescription, and hand over the set of items to the patient.

To provide a good service to customers, pharmacies must provide the correct medication, labelled with suitable instructions for use, in a reasonable amount of time. Any errors made during the process can lead to bad outcomes for patients in terms of their health, and the time and resources spent dispensing the prescription are wasted. There are also the potential negative impacts on the vendor to consider. It is therefore important for pharmacies to deliver the correct medication to their patients with a high reliability.

Additionally, UK pharmacies operate in a competitive environment, where multiple providers are competing for patients to use their service (DOH, 2016). To succeed in this respect, they must satisfy patients' needs, and it is understood

that patient satisfaction with pharmacy services is closely linked to patient waiting times. Therefore, decision makers need to be able to understand how the choices they make about setting up their pharmacies can impact upon pharmacy efficiency.

Pharmacy decision makers are in the position of attempting to choose the best way to run their stores with respect to these two goals, whilst reducing their staff costs. If a tool could be developed to better inform these key decisions, by proposing optimal community pharmacy set-ups at various price points, this could potentially improve the reliability of the over 1 billion prescriptions dispensed each year in the UK (Prescribing and Medicines Team, 2017), and provide a higher level of customer satisfaction for patients using pharmacy services.

1.2 Research questions

The purpose of this research is to investigate the potential benefits of applying reliability engineering techniques to the community pharmacy dispensing process in terms of the reliability of the process, and to improve patient safety, and process efficiency, to improve the customer experience and pharmacy output.

The key research questions addressed by the thesis are:

- Can a model of the community pharmacy dispensing process be developed which is capable of testing and evaluating pharmacy set ups involving different work patterns, checking strategies, and staff configurations?
- If such a model can be developed, how well does it match up with practitioners understanding of the dispensing process, and how does it differ, and can it accurately reflect how the process is completed?
- Can the proposed method be used in tandem with an optimisation framework to aid decision makers answer key questions about how best to set-up pharmacies, with respect to: staff choice, checking strategies, and work patterns?

1.3 Summary of contributions

This thesis presents the development of a technology based on Coloured Petri Nets for modelling and analysing the process of community pharmacy dispensing. The novel contributions of the thesis are as follows:

1. A Coloured Petri Net (CPN) is developed, where the actions carried out by a team of practitioners are modelled. The model simulates a team of pharmacy practitioners, pharmacists and dispensers, completing a set of key dispensing and non dispensing activities. The novelty of the CPN within the healthcare modelling field is that the model considers both the reliability and efficiency of the community pharmacies' activities within a single model.

2. The findings of 4 on-site studies carried out on UK community pharmacy sites are presented as a part of this research. The findings include: probability distributions derived from timing 6 key stages of the dispensing process, a qualitative analysis of interviews carried out with practitioners from each of the sites. The probability distributions derived from timing 3520 individual stages of the dispensing process are fed back into the CPN model, and the interviews highlight some strengths and limitations of the modelling framework.

3. The CPN is combined with an Ant Colony Optimisation (ACO), and optimal community pharmacy set ups are found. This tool has the potential to aid decision makers about the best way to staff community pharmacies under different price constraints.

1.4 Publications

Through the completion of this thesis, four publications have been submitted and accepted to international journals and conferences. An additional journal paper has been submitted to the Journal of Operational Research, and is currently undergoing the review process. Similarly, a conference paper has been submitted to the 2020 Reliability and Maintainability Symposium (RAMS) conference, and is undergoing their review process. The accepted papers are as follows:

1. Matthew Naybour, Rasa Remenyte-Prescott, and Matthew Boyd. Reliability modelling of dispensing processes in community pharmacy. In 10th IMA International Conference on Modelling in Industrial Maintenance and Reliability (MIMAR), June 13-15, 2018.

2. Matthew Naybour, Rasa Remenyte-Prescott, and Matthew Boyd. Evaluation of a community pharmacy dispensing process using a coloured petri net. In Proceedings of the European Safety and Reliability Conference, ESREL, Trondheim, Norway, 2018.

3. Matthew Naybour, Rasa Remenyte-Prescott, and Matthew J. Boyd. Reliability and efficiency evaluation of a community pharmacy dispensing process using a coloured Petri-net approach. Reliability Engineering & System Safety, 182:258-268, February 2019.

4. Matthew Naybour, Rasa Remenyte-Prescott, and Matthew Boyd. Ant colony optimisation of a community pharmacy simulator using in field quantitative data. Paper accepted for Proceedings of the European Safety and Reliability Conference, ESREL, Hamburg, Germany, June 17-21, 2019.

5. Matthew Naybour, Rasa Remenyte-Prescott, and Matthew Boyd. Ant colony optimisation of a community pharmacy dispensing process using Coloured Petri-Net simulation and UK pharmacy in-field data. Submitted to the Journal of the Operational Research Society.

6. Matthew Naybour, Rasa Remenyte-Prescott, and Matthew Boyd. Modelling Reliability and Efficiency of Community Pharmacy Processes. Abstract Accepted on May 2019, for the Preceedings of the Reliability and Maintainability Symposium, RAMS, Palm Springs, USA, 2020.

1.5 Thesis outline

The remaining chapters of this thesis are as follows. Chapter 2 outlines the required background theory of reliability engineering, introduces a set of statistical techniques and distributions used later in the thesis, and presents a theory of the dispensing process. Chapter 3 shows a review of the literature of pharmacy analysis projects in 4 parts: a background review of the literature of safety studies carried out in community pharmacies, a review of studies related to pharmacies'

efficiency requirements, an overview of reliability methods being applied to healthcare processes, and detailed reviews of contemporary work applying reliability engineering techniques to the community pharmacy dispensing process (or closely related healthcare fields). Chapter 4 shows work completed at an early stage of the research where two classical reliability engineering techniques are applied to the dispensing process, as a feasibility test for each of the respective techniques. Chapter 5 reports the main findings of a study carried out in 4 UK community pharmacies. Chapter 6 shows the development stages of a Coloured Petri Net (CPN) model designed to simulate the process of community pharmacy dispensing. Chapter 7 shows how an Ant Colony Optimisation (ACO) algorithm was developed to optimise the 4 set-up decision variables of the CPN model. Chapter 8 concludes the thesis by outlining the novel contributions made, and making suggestions for Future work. Full descriptions of 4 on-site profiles, an example copy of raw timing data, an ethics approval letter, and an example of coding implementation are included in the appendices.

Chapter 2

Background: Modelling and Statistical Techniques, and the Community Pharmacy Dispensing Process

2.1 Introduction

Reliability engineering involves using scientific knowledge to ensure that a system performs it's desired function for a required duration of time. The application of reliability engineering techniques has yielded improvements in safety in a range of high risk industries (such as nuclear, aviation, and space) (Garrick, 1988). On this basis, it is proposed that it might be possible to find similar gains in reliability by using reliability engineering techniques to the healthcare sector (Reason, 2000). The main challenge of this approach will be dealing with the inherent differences between applying modelling techniques to the human based field of community pharmacy dispensing, and more mechanistic processes. This Chapter of the thesis introduces the background theory of Reliability Engineering, some core statistical methods, and a formulation of how prescriptions are being dispensed in community pharmacies. These are all used to model community pharmacy dispensing in this thesis. Section 2.2 focuses on accurately describing how a team of practitioners work together to dispense prescriptions in community pharmacies, the types of non-dispensing activities that take place, and the different staff roles seen

in practice. Section 2.3 presents the core concepts of reliability theory, followed by an overview of three risk assessment modelling techniques, with an in depth review of the Coloured Petri net modelling extension. This is followed by Section 2.4, where key statistical methods, such as QQ plots, the method of maximum likelihood estimation, and others, are introduced.

2.2 Community pharmacy

2.2.1 Introduction

To build an accurate model of the dispensing process, a full and deep understanding of the types of tasks completed in pharmacies, the types of roles available to practitioners, and how teams of practitioners work together to dispense medication in practice is presented in this Chapter.

2.2.2 The dispensing process

The full process of dispensing a prescription can be described by seven key stages shown in the list below (Langley, 2009) (NPSA, 2007) (Waterfield, 2008).

1. Prescription reception
2. Legal and clinical check
3. Label generation
4. Picking
5. Label Application
6. Final accuracy check
7. Hand over to patient or store for delivery

These stages, completed in this order in the UK, are the basis of how community pharmacies dispense medication to the public.

2.2.2.1 Prescription Reception

When a patient presents a prescription at a community pharmacy, a member of staff must be available to take the prescription off the customer and complete a set of administrative tasks before the process can begin (NPSA, 2007).

The administrative tasks completed at this stage include: taking a payment (if necessary), accessing the Patient Medication Record (PMR) to update or confirm patient details, and checking if the item can be bought over the counter for less than the price of a prescription (Waterfield, 2008). Whoever initially receives prescriptions from customers is also responsible for managing customer expectations by making a judgement of how long they will have to wait for their medication (Waterfield, 2008).

This set of tasks is usually carried out by a dedicated counter assistant, however, in smaller pharmacies with fewer staff, it can be a dispenser or pharmacist receiving prescriptions.

2.2.2.2 Legal and clinical check

All prescriptions need to be legal. If a prescription form is deemed to be legally incomplete, then the pharmacy will not be able to dispense against it (Langley, 2009). Sometimes the person who received the prescription may have spotted an error with the prescription form, in this case, it is best practice for the member of staff who found the error to communicate and confirm it with the pharmacist, before engaging a patient (Waterfield, 2008).

In addition to fulfilling the legal requirements, prescriptions must also be considered to be clinically appropriate if they are to be dispensed (Langley, 2009). If a pharmacist believes that a prescription is unsafe, or constitutes inappropriate treatment for a patient, they won't dispense the prescription. A clinical check is performed by the pharmacist to ensure it is clinically suitable. Pharmacists consider the health of the patient, the dose of medication, and any interactions with other drugs the patient may be taking (Langley, 2009). The PMR can aid the check by highlighting drug interactions with medicines the pharmacy staff may be unaware the patient is taking.

The National Patient Safety Agency (NPSA) recommended that the clinical check should take place at the start of the dispensing process. Performing it early facilitates a conversation which allows any queries to be identified and dealt with quickly (NPSA, 2007). In addition, having a temporal separation between the clinical and accuracy check is thought to be beneficial to patient safety. It should be noted, however, that since a patient safety advocacy group made the recommendation, it is potentially not currently a universal practice.

2.2.2.3 Label generation

It is considered good practice in the NHS to produce labels for each prescription before locating medicines (Langley, 2009). Using this order, labels can be applied to medicines as soon as they have been fetched, reducing the risk of labels being applied to the wrong items.

The production of labels is done using computers, however, this is not a fully automated process since the user still has to select what to print. The Royal Pharmaceutical Society (RPS) provides a summary of information required for labels (RPS, 2011). Full legal documentation of the labelling requirements can be found in the medicines act 1968 (Government, 1968), and below is the full list of required information:

1. Patients name
2. Address of the pharmacy
3. Date the medicines were dispensed
4. Drug name
5. Instructions for use
6. Quantity of medicine
7. Concentration/strength
8. Any cautions
9. Any extra instructions

Labelling requires focus, therefore, labelling stations should be located in a section of the pharmacy which is free from distractions (Waterfield, 2008). Labels should be generated for *only* one prescription at a time, using baskets to keep prescriptions together, and using prescription holding clips to ensure that the prescription form is at eye level when typing dispensing labels (NPSA, 2007).

The text included on labels is also subject to safety requirements. Active verbs are preferred over passive verbs on labels. For example, *"Take one"* is preferred over *"One to be taken"*. Adjacent numbers are also considered to be potentially confusing, so where possible they should be avoided. For example, *"Take three four times a day"* can be changed to read *"Take three doses four times a day"* (Langley, 2009).

Considering the recommendations for safe labelling practice, there is a diverse range of possibilities for a label to be considered incorrect, and the consequences can be variable. For example, a label instructing someone to take twice the intended amount of medication could prove to be critical, while on the other hand, there are many comparatively inconsequential errors, such as misspelling names or addresses.

Maximising the reduction in harm done to patients may include an effort to focus on the types of errors that are likely to have the worst consequences. While maximising the efficiency of community pharmacies may instead focus on eliminating the more frequent, but less severe types of error.

2.2.2.4 Picking

Following label generation, the contents of a prescription can be picked (Langley, 2009). Picking is the action of dispensers collecting the required medicine from the pharmacy's stocks. The size of a pharmacy can affect how long this part of the process takes. A larger pharmacy may take longer to walk around, whereas a smaller pharmacy may become congested if many people are working in a small area (Waterfield, 2008). In terms of safety while dispensing, the literature is clear that items should always be selected using the original prescription form, not the labels generated earlier (Waterfield, 2008) (NPSA, 2007). This avoids the propagation of errors that were made while generating the labels.

Finally, there are many medicines that can be prescribed on the NHS and it is vital that community pharmacies ensure that the correct item is being supplied. However, it is not always the case that there is only one medication that can satisfy the requirements of a given prescription. In a community pharmacy, if a proprietary name is prescribed for a drug then the proprietary product must be supplied. But if a drug is prescribed by its generic name, then in most cases any equivalent drug can be supplied (Langley, 2009). Here 'proprietary' means the name of a brand of medicine, given by the manufacturer. For example, Phenoxymethylpenicillin is a generic name for a chemical that can be used as a medicine, while Nadopen V is a specific brand of a type of medicine containing Phenoxymethylpenicillin.

2.2.2.5 Label application

Once picked, the labels can finally be applied to medicines. However, before finally attaching a label to a medicine box, the dispenser performing the assembly should check for a correspondence between the labels, FP10 form, and the items being labelled. This acts as an intermediate accuracy check, which helps to reduce the number of errors found at the final accuracy check (Waterfield, 2008).

2.2.2.6 Final accuracy check

The final accuracy check is the final fail-safe of the dispensing process. The check is designed to make sure that the labels and items being handed out all match what was prescribed on the prescription form. This task may *only* be performed by either a pharmacist or an ACT (Accredited Checking Technician). Any errors that have occurred are to be identified and then rectified at this stage, they should then be recorded as a "near miss" for use in future analysis of the failure (Waterfield, 2008). Meanwhile, any failures that go unspotted at the final check are likely to reach the patient, these events are classified as dispensing errors. This stage, along with the intermediate accuracy check, are presented as the main safety features of the dispensing process.

The final check should be carried out in a specific checking area of the pharmacy, and ideally should be completed by someone who was not involved in assembling the prescription (Waterfield, 2008). This is not an essential requirement, as there are cases of pharmacists who work alone, but if possible it is recommended. If

not possible, taking a short mental break between assembling and checking is recommended. After completing the check, prescriptions are usually packaged into bags. It is then considered good practice to attach the prescription forms to the bags of dispensed medicines, for later use when handing the prescriptions out.

2.2.2.7 Hand over to patient or store for delivery

If a completed prescription is going to be picked up later, rather than handed over to the patient immediately, there needs to be a process for storing the dispensed medicine. This can apply for all types of delivery, EPS (Electronic Prescription Service) or call back prescriptions. Recommendations for safe medicine storage include keeping prescriptions attached to the bag of dispensed medicines, storing medicines in alphabetical order, and have a separate place for items requiring amendments (NPSA, 2007). Example amendments could be; a patient returning an incomplete prescription, or a prescriber being asked for confirmation about a prescriptions suitability.

Finally, the completed prescription is handed over to the patient. Although this sounds simple enough, there can be errors that occur at this stage. It is possible to give a prescription to the wrong recipient by accident, this can happen when people miss-hear their name being called out. A recommended way to stop this is for the person handing out prescriptions to ask patients to confirm a personal detail, an address or their date of birth, before handing over a prescription (Waterfield, 2008) (Langley, 2009). These details should be readily available from the prescription form attached to the bag of medication.

While handing out medication, pharmacists offer advice for patients on how to use the medication in a safe and effective manner (Langley, 2009). This can take the form of a verbal explanation, or in some cases, a physical demonstration of how a medical device is used.

2.2.3 Non-dispensing activities

Where traditionally the main duties of pharmacists were restricted to the dispensing of medications, it has become common for them to perform other non

dispensing services (Pande et al., 2013). These activities can range from simple housework and tidying up, to ensuring that patients are taking medications correctly, or delivering flu vaccinations (PSNC, 2017). Models of community pharmacies should consider taking such activities into account, since work sampling studies have demonstrated that pharmacists spend a significant portion of their time on non-dispensing activities (Davies et al., 2014).

Work sampling studies provide further more detailed insight into what community pharmacists actually do throughout their days (Davies et al., 2014) (Bell et al., 1999) (McCann et al., 2009). These observational studies proceed by recording at random or fixed intervals of time, what type of tasks pharmacists are completing (Emmerton and Jefferson, 1996). The coding frameworks used in such studies contain many fields which are generally unrelated to dispensing medicines. What follows is a brief summary of the non dispensing activities listed in the coding framework of the most recently available work sampling study of community pharmacies in the UK (Davies et al., 2014). The most recent study was chosen since it may be more representative of modern practice.

1. Professional encounters with non-patients. This includes talking to drug company salespeople, contacting the Prescription Pricing Authority, etc.

2. Health related communication. This includes any communication related to improving health outcomes which is outside the field of direct patient counselling. Examples could include talking to GP's or nurses.

3. Provision of advanced services, enhanced or other national health services. These include the 7 advanced pharmacy services: Medicine Use Reviews (MUR) and Prescription Intervention (PI) services, Stoma Appliance Customisation service, Appliance Use Review, New Medicine Service (NMS), Flu Vaccination Service, and the Urgent Medicine Supply Advanced Service (UMSAS) (PSNC, 2017).

4. Provision of private enhanced services. Any private services, such as paid for vaccinations.

5. Inventory and stock control. Maintaining the inventory of both prescription and non-prescription medications, as well as non-medicinal products.

6. Staff training and education.

7. House keeping. General maintenance and cleaning of the pharmacy.
8. Sales transactions. The selling of non-health related products.
9. Money and managerial administration.
10. Rest, waiting and personal time.
11. Non-professional encounters. This relates to general chat with friends, or customers which is not directly related to health provision.

In the study (Davies et al., 2014), non-dispensing activities were recorded to take up 35.5% of pharmacists time. Non-dispensing activities were sorted into 4 broad categories of, non-counselling communication, rest, premises and services, which were groupings of the non-dispensing activities listed above. Non-counselling communication took up 15.1% (1+2+6+11, in the list above), rest or personal time accounted for 8.6% (10), as well as maintaining the premises (5+7+8+9), and advanced and enhanced services (3+4) only took up 3.2% of pharmacist's time. To effectively model the dispensing process there must be some consideration of how practitioners devote time to non-dispensing activities. Section 6.4.1, shows how 3 non-dispensing tasks are considered within the CPN model developed in Chapter 6.

2.2.4 Staff roles

Most pharmacies employ a mixture of staff with different levels of qualification, training, and experience, and this mix of staff is commonly referred to as the 'skill mix' in the pharmacy literature (Schafheutle et al., 2008). As well as pharmacists, pharmacies employ non-pharmacist support staff, of which there are four recognised non-pharmacist roles in the UK (PSNC, 2017). These pharmacy support staff are employed to help deliver primary care in pharmacies, and in some cases they are the only contact a patient receives during a visit (Schafheutle et al., 2008). The four roles and a brief description of each are given below (PSNC, 2017):

1. Medicines counter assistant (MCA). MCAs receive prescriptions and support the retail function of a pharmacy, as well as providing general healthy lifestyle support to customers.

2. Dispensing/pharmacy assistant (DA). DAs aid the pharmacist with dispensing prescriptions, and the upkeep of the dispensing stock. They can also perform the role of an MCA if required.

3. Pharmacy technician (PT). PTs support the pharmacist in managing the dispensary and dispensing prescriptions. They have a higher level of training and certification than DAs or MCAs.

4. Accredited checking technician (ACT). ACTs undertake additional training which allows them to accuracy check dispensed prescriptions, although a pharmacist must still perform a clinical check.

5. Pharmacist. Every pharmacy is required to operate under the supervision of a 'responsible pharmacist', who is responsible for the safe running of the dispensary. They perform a number of daily duties, some of which are listed below:

    - Clinically and accuracy checking prescriptions
    - Offer advice about medicines and treatments
    - Advanced services
    - Directing patients to other services

Pharmacies tend to include a mix of staff roles. A 2008 study of support staff in community pharmacy (Schafheutle et al., 2008) found that of 25 participating pharmacies, every respondent employed an MCA. Furthermore, 80% of respondents employed one or more DA, while the majority of participants employed 1 (44% of respondents) or 0 (40% of respondents) PTs. The role of ACT was not included in the study.

A 2012 study (Braund et al., 2012) also found a mixture of staff roles working in community pharmacy. This study asked people working in community pharmacies to identify what their roles and responsibilities within the team were. MCAs identified their role as handling stock, serving customers and liaising with product representatives. PTs saw their main responsibilities as dispensing, compounding, handing out medicines, offering advice, and dispensing rest home medications. Pharmacists viewed their most important role as providing the final check on dispensed prescriptions, and ensuring the correct person receives it. They also

identified their contribution to dispensing, counselling, medicine sales, and further managerial duties.

Attempts to model the community pharmacy work environment should include the different roles and responsibilities given to members of staff. This need is addressed in Chapter 6 of the thesis when a Coloured Petri Net model is built with dispenser and pharmacist tokens.

2.2.5 Summary

The dispensing process has been described in this section, by breaking it down into the seven main stages of dispensing. Each stage has been described, along with a consideration of the elements of safety within the process. A set of non-dispensing tasks, reported to take up 35.5% of pharmacists time, completed in community pharmacies have been outlined, as well as a description of each of the staff roles available to community pharmacy practitioners.

2.3 Reliability Engineering

2.3.1 Introduction

All processes are exposed to some level of risk, from trips, slips, and falls, to fires or floods. Whereas it doesn't take much for an average person to fall, failures of more complex systems are usually arrived at after a series of events combine to produce a large scale failure (Ostrom and Wilhelmsen, 2012). A systematic approach to risk assessment can alleviate some of these risks, and highlight critical components, vulnerabilities, or risky operating conditions in complex systems.

When failure events occur, they inflict damage to the parent company (Andrews and Moss., 2002). In high risk industries these can be in the form of; reduction in profits, introduction of new safety regulations, injuries, or even deaths. In the context of community pharmacy, a pharmacy that makes many errors could lose business as the public learn about the regular occurrence of errors. The negative outcomes of actions create incentives to prevent similar failures from re-occurring. As well as preventing the re-occurrence of known failures, reliability engineering

also aims to adopt an effective forward facing attitude to risks that may not have happened before (Hollnagel et al., 2006).

2.3.2 Risk and consequences

Consider placing a bet at a roulette table. The action of placing a bet at the table is a risk, which may have positive or negative consequences depending on the outcome. Actions with potential negative outcomes are often referred to as risks, whereas actions with potential positive outcomes are more commonly seen as opportunities (Ostrom and Wilhelmsen, 2012).

Risk, for the purpose of risk assessment, can be defined as the probability of a negative event occurring. Or alternatively, as the product of the probability of a failure event occurring, P, and the value of the negative consequences caused, C (Andrews and Moss., 2002). This second definition of risk can be thought of as the expected loss of a given failure, and it is expressed in Equation 2.1.

$$Risk = C \cdot P \tag{2.1}$$

where Risk, is expressed in terms of C, the cost associated with a negative event, which occurs with probability, P. Using Equation 2.1, it is clear that risk can be lessened by reducing the probability of occurrence, or by reducing the consequences if a failure was to occur (Andrews and Moss., 2002). As an example, consider trying to reduce the risks associated with the road network. Lowering the probability of car accidents occurring would reduce the risk. This could be achieved by lowering the speed limit, or by using autonomous cars less prone to accidents (Chan, 2017). Alternatively, one could aim to reduce the consequences of crashes by improving the safety standards of cars by installing extra air bags, or stronger crash bumpers.

2.3.3 Reliability

The usual engineering definition of reliability, denoted as R, is, the probability that an item will perform a required function without failure under stated conditions for a stated period of time (Andrews and Moss., 2002) (O'Connor and Kleyner,

2012). Alternatively, R can be expressed as the number of failures over a given time period (O'Connor and Kleyner, 2012).

When applying these definitions of reliability to the process of community pharmacy dispensing, it became clear that developing an alternative definition of reliability, related to the literature of errors in community pharmacies, is needed.

This thesis will use the definition of reliability given below.

Definition 2.1. The reliability of the dispensing process, R, is defined as the proportion of dispensed prescriptions which are correct, such that, $R = p_{correct}/p_{total}$, where $p_{correct}$ is the number of correct prescriptions dispensed, and p_{total} is the total number of prescriptions dispensed, over a given time period.

2.3.4 Efficiency

The efficiency of a process is usually calculated through a comparison between useful outputs and the required resources to sustain the production. It is known that patient satisfaction with community pharmacies in closely linked to waiting times. In this thesis, a set of indicators are used to measure a pharmacies performance in terms of efficiency, the indicators used are as follows:

1. The number of prescriptions dispensed in a given period
2. The average waiting time
3. The number of delayed prescriptions, taking longer than 15 minutes to dispense
4. The number of advanced services (examples include: the morning after pill, malaria vaccination, or medicine use reviews) completed

2.3.5 Fault Tree Analysis

Fault tree analysis (FTA) is a classical technique for analysing system reliability. The goal is to identify some key undesirable states of a given system, and then find all the reasonable ways in which the system can reach those states through a combination of lower level events, which eventually includes component failures.

Deductive reasoning is used in a logical way to build up the combinations of more fundamental events that lead to the overall undesirable system state. The method has been used extensively in high hazard industries. Aerospace, nuclear and petrochemical sectors are some typical examples (Garrick, 1988).

To begin a fault tree analysis of a given system, the system's modes of successful operation and failure must be understood. An undesirable top event is chosen as the starting point, and a deductive analysis proceeds by determining how the undesirable event can be caused by the occurrence of individual, or combinations of, lower level failures (O'Connor and Kleyner, 2012). Top events should be chosen carefully, since it is possible to include multiple failure modes within a single top event. Example top events for the community pharmacy dispensing process are proposed as follows:

1. A patient receives a prescription which contains incorrect items
2. A prescription is handed out to the wrong patient
3. A label on an item has incorrect instructions for use

Fault trees for top events 1 and 2 above, are developed in Section 4.2. Fault trees are made up of two elementary objects, gates and events (Andrews and Moss., 2002). A basic set of elements used in FTA, with a brief description of each, is shown in Figure 2.1.

Lower level events are produced by working logically backwards from the chosen top event, by determining the necessary and sufficient conditions for the top event to occur. These lower events are connected through gates to higher level events and are continually redefined in terms of more basic events, until basic component failures or the desired level of detail is reached (Andrews and Moss., 2002). The level of detail at which the analysis stops is called the resolution of the FTA.

To illustrate how lower level events are related to higher level events using gates, an example fault tree is given in Figure 2.2. An event positioned above an AND gate, requires all the events connected below the gate to occur for the event itself to occur. An event above an OR gate, only requires a single event connected below the gate to occur for the event above the gate to occur (Vesely et al., 2002). The fault tree in Figure 2.2 says that if any one of the events A, B or C occurs, and

FIGURE 2.1: Elements of FTA

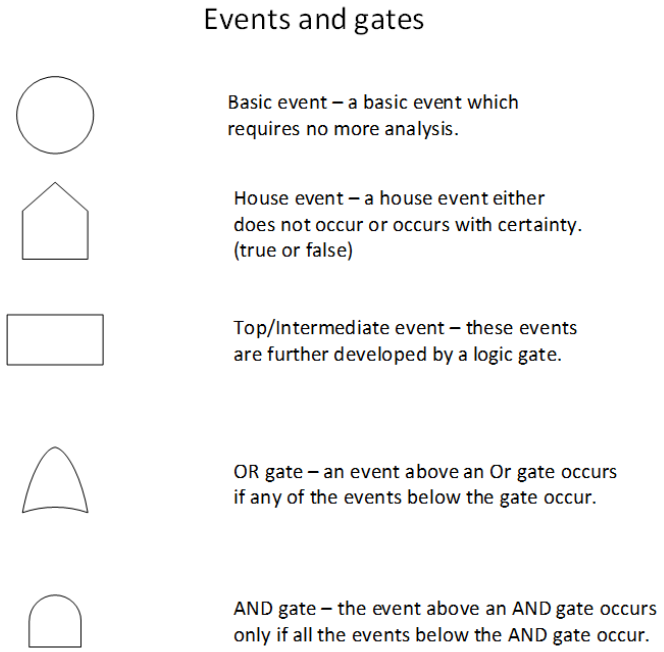
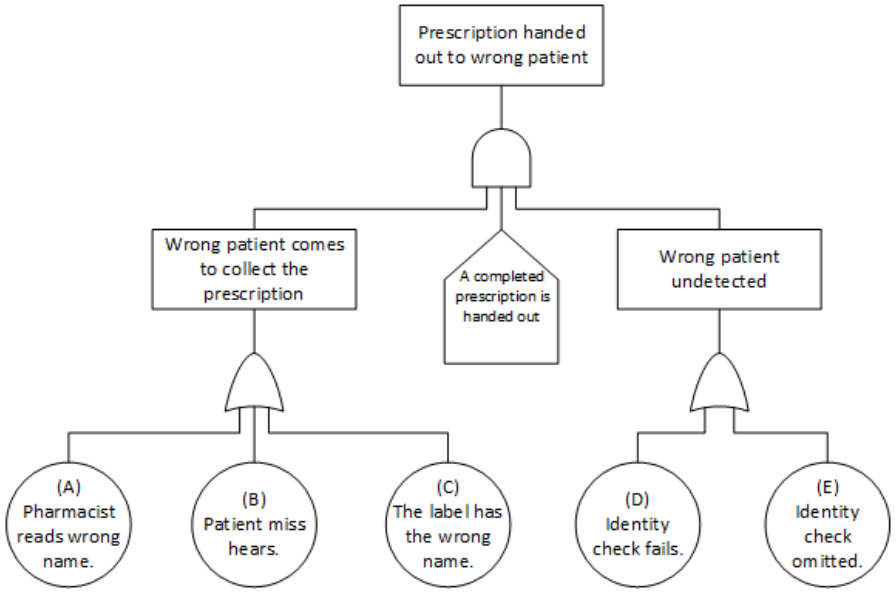


FIGURE 2.2: Example community pharmacy fault tree



one of either D or E occurs, then a prescription will be given to the wrong patient when the pharmacist is handing out the prescription.

A completed fault tree diagram graphically represents the Boolean logical relationship between the top undesirable event, and lower basic events (Andrews and Moss., 2002). Each fault tree considers a single undesirable top event, and maps out how individual failures can combine to produce the top event. Since only a

single top event is considered in each fault tree, multiple fault trees may need to be constructed for a single system, to account for varying failure modes.

When constructing a fault tree, it is important to consider the no miracles principle (Andrews and Moss., 2002) (Vesely et al., 2002) (Ostrom and Wilhelmsen, 2012). This principle states to ignore cases in which a component miraculously fails in such a way that a system failure is prevented. For example, imagine a pharmacy gives out a prescription to a person different to the intended recipient. There is a small chance that the unintended recipient of the prescription happens to need exactly the same medicine as the person that was supposed to be receiving the prescription. In practice this situation would lead to each patient receiving the correct medication, however, a credible FTA should not consider the possibility of such lucky coincidences.

Qualitative evaluation of fault trees is done by expressing the fault tree using Boolean logic and finding the minimal cut sets. A minimal cut set is defined as the smallest group of basic events, such that if all events in the cut set occur, the top event will occur. The set is minimal in the sense that if any of the basic events included in the set were to be removed, then the set would no longer be a cut set (Vesely et al., 2002). The cut sets for the fault tree in Figure 2.2 are as follows: AD, AE, BD, BE, CD, and CE. Spotting the minimal cutset is simple for small trees like the one in Figure 2.2, however, larger trees may require computer analysis to find all the minimal cut sets (Ostrom and Wilhelmsen, 2012).

2.3.6 Quantitative analysis of fault trees

Quantitative analysis of fault trees consists of two main tasks (Vesely et al., 2002):

1. Calculating top event probabilities or frequencies
2. Finding the relative importance of basic events, in terms of their contribution to top event occurrence.

The focus of quantifying fault trees is to estimate the probability of the top event occurrence, although the probability of any lower events in the tree occurring can also be calculated. Data (or estimates) for the probability of occurrence for all

the basic events included in a fault tree are needed to perform a full quantitative analysis.

To demonstrate how top event probabilities are calculated, imagine a fault tree with n non-repeated basic events (B_1, B_2, ... , B_n), with k minimal cutsets, (C_1, C_2, ... , C_k). Then equation 2.2 shows the probability of the top event occurring.

$$P(\text{Top event occurs}) = P(C_1 \cup C_2 \cup ... \cup C_k) = P(\cup_{i=1}^{k} C_i) \tag{2.2}$$

The inclusion-exclusion expansion principle must be used to determine the exact probability. For a general set of k events, the inclusion-exclusion expansion principle gives:

$$\mathbb{P}\left(\bigcup_{i=1}^{k} C_i\right) = \sum_{i=1}^{k} \mathbb{P}(C_i) - \sum_{i<j} \mathbb{P}(C_i \cap C_j) + \sum_{i<j<l} \mathbb{P}(C_i \cap C_j \cap C_l) - \ \cdots \ + (-1)^{k+1} \mathbb{P}\left(\bigcap_{i=1}^{k} C_i\right) \tag{2.3}$$

This calculation can be cumbersome for large numbers of cutsets, and binary decision diagrams can be used as an alternative method to attain exact calculations for top event probability (Reay and Andrews, 2002). Alternatively, upper bounds and approximations can be used to calculate the maximum value, that the probability of top event occurrence, can take. The minimal cutset upper bound on the top event probability is shown in Equation 2.4:

$$P(\text{Top event occurs}) \leq 1 - \prod_{i=1}^{k} (1 - P(C_i)) \tag{2.4}$$

The rare event approximation is a less tight upper bound on top event probability than the minimal cutset upper bound, it is shown in Equation 2.5.

$$\mathbb{P}(\text{Top event occurs}) \leq \text{Minimal Cutset Upper Bound} \leq \sum_{i=1}^{k} \mathbb{P}(C_i) \tag{2.5}$$

The rare event approximation is used to calculate top event probabilities. It assumes that the probability of the intersection of multiple events is small, compared to the sum of cutset probabilities. It is most accurate when event probabilities are small, and it will provide a conservative estimation when basic events have large probabilities.

Basic events are given probabilities of occurrence from component failure data, or from expert consultation if no data is available. By using this method, top event probabilities are calculated in terms of basic event failure data (Andrews and Moss., 2002). Depending on the type of failure data available, time dependent failure calculations can be made. These can include finding a probability distribution for the time to top event occurrence, failure rates of lower events, or system availability (Vesely et al., 2002).

Importance measures provide useful information about how much each basic event is contributing to the occurrence of system failures (Andrews and Moss., 2002). By identifying which events are large contributors, funds and efforts to improve the system may be directed where the potential for improvement is greatest (Vesely et al., 2002).

There are a number of importance measures which can be calculated for each basic event in a fault tree (Vesely et al., 2002). Each measure may be categorised as either probabilistic or deterministic, where probabilistic importance measures take component failure probabilities into account, and deterministic measures do not (Andrews and Moss., 2002). 5 importance measures, commonly used in fault tree analysis, are now introduced.

2.3.6.1 Structural Importance

The structural importance, I, of the ith component is a deterministic measure, defined in Equation 2.6.

$$I = \frac{\text{number of critical system states for component } i}{\text{total number of states for the } (n\text{-}1) \text{ remaining components}} \tag{2.6}$$

where a critical state for component i, is a state of the system such that if component i was to fail, the system would change from working to a failed state (Andrews and Moss., 2002). This measure is independent of the basic event failure probabilities.

2.3.6.2 Fussell-Vesely importance

The FV (Fussell-Vesely) importance of an event is related to that event's contribution to the top event probability. It can be calculated for every event in a fault

tree below the top event. If a fault tree has n non-repeated basic events (B_1, B_2, ... , B_n), with k minimal cutsets, (C_1, C_2, ... , C_k), the absolute FV importance can be found using Equation 2.7 (the approximation in Equation 2.5 may be used to simplify calculations):

$$\text{Absolute FV importance of event } B_j = \mathbb{P}\Big(\bigcup_{B_j \in C_i} C_i\Big) \approx \sum_{B_j \in C_i} \mathbb{P}(C_i) \tag{2.7}$$

where the absolute FV importance of an event B_i is the total probability of all the minimal cut sets that contain the event. The relative FV importance of an event B_i represents the proportion of top event probability, which consists of minimal cut sets that contain a given event.

$$\text{Relative FV importance of event } B_j = \frac{\mathbb{P}(\cup_{B_j \in C_i} C_i)}{\mathbb{P}(\text{Top event})} \approx \frac{\sum_{B_j \in C_i} \mathbb{P}(C_i)}{\mathbb{P}(\text{Top event})} \tag{2.8}$$

2.3.6.3 Risk Reduction Worth

The Risk Reduction Worth (RRW) of an event B_i is defined as the decrease in the top event probability if event B_i was guaranteed not to occur (Vesely et al., 2002). Therefore, the RRW shows the benefit gained by upgrading a component to the standard of never failing. If component B_i has failure probability p, then

$$\text{RRW of event } B_i = \mathbb{P}(\text{Top event occurs}) - \mathbb{P}(\text{Top event occurs} | p = 0) \tag{2.9}$$

2.3.6.4 Risk Achievement Worth

The Risk Achievement Worth (RAW) of an event B_i is defined as the increase in the top event probability if B_i were guaranteed to occur (Vesely et al., 2002). Events which have a large risk achievement worth should be the focus of improvements, since they have the largest impact on the probability of system failure. If component B_i has failure probability p, then its RAW can be calculated using:

$$\text{RAW of event } B_i = \mathbb{P}(\text{Top event occurs} | p = 1) - \mathbb{P}(\text{Top event occurs}) \tag{2.10}$$

2.3.6.5 Birnbaum's importance

Birnbaum's importance measure (BM) is defined as the rate of change of the top event probability with respect to the probability of a basic event (Vesely et al., 2002). It is related to both RRW and RAW. If component B_i has failure probability p, then its BM importance can be calculated using equation 2.11.

$$\text{BM of event } B_i = \mathbb{P}(\text{Top event occurs}|p = 1) - \mathbb{P}(\text{Top event occurs}|p = 0) \quad (2.11)$$

The three importance measures RRW, RAW and BM are somewhat similar to partial differentials of the top event probability with respect to a given events probability, since all the other events probabilities are kept constant.

2.3.6.6 Limitations

The are some limitations to the range of applicability of FTA. The fundamental issue is that much of the quantitative evaluation of fault tree models relies on the assumption that basic events are independent. This assumption is essential when evaluating cut-set probabilities. For example, consider a cut-set, S_1, consisting of two events such that, $S_1 = \{A, B\}$, then the probability of S_1 is calculated using 2.12.

$$P(S_1) = P(A \cap B) = P(A)P(B) \quad \textbf{iff A and B are independent events} \quad (2.12)$$

In the case of A and B being dependent events, conditional probabilities must be introduced, as shown in 2.13.

$$P(S_1) = P(A \cap B) = P(A)P(B|A) \quad (2.13)$$

2.3.6.7 Summary

Fault tree analysis has shown to be an effective tool for analysing many types of complex systems across multiple industries (Ostrom and Wilhelmsen, 2012). They can be useful for analysing failures to a high level of detail, aiding decision makers to find the optimal place to invest resources, if the objective is to improve the

safety of a system. However, their use can be restricted by the requirement for the independence of basic events, and a focus on reliability.

2.3.7 Markov Models

Markov models can be used to model some systems that do not meet the required assumption for the independence of basic events used in FTA modelling. A Markov processes is any stochastic process with the Markov property, i.e. the next state of the process depends only on the current state, and is independent of all previous states. Formally, a discrete space and time process X_n, is a Markov process if the equality in Equation 2.14 holds.

$$\mathbb{P}(X_{n+1} = k | X_n = j, X_{n-1} = k, X_{n-2} = l, ..., X_0 = a) = \mathbb{P}(X_{n+1} = k | X_n = j) \tag{2.14}$$

If the Markov property is to hold for a given system, this implies that the process must be homogeneous, i.e. the component failure and repair rates must be constant.

Markov processes can be constructed using either continuous or discrete state spaces, and analysed using discrete or continuous time. Typical reliability applications of Markov processes use discrete state spaces with continuous time, i.e. systems made of components with finite modes of operation working in continuous time, such that failures can occur at any point (Andrews and Moss., 2002).

For discrete state space processes, it is common to use a graph to represent the process. Graphs consist of two elements, nodes and edges, where directed or undirected edges run between nodes. The existence of a directed edge connecting two nodes means that the system can only transition between the two states in the direction indicated on the edge, as opposed to both directions for an undirected edge.

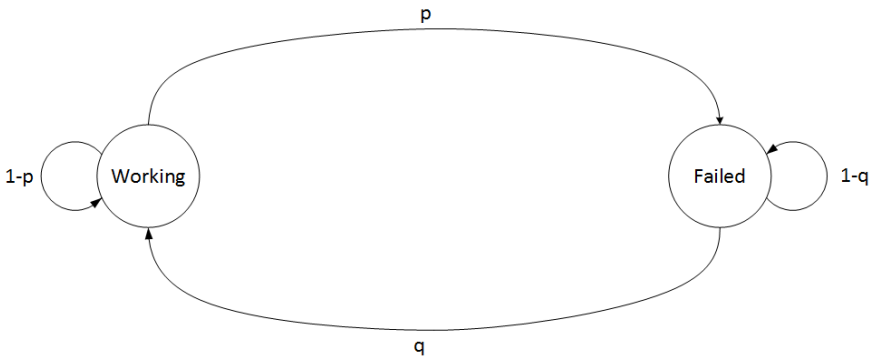
There is a nomenclature for Markov processes used to describe different types of nodes, absorbing, transient and recurrent. Absorbing nodes have the property that once entered, the absorbing state cannot be left. Transient nodes have the property that if the process is left to run for infinite time, a transient state will be visited less than an infinite number of times. Recurrent nodes have the property

that if the process is left to run for an infinite amount of time, a recurrent node will be visited an infinite number of times.

2.3.7.1 A repairable component

In a reliability analysis, the states of the graph represent the possible states that a system may be in. The set of edges represent the system level transitions caused by individual component repairs or failures. Figure 2.3 shows an example of a graph used to represent a 2-state Markov process (Andrews and Moss., 2002). This small example will be used to demonstrate the core principles of Markov analysis.

FIGURE 2.3: Example of a 2 state repairable component



This transition diagram in Figure 2.3 represents a component undergoing failure and repair at constant rates, p and q. Exponential distributions of rate p, and q are used to control when the component changes state. Assuming that the component starts at t=0 in the working state, it transitions to the failed state with constant rate p. Similarly, the component transitions from failure to working with a constant rate q.

If X_t indicates the state of the component as either working or failed at time t, we can consider what state the component will be in at time $t + dt$ (where dt is small enough such that two or more state transitions cannot occur simultaneously). This depends only on the current state of the component.

$$P(X_{t+dt} = Failed) = P(X_t = Failed)(1 - q)dt + P(X_t = Working)pdt \quad (2.15)$$

Equation 2.15 states that either the component was in a failed state at time t and failed to repair in the time interval dt, or the component was in a working state at

time t and repaired in time interval dt. This can be rearranged to give Equation 2.16.

$$\frac{P(X_{t+dt}=Failed)-P(X_t=Failed)}{dt} = P(X_t = Working)p - P(X_t = Failed)q \tag{2.16}$$

and as $dt \to 0$, letting $P(X_t = Failed) = P_F(t)$ and $P(X_t = Working) = P_W(t)$),

$$\frac{P(X_{t+dt} = Failed) - P(X_t = Failed)}{dt} \to \frac{dP_F(t)}{dt} \tag{2.17}$$

and hence

$$\frac{dP_F(t)}{dt} = P_W(t)p - P_F(t)q \tag{2.18}$$

We have that $P_W(t) + P_F(t) = 1$, so we can write

$$\frac{dP_F(t)}{dt} = (1 - P_F(t))p - P_F(t)q = p - (p + q)P_F(t) \tag{2.19}$$

Assuming that the component starts in the working state, using an integrating factor of $\mu(t) = e^{(p+q)t}$ to solve the 1st order differential equation gives

$$\frac{dP_F(t)}{dt} = \frac{p}{p+q} - \frac{p}{p+q}e^{-(p+q)t} \tag{2.20}$$

where a similar calculation for $P_W(t)$ can be performed to derive $\frac{dP_W(t)}{dt}$. This gives

$$\frac{dP_W(t)}{dt} = \frac{q}{p+q} + \frac{p}{p+q}e^{-(p+q)t} \tag{2.21}$$

2.3.7.2 The general case

If we assume a general Markov process X_t, has n reachable, non absorbing states, S, and let $P_i(t)$ be the probability that the process is in state i at time t. Then consider $P_{ij}(t+s)$, the probability that the process transitions from state i to state

j in time s. Then for all i, $j \in S$, we have the Chapman-Kolmogorov equations (Epstein and Weissman, 2008).

$$P_{ij}(t+s) = \sum_{k \in S} P_{ik}(t) P_{kj}(s) \tag{2.22}$$

which can be written in matrix form,

$$\mathbb{P}(t+s) = \mathbb{P}(t)\mathbb{P}(s) \tag{2.23}$$

and has solution

$$\mathbb{P}(t) = e^{Qt} = I + \sum_{k=1}^{\infty} \frac{(Qt)^k}{k!} \tag{2.24}$$

Where I is the n dimensional square identity matrix, and Q is an n dimensional square matrix. Differentiating 2.24 gives

$$\mathbb{P}'(t) = Q \sum_{k=0}^{\infty} \frac{(Qt)^k}{k!} = Q(I + \sum_{k=1}^{\infty} \frac{(Qt)^k}{k!}) = Q\mathbb{P}(t) \tag{2.25}$$

This equation is dependant, so a further equation is needed to be able to fully solve the system.

$$\sum_{i=1}^{n} P_i = 1 \tag{2.26}$$

The Matrix Q in 2.25 is the transition rate matrix of the process, it describes the changes of system states. If X_t is a continuous time Markov process with n states, then the square matrix $Q^{n \times n}$ is the transition rate matrix if elements q_{ij} of the matrix are such that

$$q = \begin{cases} q_{ij} = \text{Transition rate from state } i \to j \\ q_{ii} = -\sum_{i=1 j \neq i}^{k} q_{ij} \end{cases} \tag{2.27}$$

Finally, to be able to calculate $P_i(t)$, an initial distribution for the system to start in has to be either assumed or known. Let the initial distribution be denoted by the row vector π, where $\pi_i = P_i(0)$, then if $P(t) = (P_1(t), P_2(t), P_3(t), ...)$ is the

row vector of the state distribution at time t, then

$$P(t) = \pi\mathbb{P}(t) \tag{2.28}$$

where $\mathbb{P}(t)$ is the transition probabilities matrix.

2.3.7.3 Solving for $\mathbb{P}$(t)

There are multiple methods for analytically computing the transition probabilities matrix, such as a Laplace transform, the eigenvectors of Q, or a Taylor expansion approximation (Epstein and Weissman, 2008). Alternatively, a numerical approach can be used to calculate the value of the state equations for larger systems where analytical solving methods may be cumbersome. An example of how to find the state equations numerically will be demonstrated.

2.3.7.4 A numerical approach

An alternative method for finding values of the state equations is to numerically approximate the process. This can be done by assuming an initial distribution, and approximating the process over consecutive small discrete time intervals dt. Given an initial distribution π_0, the state equations are defined for $t = 0$.

$$P(0) = \pi_0 \tag{2.29}$$

The state equation for $P(0+dt)$ can be approximated by using the rate transition matrix Q in a first order numerical solving technique.

$$P(0+dt) \approx P(0) + \frac{dP}{dt}dt = P(0) + P(0)Qdt \tag{2.30}$$

where from Equation 2.25, $\frac{dP}{dt} = P' = PQ = P(0)Q$. Iterating this process calculates approximations for the state equations for the Markov process at a number of time intervals determined by the size of the interval dt.

2.3.7.5 Steady state probabilities

Steady state probabilities, π_n, of an n state Markov chain, determine the probability the process will be in a given state after it has been running for a long period

of time. The process is defined as ergodic, if the steady state probabilities are independent of the initial starting state of the system (Andrews, 2009). Such a distribution exists for irreducible Markov processes, where a process is irreducible if all the states communicate, i.e. any state i can, by some set of transitions of the system, be reached from any other state j. The 1 × n row vector π_n satisfies the following equations.

$$\pi_n = \pi_n Q \tag{2.31}$$

$$\sum \pi_n = 1 \tag{2.32}$$

Steady state probabilities for state i can be obtained by solving the system of simultaneous equations 2.31, or more directly by using Equation 2.33 (Andrews and Moss., 2002).

$$\pi_i = \frac{\begin{vmatrix} q_{11} & \cdots & q_{1,n-1} & 0 \\ q_{21} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 \\ q_{i1} & \cdots & q_{i,n-1} & 1 \\ q_{i+1,1} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & \cdots & q_{n,n-1} & 0 \end{vmatrix}}{\begin{vmatrix} q_{11} & \cdots & q_{1,n-1} & 1 \\ q_{21} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & 0 \\ q_{i1} & \cdots & q_{i,n-1} & 1 \\ q_{i+1,1} & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ q_{n1} & \cdots & q_{n,n-1} & 1 \end{vmatrix}} \tag{2.33}$$

The numerator of this expression is the determinant of the rate transition matrix Q, with the final column set to zero apart from the element $q_{i,n}$, which is set to 1. The denominator is the determinant of the rate transition matrix Q, with the nth column all set to 1. This direct expression can be a useful check on steady state values derived from a numerical approximation of a Markov process.

2.3.7.6 Limitations

The Markov analysis method has some major drawbacks (O'Connor and Kleyner, 2012). It is a required assumption that the failure and repair rates for all components are constant, throughout the whole life of the system. Furthermore, the occurrence of events must be statistically independent. Such a situation rarely occurs in practice.

Another issue arises while trying to model larger systems containing many components. The dimension of the matrix Q is a function of the number of components of the system being modelled. If $Q^{n\times n}$ is the rate transition matrix of a system, letting X_{s_i} be the number of states the ith component contained in the system can be in, then for a system with k components, n can be calculated as a product using Equation 2.34.

$$\textbf{Number of system states} = n = \prod_{i=1}^{k} X_{s_i} \tag{2.34}$$

If the components are of the simplest possible type, then they will only have 2 possible states of operation, working or failed. Thus we can write a lower bound on n in terms of the number of components. If k is the number of components in the system being modelled, we have Equation 2.35.

$$n \geqslant 2^k \tag{2.35}$$

Thus the size of the Matrix Q required to model a system using a Markov process, grows exponentially with the number of components. As the state space grows, the tractability of the model lessens, as the numerical calculations required to estimate the process become increasingly difficult to compute.

2.3.7.7 Summary

If a system exhibits a lack of memory then Markov models may be a suitable assessment technique. However, the requirements for their application, constant failure rates, and independence of events, are very strong and are often not satisfied (O'Connor and Kleyner, 2012).

2.3.8 Petri Nets

A further expansion of state-space analysis modelling techniques came when Petri Nets were introduced. A Petri net is a bipartite directed graph (Schneeweiss, 1999). A bipartite graph is defined as a graph with two sets of nodes, where edges only exist between the two sets. To derive results from a Petri Net, it must be programmed and simulated for a period of time. In this thesis, a simulated environment is built using C++, in which objects are created which behave analogously to those found in a Petri Net (i.e. places, transitions, arcs, etc.), which Petri Nets can then be built from.

Initial Petri Nets took no account of the passage of time, and thus all transitions were instantaneous. The introduction of deterministic and stochastic delay timings for transitions created the extension Stochastic Petri Nets (SPN) (Bause and Kritzinger, 1996). SPNs alleviate most of the limitations of Markov chains by switching focus away from the overall system state, onto the state of components, such that the overall system state can be inferred from the states of the components (O'Connor and Kleyner, 2012). This also removed the strict restriction of using *only* the exponential distribution to control transition delays.

2.3.8.1 Basic elements

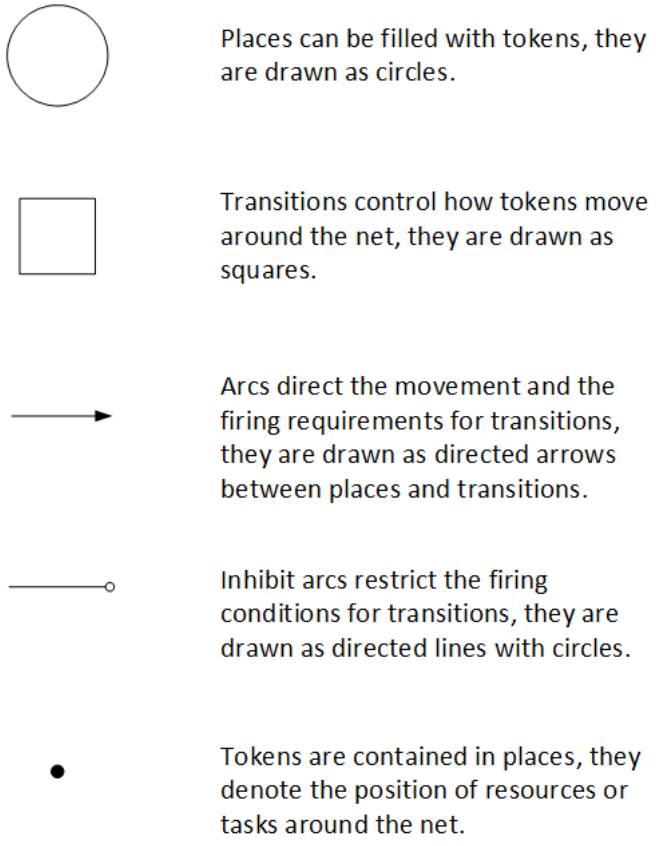
The basic symbols used for constructing Petri net graphs are shown in Figure 2.4.

As shown in Figure 2.4, Petri nets use circles to represent places, and squares to represent transitions. Directed arcs are used to connect places to transitions, and vice versa. Inhibit arcs may connect places to transtions, such that a place is said to inhibit a transition if this is the case. Places may contain an integer number of tokens, which are drawn as small black circles, and transitions control the movement of tokens around the net using the firing rule.

2.3.8.2 The firing rule

In its most basic form, on a net which contains only arcs of weight 1, the firing rule operates as follows. If all input places of a transition are marked with at least one token, the transition is enabled. Upon being enabled, the transition fires after a time $D \geq 0$, equal to the timing of the transition. Upon firing, a token is

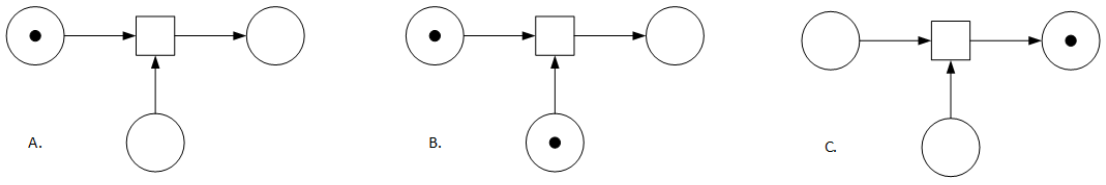
FIGURE 2.4: Elements of Petri Nets



instantaneously removed from each input place of the transition and a new token is placed in each output place of the transition (Schneeweiss, 1999).

If D is always a deterministic (constant) value, then the Petri Net is deterministic. However, if D is also allowed to be drawn from probability distributions, then the Petri Net is said to be an GSPN. Figure 2.5 shows a basic example of the firing rule.

FIGURE 2.5: Example of the firing rule

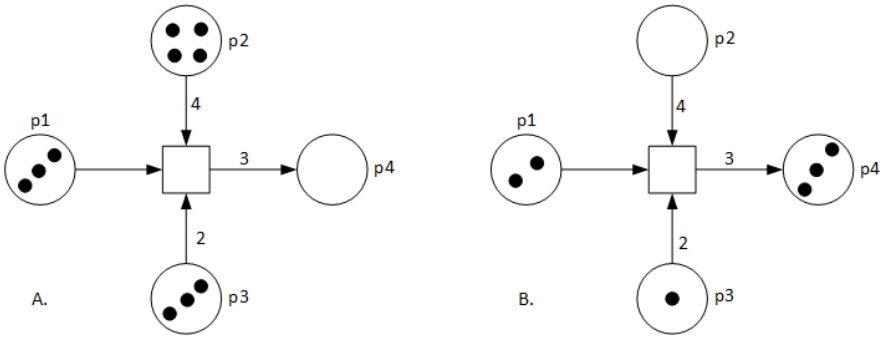


A. Initially the transition is disabled. B. A new token arrives, since a token is on each input place, the transition becomes enabled. C. After a time D after the transition becoming enabled, the transition fires using the firing rule to redistribute tokens.

2.3.8.3 Weighted edges

This basic definition of Petri Nets can be expanded upon by introducing weighted edges. The firing rule is altered by the inclusion of weighted edges. The number of tokens required to enable a transition becomes equal to the weight on the edge, and the number of additional tokens placed onto output places is equal to the weights of the edges running from transition to output places. Figure 2.6 shows a Petri Net with weighted arcs, before and after the firing of the transition (Note that arcs of weight one remain unlabelled).

FIGURE 2.6: The firing rule with weighted edges



A. A the transition is enabled. B. At a time D after the transition fires, tokens are redistributed.

Upon the transition in Figure 2.6 firing, a number of tokens are removed from each input place equal to the weight of the arc connecting input place to transition. At the same time, a number of additional tokens are placed onto each output place of the transition, where the number of additional tokens is equal to the weight of the arc connecting transition to output place.

Both Figures 2.5 and 2.6 have shown an important property of Petri Nets, that in general, the total number of tokens on the net is not conserved. Such cases can exist (for example building a Petri Net to represent a Markov process), but in general this is not a requirement.

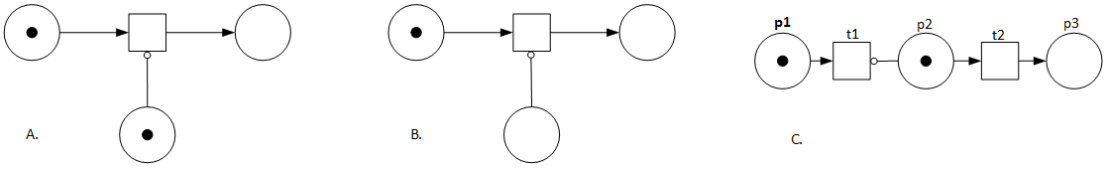
2.3.8.4 Inhibitor arcs

Another extension of the Petri Net modelling framework are inhibitor arcs. Inhibitor arcs provide the net with additional conditions for the enabling of transitions, where normally an arc running from an input place to a transition means

that the transition is enabled if there are at least the arc's weight number of tokens on the place. Inhibitor arcs inverse this principal. Inhibitor arcs run from places to transitions and they change the transitions enabling condition, such that the inhibited transition is only enabled when the input place connected by the inhibitor arc contains less tokens than the inhibitor arc's weight (Schneeweiss, 1999).

The introduction of inhibitor arcs allows transitions to be prioritised over others, and streams of tokens to be blocked. Figure 2.7 shows some examples of inhibitor arc behaviour.

FIGURE 2.7: Inhibitor arc examples



A. The inhibitor arc is disabling the transition. B. The transition is enabled. C. The token on p1 must wait for t2 to fire (removing the token on p2), before transition t1 will be enabled.

2.3.8.5 Formal definition of a general Petri net

Here a formal definition of a Petri Net is given (Schneeweiss, 1999).

A Petri Net is a bipartite graph containing two types of nodes (places and transitions) and directed edges, defined by 2.36.

$$G_{PN} = (V_p, V_t, E; M(0), D, W), \quad E \subseteq (V_p \times V_t) \cup (V_t \times V_p), \tag{2.36}$$

G_{PN} is a Petri net made up of the set of places V_p, the set of transitions V_t, and the set of arcs E, which can go from V_p to V_t, or from V_t to V_p; where M(0) is the initial marking vector (initial conditions) of the set V_p places, D is the vector of delays controlling the timing of the transitions V_t, and W is the vector of arc weights of the edges E.

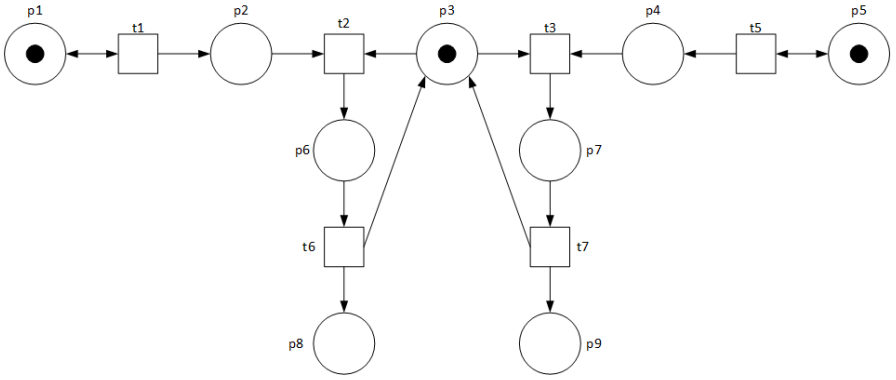
The vectors $M, W \in \mathbb{I}^p$, since places may only contain integer numbers of tokens, and arcs weights are also integer valued. The random positive vector $D \in \Re^t$, $d_i \geq 0 \quad \forall i$.

There are different types of Petri nets in use, so this definition isn't exhaustive, since some extensions require more objects to define them fully. For example, coloured Petri nets (described in 2.3.10) require transition guards, transition conditional statements, and token colours to be fully defined. Drawing out the net rather than presenting the formal definition allows humans to better understand the structure and purpose of a given net, so holding this in regard, the pictorial method is used for the remainder of the thesis.

2.3.8.6 Modelling mutual exclusion

Mutual exclusion within a system is characterised by the shared used of a limited resource. Examples could include a robotic arm working on two construction lines, or if thinking within the field of pharmacy, a pharmacist that is required to check two different types of prescriptions. The resource (robotic arm/pharmacist) is only able to work on one of the queues at a time, so units must wait in their respective queues until the resource becomes available (Schneeweiss, 1999). An example of how a mutually exclusive resource could be modelled is shown in Figure 2.8.

FIGURE 2.8: Mutually exclusive resource



The Petri Net represents two processes which each have a stage completed (transitions 2 and 3) by the same worker. The token on place 3 can be thought of as representing the availability of the working resource. If transition 2 fires before transition 3, a token is placed into place 6, representing the fact that the working resource is operating on the left hand task. Now a situation arises where transition 3 will not become enabled until transition 6 has fired.

Transitions 6 and 7 can be thought of as concurrent processes with a worker who changes between them, where the frequency with which the worker visits each assembly line is controlled by the timings contained within transitions 2 and 3.

Changing the number of available resources in the net, i.e. by increasing the number of tokens on a resource place, could be used as a way of modelling increased availability of resources in a pharmacy. In the simple net of Figure 2.8, both processes would be able to run independently if an extra resource token was added to place 3. Complex arrangements, such as 2 workers facilitating 3 production lines, can also be modelled in this manner.

2.3.8.7 Counting

Petri nets can incorporate the counting of many things during simulations. Counting can be done deterministically, for example, by counting the number of times a deterministic transition has fired during the simulation, the number of cycles a token has made, or the profits related to the process. When simulating community pharmacies it will be useful to keep track of how many tasks a pharmacy has completed throughout a day.

FIGURE 2.9: Counting example

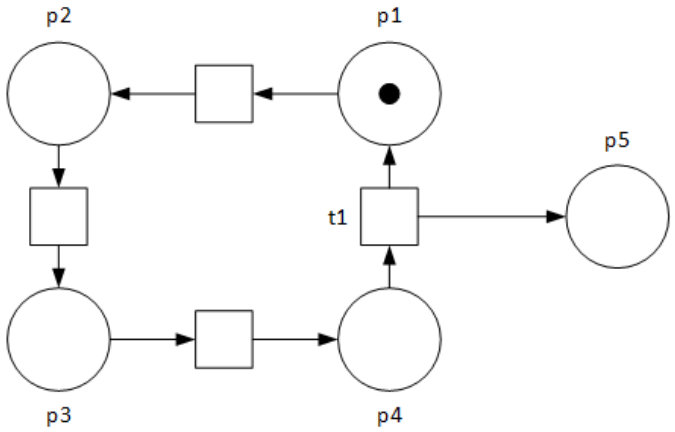


Figure 2.9 shows an example counting structure that could model a four stage process, with a counting place attached to transition 1. Upon the task represented by transition 1 being completed, a token is placed into place 5. The number of tokens in place 5 at the end of a simulation then represents how many times the process has been completed.

2.3.8.8 Concurrency and synchronisation

Concurrency within a system is characterised by two processes going on at the same time, in parallel. There is an abundance of concurrent processes going on within community pharmacies. One could think of two dispensers each working on different prescriptions at the same time, for example.

Synchronisation on the other hand, is a situation where a process cannot be completed until two separate elements are in place. This can be modelled by more than 1 input place being connected to a transition. Delivery prescriptions are packed into parcels and sealed before being handed to delivery drivers. There are a number of synchronizations that must take place before the packages should be sealed. All the required medicine should be present, all labelled correctly. Similarly one could think of the synchronization at the labelling stage, since all items must be picked and labelled correctly before the labels can be applied.

FIGURE 2.10: Concurrent synchronisation

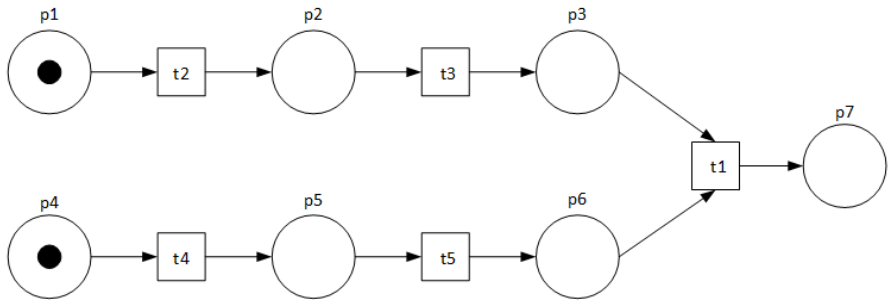


Figure 2.10 shows two concurrent processes running in parallel, which end in a synchronised step. The first process runs from places 1-3, and the second from 4-6. Once both processes have reached places 3 and 6, transition 1 becomes enabled, and after the delay attached to transition 1 the process ends. Whichever of the processes ends first must then wait for the other process to finish before the final step can be completed.

2.3.8.9 Properties of Petri Nets

The properties used to describe Petri nets are given below.

1. Liveness. A Petri Net is considered to be live if it can never reach a state of deadlock. Deadlock defines a state of a Petri Net, where no transitions

in the net are enabled. Liveness is the opposite of deadlock, which implies there is always at least one transition in the net which is enabled.

2. Boundedness. A place in a Petri Net is bounded if there exists and upper limit for the number of tokens on the place. A Petri Net is bounded, if all its places are bounded, and a bounded Petri Net has a finite number of reachable markings. A k-bounded Petri Net has no reachable states where a place contains more than k tokens.

3. Reversibility. A Petri Net is reversible if the initial marking can be reached from any subsequent markings.

There are many classes of Petri nets, where the class of net used for modelling a given system should be tailored to suit the requirements needed.

2.3.9 Timed Petri nets

A timed Petri net is a Petri net in which the transitions are given timing delays, which control the firing of transitions. Un-timed nets are nets without timed delays on transitions, in which all transitions fire instantaneously. Attaching timings to transitions allows Petri Nets to model temporal processes.

2.3.10 Coloured Petri nets

Colours may be added to a Petri net framework, allowing tokens to carry token specific information. This allows for more compact representations of non-coloured nets (Jensen, 1996). In a coloured Petri net each token is assigned a data value, called a token colour. Colours may be composed of a concatenation of data types, such as, integers, strings, or Boolean values. Each place in a Coloured Petri net may only contain tokens of certain specified types, where a token type is a set of token colours.

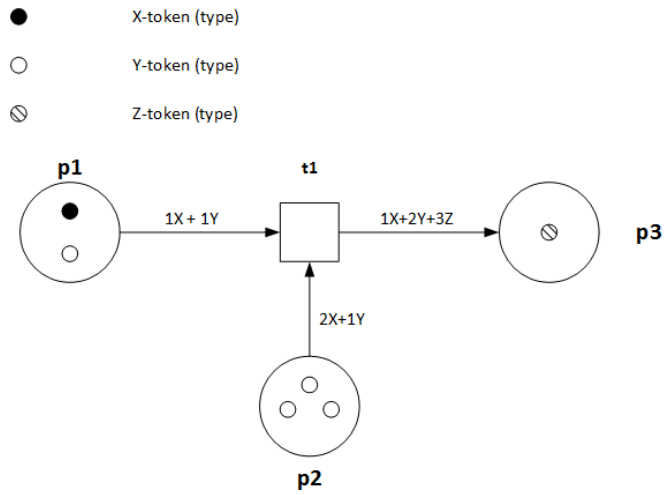
In non-coloured Petri nets, places are marked with integers indicating how many tokens are present on the place. However, places in Coloured Petri nets are marked with multi-sets, where the multi-set indicates how many tokens of each type are on a place. A multi-set is a linear combination of token types, where all coefficients are positive or zero.

The firing rule for Coloured Petri nets is similar to that of non-coloured nets, although there are additional control mechanisms to take into account. These include; transition guards, arc inscriptions, multi-set arc weights and transition bindings.

2.3.10.1 Arc expressions

Having introduced the concept of token types, the Coloured Petri net version of arc weights is now introduced. Arc weights in a Coloured Petri net framework become multi-sets of token types. Figure 2.11 below shows some examples of multi-set arc weights in a Coloured Petri net. Note that the transition in Figure 2.11 in not enabled, since there are no tokens of type X on place 2. Note that different token types can be in the same place, and token types are made up of a combination of colours, where each colour can be a type of information such as, a boolean variable, an integer, a floating point number, etc.

FIGURE 2.11: Coloured Petri net arc weights

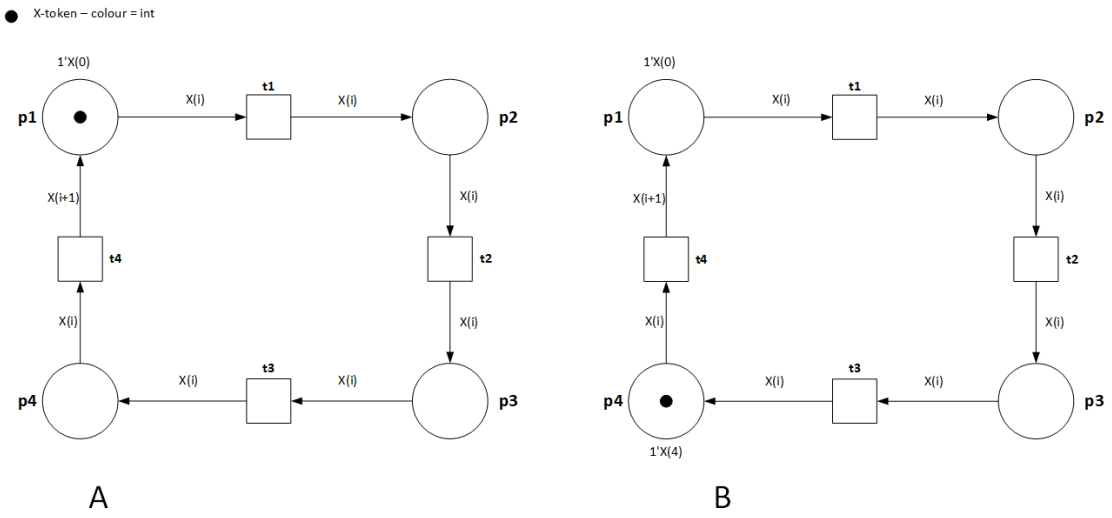


2.3.10.2 Bindings

As well as using colour sets, arc inscriptions also use variables to determine the value of colours attached to tokens when a transition fires. These variables on the arc inscriptions allow transitions to fire in ways that are similar but slightly different each time (Jensen, 1996).

Figure 2.12 shows a simple cyclic process. The arc weights are written as a multi set of token types, in this case, a singular X token for each arc, and along with the X token is the variable i. The variable i is read in from the input place of transitions. In A, transition 1 is enabled for $i = 0$. The specification of a variable for which a transition is enabled is called a binding. Since the X token on place 1 has a value of zero attached to it, if transition 1 was to fire it would use this binding, and an X token with 0 attached to it would be placed on place 2. In this example, X is the token type, and the token type is made up of a single colour, a single integer value. Types can contain an arbitrary number of colours, however, tokens of the same type will have values for each colour in the token type's set of colours.

FIGURE 2.12: Coloured Petri net transition bindings



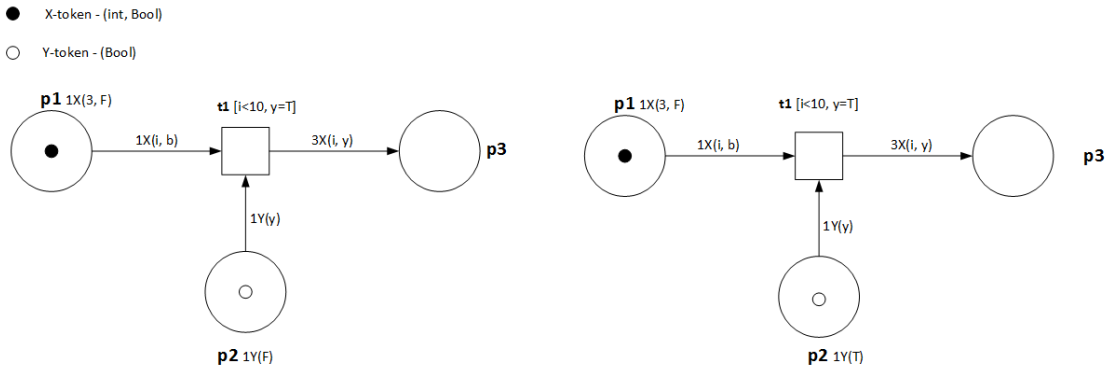
In B, transition 4 is enabled with a binding of $i = 4$. Note also that the arc expression on the output arc of transition 4 is not identical to the input arc. If transition 4 were to fire with the binding of $i = 4$, an $X(5)$ token would be placed onto place 1.

2.3.10.3 Transition guards

Transition guards can be used to constrain the enabling conditions of transitions based on the colours of tokens on their input places. Transition guards contain guard expressions, which evaluate to one of the Boolean values, true or false. If the guard expression evaluates to false, then the transition is disabled.

Guard expressions take the form of multiple Boolean expressions, $b_1, b_2, b_3, ...,$ where each b_i is composed of the variables used in the transition's binding. If a transition guard contains n components, $b_1, b_2, b_3, ..., b_n$, then the truth value of the guard is evaluated as $b_1 \cap b_2 \cap b_3 \cap ... \cap b_n$. Hence all n conditions must be satisfied for a given binding for the guard to evaluate to true, and the transition to be enabled. Figure 2.13 shows an example of a guard expression manipulating the enabling conditions of a transition. Guard expressions are written in square braces next to the transition they are guarding.

FIGURE 2.13: Guard expression example



The left hand transition is not enabled since the second condition of the guard expression requires that the Y token on place 2 has the value true. Note that the first condition that the i variable be less than 10 is met. The right hand transition is enabled, since both the guard conditions are satisfied. The tokens placed onto place 3 after the transition fires inherit colours from both input tokens, since the binding uses values from both token types.

2.3.11 Limitations

Petri nets are a flexible modelling tool capable of modelling systems containing many concurrent processes, and the modelling approach has been said to have no severe limitations (Schneeweiss, 1999). Issues can arise in determining the accuracy of results, since the results can be very sensitive to small changes in parameter values, which are often uncertain (O'Connor and Kleyner, 2012). Consider modelling a dispensing process, if the time to complete each stage of the process is unknown, distribution types and parameters may be estimated or derived from conversations with experts and the estimated parameters may vary from the true

values. An example of this substitution can be seen in Section 6.8, where the results of a CPN changed when parameter values derived from in field data collection were used instead of data derived from conversation with experts.

2.3.12 Summary

The concepts, structures, and a formal definition of Petri Nets have been introduced. The main features of the Coloured Petri net extension have been demonstrated with graphical examples. Together, these will form the basis of further applications of the technique to the process of community pharmacy dispensing in Chapter 6.

2.4 Statistical techniques and distributions

This section introduces key statistical techniques and distributions used throughout the thesis. They are used to analyse 4 sets of timing data collected from UK community pharmacy sites. The fitted distributions are used to control the firing delays of transitions within the Petri nets developed throughout the thesis.

2.4.1 Q-Q plots

Q-Q plots can be used as a qualitative test to see whether a data set follows a distribution (Rice, 2007). If X and Y are two identically distributed distributions, then the Q-Q plot of X against Y will appear as a straight line through the origin with slope 1 (Wilk and Gnanadesikan, 1968). Alternatively, if X is a linear function of Y, then the Q-Q plot will be a straight line (not necessarily of slope 1, or through the origin). The closer a Q-Q plot is to appearing straight, the more likely that the data was drawn from the corresponding distribution.

Below is a summary of the methodology used for producing Q-Q plots for collected 1 variable data against a normal distribution. To produce a Q-Q plot, the data is rank ordered from smallest to largest. Each point is then given a quantile value. If the sample contains 100 samples, the smallest value is the 1st quantile, the second smallest is the second quantile, etc.

The next step is to find the matching quantiles of the distribution being tested. To test the hypothetical data set of 100 timings against a standard normal distribution (N(0,1)), we need to identify the value of the 1st, 2nd, 3rd,... , 100th quantiles of a standard normal distribution. Once these are identified, a scatter plot is plotted where a point is plotted for each quantile pair: ($q1_X$, $q1_{N_{(0,1)}}$), ($q2_X$, $q2_{N_{(0,1)}}$), ($q3_X$, $q3_{N_{(0,1)}}$),... ,($q100_X$, $q100_{N_{(0,1)}}$).

2.4.2 Method of maximum likelihood estimation

Assume best fitting parameters are being estimated for a set of distributions, $S = (f_1, f_2, f_3, ..., f_n)$, to model a set of observations $X = (x_1, x_2, ..., x_n)$. Then for each distribution $f_i(\theta)$, where $\theta \in \mathbb{R}^d$ and $d > 0$, the best fitting parameters $\hat{\theta}$ can be estimated by maximising the likelihood function, using the assumption that the timing of each observation was drawn from the same distribution (Delignette-Muller and Dutang, 2015). The likelihood function is defined below:

$$L(\theta) = \prod_{t=1}^{n} f(x_t|\theta) \tag{2.37}$$

where x_t are the timed observations of the dispensing process, and $f(|\theta)$ is the density of the distribution being fitted.

An intuition of where this comes from can be seen by considering the likelihood of the observed data set, X, occurring. Each data point can be considered to be an outcome of a random trial (RT), where the data set is constructed by conducting multiple random trials. The likelihood of the data set X occurring, then becomes the joint probability of the sample for a given model with parameters in each of the n random trials, as seen in equation 2.38 (Cousineau et al., 2004).

$$P(RT = X) = P(RT_1 = x_1) \cap P(RT_2 = x_2) \cap ... \cap P(RT_n = x_n) \tag{2.38}$$

When the trials are independent, this becomes Equation 2.39.

$$= \prod_{i=1}^{n} P(RT_t = x_i) \tag{2.39}$$

When testing a given PDF f with parameters θ, Equation 2.39 becomes the likelihood function in 2.37.

2.4.3 Akaike Information Criterion

The Akaike Information Criterion (AIC) can be used to compare the fit of different models (Burnham and Anderson, 1998). The AIC of a model is defined in Equation 2.40:

$$AIC = 2k - 2ln(L_{max}(\hat{\theta})) \tag{2.40}$$

where k is the number of parameters, and $L_{max}(\hat{\theta})$ is the maximum value of the likelihood function (achieved at parameter values $\hat{\theta}$). If a set of distributions $(f_1, f_2, ..., f_n)$ are being tested, the distribution which best fits the data is the one which yields the minimum AIC value. Small AIC values are obtained when the likelihood function is large, and distributions with less parameters have an advantage over those with less, this is done to discourage models with many parameters (over-fitting).

The AIC value indicates a relative measure of fit, and it does not provide an objective indication as to how well the distribution is modelling the data. There may be distributions outside the set that are being tested which may fit the data better, but the AIC will not give any indication of this.

2.4.4 P-P plot

A P-P plot compares two cumulative distribution functions against each other on the unit square. If an empirical distribution, F, is generated from the observed data set, and it is tested against a theoretical cumulative distribution function G, then a straight line of points on the diagonal between (0, 0) and (1, 1) indicates that the observed data was drawn from the distribution being tested. They can also be used to test the skewness of a distribution (Thode, 2002).

2.4.5 Empirical distribution function

Empirical distribution function tests for goodness of fit can be performed, however, in this thesis the distribution fitting analysis is focused on the Q-Q plots and AIC values for fitted distributions.

2.4.6 Bootstrap sampling

The bootstrap method involves re-sampling from an observed data set, with replacement, to generate a set of bootstrap samples (Burnham and Anderson, 1998). At least 1000 such samples are necessary to generate conclusions. The set of samples is a proxy for a set of N independent samples of the observed phenomenon. For each of the N bootstrap samples, best fitting parameters $\hat{\theta}_n$ are estimated by maximising the likelihood function. Confidence intervals can then be computed using Equation 2.41:

$$C.I.(\theta_i) = \theta_i \pm 1.96\frac{\hat{\sigma}_i}{\sqrt{N}} \tag{2.41}$$

where θ_i is the estimated parameter found using the real data set, $\hat{\sigma}_i$ is the standard deviation of the set of estimated parameters $\hat{\theta}$ produced from the bootstrap sample, and N is the number of bootstrap samples.

2.4.7 Probability distributions

Distributions are used to control the delay of transitions in timed Petri nets. This section introduces 6 commonly used continuous distributions used throughout the thesis to model the time taken to complete tasks in Petri Nets.

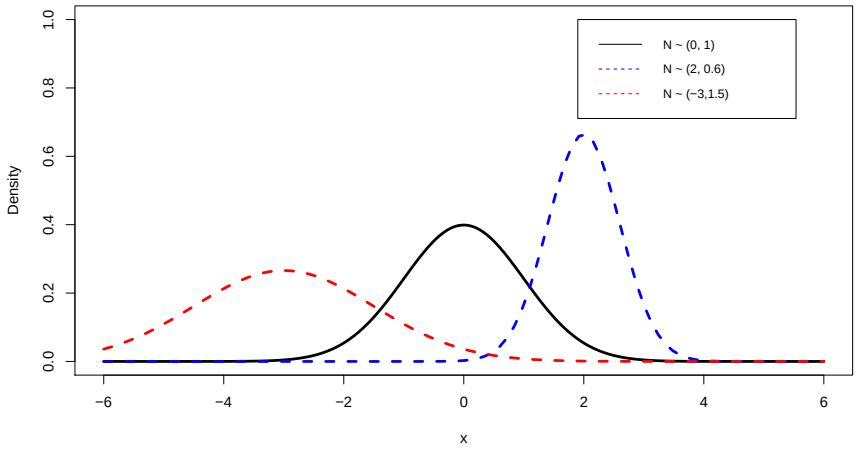
2.4.7.1 The normal distribution

The density of a normal random variable depends on two parameters, the mean μ, and the standard deviation σ, where $\infty < \mu < -\infty$, $\sigma > 0$. The density function of a normal random variable is given in Equation 2.42 (Rice, 2007).

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.42}$$

Figure 2.14 shows three normal densities produced by plotting three random variables, $X_1 \sim N(0, 1)$, $X_2 \sim N(2, 0.6)$ and $X_3 \sim N(-3, 1.5)$. By changing the values of the parameters μ and σ in the normal distribution, different densities can be produced, the three examples above are shown in Figure 2.14.

FIGURE 2.14: Normal densities



Altering the mean controls the location of the highest point of the distribution, increasing the standard deviation widens the distribution, and decreasing the standard deviation tightens the distribution.

2.4.7.2 The exponential distribution

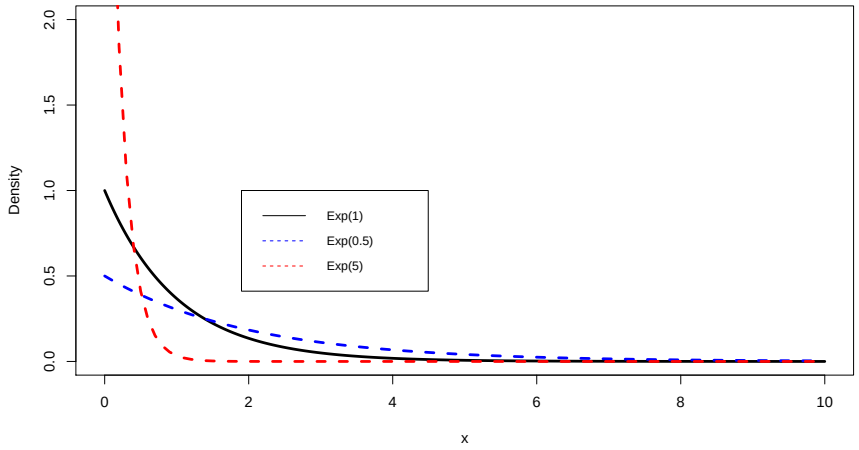
The density of an exponential random variable depends on one parameter λ, where $\lambda > 0$. The density function of an exponential random variable is given in Equation 2.43 (Rice, 2007).

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & else \end{cases} \tag{2.43}$$

Figure 2.15 shows three exponential densities produced by plotting three random variables, $X_1 \sim Exp(1)$, $X_2 \sim Exp(0.5)$ and $X_3 \sim Exp(5)$.

The exponential distribution can be thought of the time taken for a radioactive particle to decay. Changing the value of λ alters the distributions rate of decay. The lower the value of the λ, the slower the rate of decay. Note that all exponential distributions have their highest density approaching 0, and the mean of the exponential distribution is $\frac{1}{\lambda}$.

FIGURE 2.15: Exponential densities



2.4.7.3 The uniform distribution

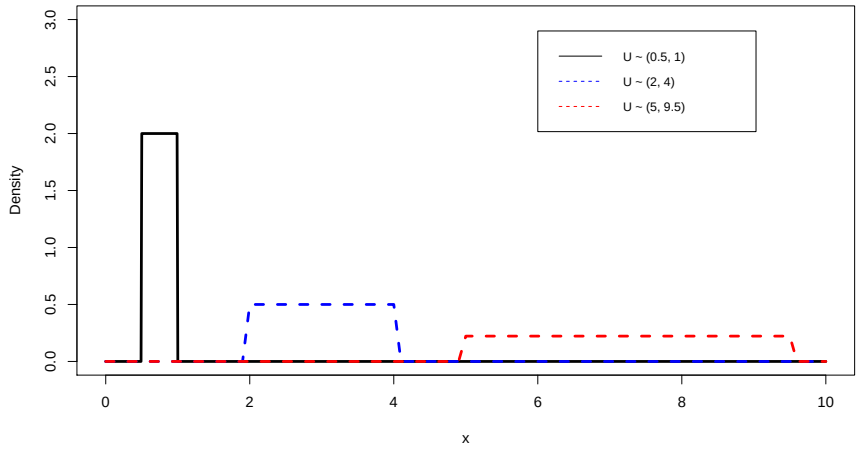
The uniform distribution has two parameters a and b. The uniform distribution gives equal probability to all values in the interval [a, b]. The uniform density is given in Equation 2.44 (Rice, 2007).

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{for } x < a \text{ or } x > b \end{cases} \tag{2.44}$$

Figure 2.16 shows three uniform densities produced by plotting three random variables, $X_1 \sim U(0.5, 1)$, $X_2 \sim Exp(2, 4)$ and $X_3 \sim Exp(5, 9.5)$.

The parameters a, and b control the width of the uniform distribution, and the width has an effect on the probability of each point within the bound [a, b]. Additionally, the larger the difference between a and b, the lower probability for each value within the range [a, b], as can be seen in Figure 2.16.

FIGURE 2.16: Uniform densities



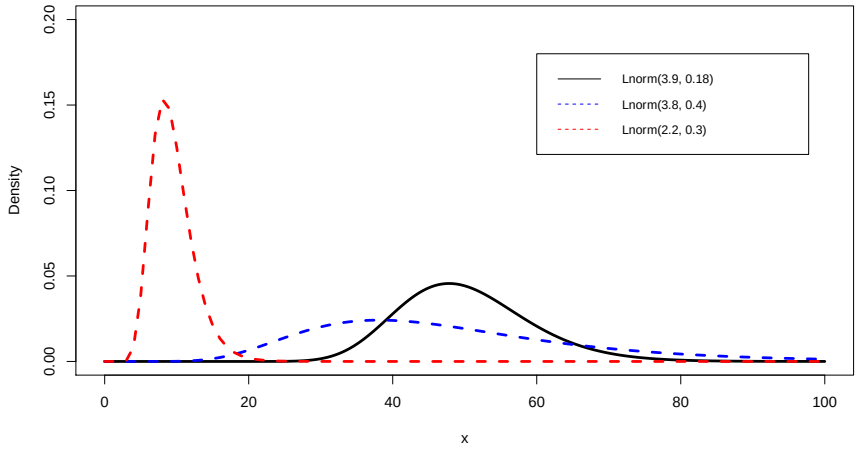
2.4.7.4 The log-normal distribution

The log-normal distribution has two parameters, μ and σ. A positive random variable X, is log-normally distributed if the logarithm of X is normally distributed, $ln(X) \sim N(\mu, \sigma)$. The density function of a log-normal distribution is given in Equation 2.45 (O'Connor and Kleyner, 2012).

$$f_X(x) = \frac{1}{x} \cdot \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right). \tag{2.45}$$

Figure 2.17 shows three log-normal densities produced by plotting three random variables, $X_1 \sim Lnorm(3.9, 0.18)$, $X_2 \sim Lnorm(3.8, 0.4)$ and $X_3 \sim Lnorm(2.2, 0.3)$. The mean of a log-normal distribution is given by $e^{(\mu+\frac{\sigma^2}{2})}$, and the variance is given by $(e^{\sigma^2}-1)e^{(2\mu+\sigma^2)}$ (Weisstein, Na). Both parameters contribute towards the mean and variance of a log-normal distribution.

FIGURE 2.17: Log-normal densities



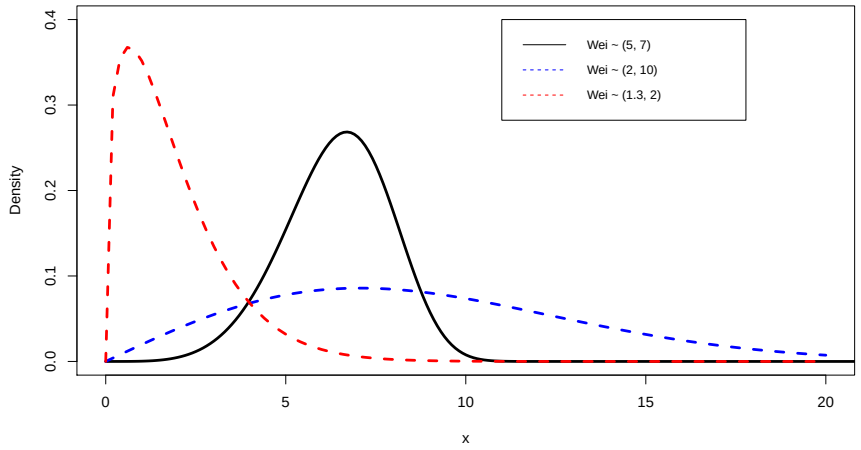
2.4.7.5 The Weibull distribution

The Weibull distribution in its most general case is a three parameter random variable, although in many applications the location parameter is set to zero. This will be the case in this analysis. The density function of a two parameter Weibull distribution, using parameters λ and k, is shown in Equation 2.46 (Ross, 2010). The Weibull distribution is commonly used to model time to failures.

$$f(x) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2.46}$$

Figure 2.18 shows three Weibull densities produced by plotting three random variables, $X_1 \sim Wei(5, 7)$, $X_2 \sim Wei(2, 10)$ and $X_3 \sim Wei(1.3, 2)$.

FIGURE 2.18: Weibull densities



2.4.7.6 The gamma distribution

The gamma distribution has two parameters, α and β. For the case $\alpha = 1$, the gamma distribution is equivalent to an exponential distribution. The density function of a gamma distribution is given in Equation 2.47 (Ross, 2010).

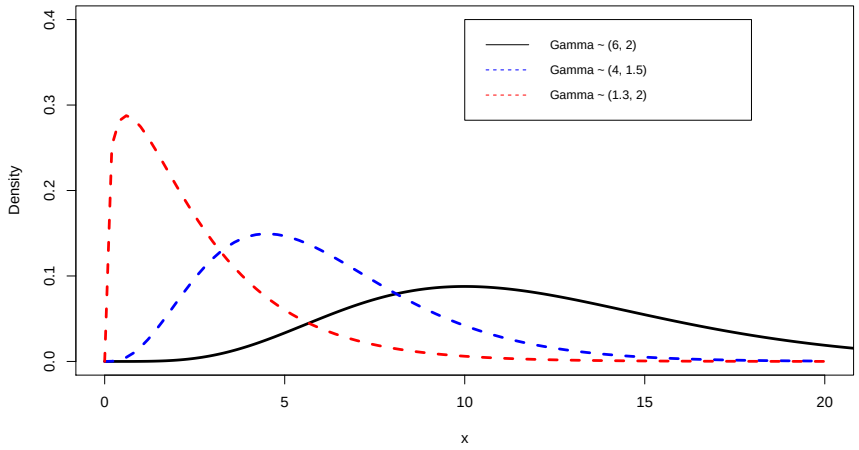
$$f(x;\alpha,\beta) = \frac{\beta^{\alpha}x^{\alpha-1}e^{-\beta x}}{\Gamma(\alpha)} \quad \text{for } x > 0 \text{ and } \alpha,\beta > 0, \tag{2.47}$$

Where $\Gamma(\alpha)$, the gamma function is defined in Equation 2.48

$$\Gamma(\alpha) = \int_0^{\infty} y^{\alpha-1}e^{-y}\,dy \tag{2.48}$$

A notable property of the gamma function is that for integer values of n, $\Gamma(n) = (n-1)!$. Figure 2.19 shows three gamma densities produced by plotting three random variables, $X_1 \sim Gamma(5,7)$, $X_2 \sim Gamma(2,10)$ and $X_3 \sim Gamma(1.3,2)$. The mean of a Gamma distribution is given by $\frac{\alpha}{\beta}$, and the variance is given by $\frac{\alpha}{\beta^2}$.

FIGURE 2.19: Gamma densities



2.4.8 Summary

A number of statistical techniques and objects which are useful for fitting distributions to a set of data have been introduced, as well as 6 common probability distributions which are used to control the firing of transitions of Petri nets throughout the thesis.

2.5 Conclusion

This chapter has introduced the background theory of the reliability engineering modelling techniques that have been used to model the community pharmacy dispensing process in this thesis. Some limitations and advantages for each technique have been discussed, although a deeper discussion of how these qualities manifest will be presented while applying the techniques. Chapter 4 presents an initial investigation of the feasibility of: using Fault Trees to model failures, and Markov models to model the availability of a community pharmacy. Chapter 6 describes the development of a CPN model of a community pharmacy which is later optimised in Chapter 7.

An understanding of how community pharmacies dispense medicines in practice, the types of non-dispensing tasks that are common to pharmacies, and the staff roles occupied have been introduced. This understanding of the dispensing process is the basis of all initial attempts to model the process, and it will be referred back

to throughout the thesis as a basis for understanding how pharmacies dispense medication.

Chapter 3

Literature Review

3.1 Introduction

This chapter of the thesis will review the literature relevant to applying reliability engineering techniques to the process of dispensing in community pharmacies. This will include: the motivation for modelling the dispensing process, a review of modelling projects or studies carried out in community pharmacies, and an overview of the application of reliability engineering modelling techniques in practice.

3.2 Background

3.2.1 Safety

The last two decades have seen a growing awareness of iatrogenic (unintentional harm caused by a physician during medical treatment) patient safety issues within healthcare systems. Reports by the US Institute of Medicine (Mullan et al., 2011) and the UK Department of Health (Department of Health Chief Medical Officer, 2000) provided evidence of clear patient safety issues within healthcare systems, and demonstrated an intention from government institutions to create safer healthcare systems. At the time in the US, it was estimated that between 44,000 and 98,000 Americans died each year from medical errors, and the national cost in the US was estimated to be between $17 and $29 billion dollars annually (Mullan

et al., 2011). Whereas in the UK, it was estimated that 10% of hospital admissions experience an adverse event, which cost the health service £2 billion annually (Department of Health Chief Medical Officer, 2000). Although these reports were focused on secondary care settings, such as hospitals, it was known that errors could be occurring at a similar rate in primary healthcare sectors. The costs associated with errors in primary care in the UK are estimated to be £84,000,000 (Elliott et al., 2018).

More recently, patient safety in primary care has become a higher priority for the World Health Organisation (WHO), which identified pharmacies as a priority primary care area for improving patient safety in 2012 (WHO, 2012). They acknowledge that the majority of patient safety work had been done within secondary healthcare settings, and hence there is far less understanding of patient safety issues in primary care, where the majority of health care is offered (WHO, 2016).

A BBC documentary titled, Boots: Pharmacists under pressure?, aired on BBC 1 on the 3rd of February 2018 (BBC, 2018). The documentary featured an ex Boots manager who had resigned over concerns about under staffing and patient safety within Boots pharmacies, as well as a number of case studies of dispensing errors which had gone on to cause significant harm. Dispensing errors that lead to significant harm often receive media coverage, and there is a literature of case studies of medical errors leading to patient harm (Fusco et al., 2015) (Silva and Krishnamurthy, 2016).

Many studies have tried to evaluate how many errors occur in the community pharmacy dispensing process (Chua et al., 2003) (Knudsen et al., 2007) (Franklin and O'Grady, 2007) (Ashcroft et al., 2005) (Flynn et al., 2002b) (Sánchez, 2013) (Cochran et al., 2013) (Flynn et al., 2003). Such studies collect data on the prevalence of near misses and dispensing errors, over a period of time, using either self reporting or external observations of the process to collect data. These studies have returned a spectrum of rates, between 0.014% (Knudsen et al., 2007) and 3.3% (Chua et al., 2003) per item dispensed. Table 3.1 shows a summary of the key information from studies undertaken to evaluate near miss and dispensing error rates within community pharmacies.

Using the dispensing error rates found in these studies, to project how many dispensing errors are occurring within UK pharmacies, highlights how even small

Table 3.1: Summary of studies of dispensing errors and near misses in community pharmacies

Study	Type: Self-report/external	Near Misses (%)	Dispensing Errors (%)	Items/prescriptions	Contents/Labelling /Administration
1 (Chua et al., 2003)	Self-report	0.48	0.08	Items	78.4 / 15.7 /5.9
2 (Knudsen et al., 2007)	Self-report	0.024	0.014	Prescriptions	15.6 / 64.9 / 19.4
3 (Franklin and O'Grady, 2007)	External	n/a	3.3	Items	51.5 / 48.5 / n/a
4 (Ashcroft et al., 2005)	Self-report	0.22	0.034	Items	62.9 / 33.5 / n/a
5 (Flynn et al., 2002b)	Both	1.28	1.57	Prescriptions	35.2 / 58.2 /6.6
6 (Sánchez, 2013)	Self-report	1.84	0.51	Prescriptions	31.4 / 68.6 / n/a
7 (Cochran et al., 2013)	External	N/a	2.3	Items	n/a / 100 / n/a
8 (Flynn et al., 2003)	External	N/a	1.7	Prescriptions	33.8 / 66.2 / n/a

rates of error can translate to a sizeable patient safety issue. For example, the Office for National Statistics reported that UK community pharmacies dispensed 1.104 billion items in 2016 (Prescribing and Medicines Team, 2017). Combining this figure, with the 2 boundary error rates described above, suggests that nationwide errors number between 155 thousand to 36 million each year. Another study examined how many errors make it past a hospital pharmacist's accuracy checks, which returned an error identification rate of 79% (Cina et al., 2006).

The variability in error rates of these studies can be attributed to a number of factors. These include the different methodologies used to collect data, variable definitions of dispensing error and near miss used between studies, different sample sizes, and study durations. However, a key variable is the methodology of data collection. Studies which used external examinations returned significantly higher rates of error (Flynn et al., 2003) (Franklin and O'Grady, 2007). Evaluations of error detection methodologies have shown that all error detection methodologies underestimate the true number of errors (Flynn et al., 2002a) (Cullen et al., 1995), although direct observation was found to capture the highest proportion. Direct observation is generally recognised as the most accurate method of error detection, recording many more errors than other methodologies (including self-report), when used in other healthcare settings (Flynn et al., 2002a) (Barker et al., 2002) (Dean and Barber, 2001).

Ashcroft et al. (Ashcroft et al., 2005) analysed data on 125395 prescribed items from 35 UK pharmacies and found that incidents occurred during every stage of the process. An 'incident' was defined as either a labelling, content or bagging error. Out of the 330 incidents found during the study, errors were found during creation of labels (27.9%), assembly (19.1%), final accuracy check (27.0%), bagging up (1.5%), handing over to patient (6.4%) and, found by the patient or representative when in their possession (15.5%) or other stages (2.7%). These

results may not be representative of all pharmacies, but they suggest that safety should be a consideration during every stage of the dispensing process.

Taking the problem of imperfect error detection, and the generally accepted higher accuracy of the observational error detection methodology into account, the 36 million projection of yearly dispensing errors in the UK may be an underestimate. For comparison, a large scale study of the number of medication errors within the NHS estimated that 237 million errors were occuring each year, of which 15.9% (37 million), were attributed to dispensing errors (Elliott et al., 2018). Although, it should be noted that not all such dispensing errors would have medical consequences. In the study which returned the dispensing error rate of 3.3%, each error was rated in terms of its clinical significance on a 10 point scale by a panel of 4 experts. Scores of 0 represented no effect, and 10 meant death. Scores of 0-3 were classed as minor errors, 4-7 moderate, and 8-10 serious errors. The mean score of all errors found in the study was 2.4. Of the 95 errors spotted by external examiners, 67% were minor, 34% were moderate, and 1% (a single error), was classified as serious. Although this study only included a single pharmacy, the example provides some context to the 36 million figure, and the relative severity of the majority of errors occurring in community pharmacies.

3.2.2 Efficiency

As well as being required to provide a reliable supply of medication for patients, studies have reported that patient's satisfaction with pharmacy services is linked to waiting times (Afolabi and Erhun, 2003), and long waiting times have been given as a reason for why patients will not return to a particular pharmacy (Somani et al., 1982). Although there is some evidence that in other healthcare settings, managing patient expectations can be a way to make up for longer waiting times (Dansky and Miles, 1997). Furthermore, in the UK, community pharmacies operate in a competitive environment. There are over 11,000 stores, of which 40% are in clusters, where three or more pharmacies are located within a 10 minute walk of each other (DOH, 2016). To encourage customers to return to use the same pharmacy repeatedly, and not use an alternative provider, pharmacies must offer a good service and meet customer's requirements (Dansky and Miles, 1997).

3.2.3 Summary

This Section has reviewed the background literature related to safety issues within community pharmacies and the wider healthcare sector. This included a detailed review of dispensing error studies carried out in community pharmacies, and a review of the literature related to efficiency concerns with pharmacy performance, and the reality of how pharmacies operate in competitive conditions.

3.3 Reliability Engineering methods in healthcare

There is a growing school of thought within the healthcare community that their attitude to risk should mirror practices seen in other high risk industries (Reason, 2000). This will involve healthcare organisations learning from mistakes effectively, so as not to repeat the same mistake twice. Reliability engineering techniques have been used to manage risk in a wide range of fields, such as the aviation, space, nuclear, and chemical industries (Garrick, 1988) (Hsueh and Mosleh, 1996) (Netjasov and Janic, 2008). There is an established collection of reliability engineering techniques capable of performing system failure analyses (Andrews, 2009). This Section presents a review of 3 reliability engineering techniques, Section 3.3.1 reviews fault trees, Section 3.3.2 presents projects using Markov models, and Section 3.3.3 shows applications of Petri Nets and Coloured Petri Nets to the field of healthcare.

3.3.1 Application of fault trees

Fault and event trees are commonly used reliability engineering techniques dating back to 1961, when they were first used to study the Minuteman missile launch system (Bell Telephone Laborotories, 1961); NASA further promoted the technique with the publication of their Fault tree handbook in 2002 (Vesely et al., 2002). Fault trees have seen use in improving reliability in a range of industries, such as the aerospace (Bell Telephone Laborotories, 1961), nuclear (Cummings, 1975), food production (Park and Lee, 2009) (Crosetti and Bruce, 1970), and the

healthcare sector (Makajic-Nikolic et al., 2016) (Hyman and Johnson, 2008) (Liu et al., 2011) (Abecassis et al., 2015).

For example, Fault trees may be evaluated qualitatively using one of two major methods. These two major approaches are Monte Carlo simulation, and deterministic methods (Lee et al., 1985). Monte Carlo simulation assigns failure rates to each component using exponential distributions. During a simulation, the time to failure of all components are sampled from their corresponding distributions. Components switch from 'working' to 'failed' after their time to failure has elapsed, and eventually the top event occurs. This generates a single cut-set, which may be truncated to produce a minimal cut set. The deterministic method involves a reduction of the top event, into terms of purely basic events using Boolean algebra. Computer algorithms have been developed to automatically perform both of these procedures (Vesely and Narum, 1970). Quantitative analysis involves applying the inclusion-exclusion expansion principle to produce a top event probability.

Marx and Slonim (2003) and Wreathall and Nemeth (2004) introduce socio technical probabilistic risk assessment (ST-PRA), as an alternative prospective risk mitigation technique to failure mode and effects analysis (FMEA) in the healthcare sector. Marx et al. present FMEA as the current tool used in accredited hospitals for their annual risk assessments. The pros of FMEA are as follows: it is a prospective technique which aims to mitigate the adverse effects of failures before they occur, and it aids designers in understanding the effects of failures, the cons are: risks are quantified in a subjective way, and combinations of errors which can lead to failures are not considered. ST-PRA is presented as an alternative technique which is used in high risk industries, and it is defined as a probabilistic risk assessment (PRA) which also includes human errors and actions within the analysis. Two example fault trees are presented to demonstrate the difference, where the ST-PRA fault tree contains some human actions, such as 'tubing kinked by patient movement', and 'clamp not removed from tube', as basic events. In a similar vein, Wreathall et al. present a more detailed analysis of the pros and cons of using ST-PRA in a healthcare setting. Some of the main differences between traditional fields of application and the healthcare sector are highlighted. The main difference is that a team delivering a healthcare procedure is not as fixed as a mechanical or structural component, since teams are made up of aggregations of equipment and expertise that may differ in each instance.

Makajic-Nikolic et al. (2016) present a fault tree analysis (FTA) of the medical waste management process in a healthcare centre in Serbia. Infection spreading is the top event, which can occur at 4 stages of the process: during treatment at a hospital, while being transported either to the warehouse, or the treatment facility, or while the waste is being treated at the treatment facility. 4 sub trees are presented for each of these failures, with an OR gate above. The sub trees include many repeated events, including some human actions, such as 'intentional failure to respect procedures', 'not using protective equipment', and 'inattention at work/fatigue'. The tree contains 28 basic events, many of which are repeated. Since the tree is small, the top event is written as a sum of 9 minimal cut sets. To allow a quantitative analysis, basic event probabilities were estimated using advice from people working in each of the 4 areas of potential failure. FV, Birnbaum, RAW, and RRW importance measures were calculated.

Abecassis et al. (2015) reviewed the literature of wrong site surgery, to produce a fault tree of the undesirable event where a medical surgical procedure is performed on the wrong person. A 14-step process map is drawn, and a corresponding fault tree contains 24 basic events. No quantitative analysis is performed. Their fault tree was validated by a mix of surgeons, and industrial engineers experienced in FTA.

(Hyman and Johnson, 2008) illustrate how FTA can aid the design of processes to reduce errors by including more AND events, so that failures can only occur if multiple independent events occur simultaneously. A detailed FTA of the failure of clinical alarms to effectively alert medical staff to adverse patient conditions is presented. Four main sub categories of error are considered; delayed response to an activated alarm, failure to trigger alarm, relevant patient parameter not measured, or patient condition not observed. Some of these considerations of alarm efficacy may be applicable to patient medication record (PMR) dispensing alerts, which are used in community pharmacies (Ojeleye et al., 2013). PMR systems store past information about prescriptions dispensed to patients, and perform patient specific checks of medical appropriateness based on demographic characteristics and clinical data. The FTA highlights how alarms can sometimes be ignored if they frequently deliver false positive signals, have a lower than adequate volume, or fail to trigger when required.

Liu et al. (2011) use fault tree analysis with some branches of the tree replaced by modular Markov models, to analyse a gastric esophageal surgery. Surgeries

are complex medical procedures with a large number of contributing risk factors, such as patient age, operation type, equipment condition, and surgical experience of staff involved. A large fault tree of the surgical procedure containing 103 basic events is constructed using a case study hospital in Beijing. Equipment failures and remediable human errors are considered to be time dependent failures, and these branches of the fault tree are replaced by small modular Markov models, which use 4 states for equipment failures and 3 for remediable human errors.

3.3.2 Markov models

Markov models are an effective technique for modelling systems containing repairable components, dynamic operation, and general behaviour which cannot be modelled using simpler methods (Boyd, 1998). This includes modelling systems which exhibit phased modes of operation, where a system evolves dynamically over time (Smotherman and Zemoudeh, 1989). Unlike fault trees, they also see use in a range of areas outside of the reliability field. Examples have included modelling rainfall levels (Gabriel and Neumann, 1962), economic strategies (Chouaid et al., 2004) (Judge and Swanson, 1962), environmental impacts of human activity (Lusseau, 2003), and the gambling industry (Oses, 2008). Reliability applications of Markov models have seen use in the power generation industry (Anderson et al., 1997) (Lisnianski et al., 2012), rail transportation management (Tan et al., 2018) (Braga and Andrade, 2019) (Prescott and Andrews, 2015), and computer software reliability analysis (Littlewood, 1975). There have been some applications of Markov modelling in the wider healthcare sector (Sonnenberg and Beck, 1993) (Beck and Parker, 1983) (Melnikow et al., 2006) (Melnikow et al., 2008) (Li and Pack, 2004) (Neuman et al., 2009).

For example, Sonnenberg and Beck (1993) and (Beck and Parker, 1983) demonstrate how Markov models can be used to model life expectancies of patients within a healthcare system, where patients are assumed to be in one of a finite number of discrete health states. Markov models improve upon tree based patient prognoses by accommodating the fact that negative events, such as bleeding or heart attacks, may occur multiple times. A simple 3-state Markov example is shown with the three states: well, disabled, and dead. Outcomes of such a model are the duration of how long patients spend in each of the states. A simple analysis would sum total time spent alive, however, different health states can also be attributed

variable qualities, and they can be weighted using a utility function. Including a cost value for each state can further allow cost effectiveness calculations. Two treatments can be evaluated as Markov processes, where the utility of each process is calculated to determine which is the most appropriate treatment.

Melnikow et al. (2006) and (Melnikow et al., 2008) used Markov models to calculate the quality adjusted life years (QALY's) for female patients undergoing, Tamoxifen treatment. Tamoxifen is given to women to reduce the risk of breast cancer, however, taking the drug frequently incurs negative side effects such as menopausal symptoms, or increased risk of other cancers. A Markov model was used to calculate the QALY's gained by the two prospective courses of treatment (tamoxifen, or no tamoxifen), over a set of prospective patients. Their results showed that tamoxifen treatment was dominated (more expensive with less utility) than the no tamoxifen treatment for many groups of women, although for some groups there was a small utility benefit with a high yearly cost. Probabilities for the model were derived from the published medical literature.

Neuman et al. (2009) have used a similar technique to model the care outcomes of patients with rectal cancer. Their model used 5 states representing the possible outcomes of treatment. Quality adjusted life years (QALYs) are calculated for two alternative treatments. The values for transition probabilities and health state utilities were derived from the literature where possible, or based on expert advice if no relevant literature was available.

Li and Pack (2004) develop a Markov model for the occurrence of vertebral fractures in post menopausal women with osteoporosis. Clinical trials have recorded the number of vertebral fractures suffered by women undergoing different courses of treatment at yearly intervals. The results of which can be written as a 4-tuple, where the ith element indicates how many fractures the patient had at the ith year. Since humans have 13 vertebrae, a 14×14 matrix can be used as a one step probability matrix to model how many fractured vertebrae will be present in the next year. One step probabilities p_{ij}, the probability a patient with i fractures at the current year will have j fractures in the next year, can be calculated using:

$$p_{ij} = \frac{k_{ij}}{\sum_j k_{ij}} \quad for \quad i, j = 0, ..., 13, j >= i \tag{3.1}$$

where k_{ij} is the number of patients who transitioned from i fractures to j fractures in a single year during the clinical trial. A one year transition matrix is constructed for two categories of patient, those undergoing a Risedronate treatment and a placebo group. These are then multiplied to generate 5 and 10 year transitions matrices, which are used to predict the probabilities of a patient sustaining vertebral fractures over those time periods based on the treatment undertaken.

Perraudin et al. (2013) used a 4-state Markov model to evaluate the cost effectiveness of a community pharmacist led sleep apnoea intervention. The model used year long cycles for time intervals which matched the literature on long term compliance with treatment. The cost effectiveness of 3 strategies was compared, screening with pharmacists, screening without pharmacists (GPs only), and no screening. The effectiveness of the screening process was not fully understood so a range of values from 25%-65% were tested. Screening with pharmacists dominated no screening, and if the successful screening rate was 35% or above, then it also dominated screening without community pharmacists. Similar evaluations of the cost effectiveness of pharmacist interventions for Warfarin monitoring Chang et al. (2017) and diabetes care plans Yu et al. (2013) have been done.

3.3.3 Petri Nets and Coloured Petri Nets

Petri nets, first seen in 1962 and named after inventor Carl Adam Petri (Petri, 1962), have proven to be a highly effective way of modelling systems containing concurrent processes (Schneeweiss, 1999). A number of Petri net modelling projects have focused on modelling healthcare systems in secondary healthcare settings, such as hospital departments (Dotoli et al., 2010) (Dotoli et al., 2009) (Darabi et al., 2009) (Xiong et al., 1994) (Mahulea et al., 2017). Much of this work is focused on the efficiency of the care delivery system.

Dotoli et al. (2010) modelled a Pulmonology department (PD) using timed Petri nets to analyse the pharmacy ordering strategy in an effort to reduce the number of orders sent to the central pharmacy each year. The way patients enter, recieve treatment, and leave the PD is given, as well as a description of how the PD's pharmacy stock is maintained. Three TRQ stock management policies are outlined, where if the stock levels drop below R at a given time T, a new stock order of size Q is placed. The first of these is the policy used by the case study PD in an Italian hospital in Bari. Historical patient data is used to derive distributions

for patient arrivals at the PD. The Weibull, exponential and Poisson variables are tested for distribution fitting, with the Poisson distribution being chosen as the best fit. Results of the current ordering strategy taken from the hospital are compared against those derived from simulations of alternative ordering strategies using the timed Petri net, which reduced the number of yearly orders by taking extra information into account, such as the number of patients currently in the PD, or their average medicine intake, when deciding the size of orders (Q).

Dotoli et al. (2009) also modelled an emergency cardiology department of the same hospital in Bari using a continuous Petri net methodology. The paper explains how a standard timed PN was converted to a continuous timed Petri Net by allowing; places to contain non-negative real values, transitions to displace real values of resources, and a new firing rule which determines the values a transition may fire with based on the values on places. Additionally each transition is given a flow rate, λ. Each resource is given a weight related to their relative costs which are then summed to produce an objective function. The rate of patient outflow is designated as a performance metric of the emergency department. The patient outflow must be maximised, and this is written as a constraint in the linear program of the problem, where the overall cost of resource must be minimised while the output of the department is maximised. The linear program is then solved to find the optimal integer values for resources in the ED.

Darabi et al. (2009) present Petri Net modelling as a technique for modelling work flows within the University of Illinois hospital. The proposed model is designed to keep track of the availability of resources within the system to allow for sound strategic management decisions. Patients are modelled individually, while resources are grouped together into resource classes. There are 3 classes of resource, human resource (nurse, doctor, transport, assistant, etc.), equipment (X-ray machine, CT scanners), and facilities (operating rooms, recovery rooms, etc.), where some human resources are capable of completing a wider range of tasks than others depending on their qualification. The model is built by mining patient work flows from patient logs, where a work flow consists of a series of actions or activities performed on patients during their hospital visit. Resources are modelled as being available, unavailable, or assigned, and some considerations are made to how equipment can be decommissioned. This decommissioning is modelled by removing tokens from the PN. The model makes use of transitions

to determine which human resources complete tasks if there is a choice due to multiple available staff being qualified. No simulation results are presented.

Xiong et al. (1994) present a Petri Net performance analysis of an emergency medical services system. The emergency department is tested under 2 scenarios, the first with all patients treated with the same priority, and the second where patients arrive in 4 states of ill health: life threatening, urgent, serious, and non-urgent. The state of the patient determines how they are treated by the hospital, where patients in the most serious two classes skip registering on arrival and receive prioritised treatment.

Mahulea et al. (2017) showed how a large healthcare system can be modelled using a modular Petri Net approach. A case study healthcare system in Zaragoza is modelled by 4 interconnected Petri net modules, one each for two primary care trusts, one for the hospital, and one for a specialised care facility.

Coloured Petri nets have seen use in a wide range of fields. These have included modelling biological systems (Liu et al., 2017a) and processes (Bardini et al., 2017), cloud based information architecture (Narayanan and Cherukuri, 2018), cyber physical attacks on key infrastructure (Liu et al., 2017b), supply chain business analysis (Zegordi and Davarzani, 2012), manufacturing systems (Viswanadham and Narahari, 1987) (Wang, 1996), and the healthcare sector (Dammasch and Horton, 2007) (Reed et al., 2017) (Hughes et al., 2000) (Salimifard et al., 2013). Such a wide range of application areas demonstrates the flexibility of the modelling technique.

Dammasch and Horton (2007) modelled patients travelling through the German mental healthcare system using coloured stochastic Petri Nets, which additionally included active tokens to dynamically model patients compliance during treatment programs. The compliance of patients was an aggregation of the patients mood, and the complexity of their treatment program. Their CPN was small and only consisted of 3 places and 4 transitions. The way the model is focused on states that patients can be in during treatment, is similar to the Markov models seen for evaluating different treatment programs. The active tokens are used to model patients. Their level of compliance is the active component, which can change independently of events (e.g. reducing compliance after unsuccessful treatments), while also in some cases being affected by events (e.g. increasing compliance after successful treatment).

Reed et al. (2017) presented a paper outlining a Coloured Petri net based technique for modelling variations in venepuncture procedures. Their aim was to analyse the performance of different variations seen in the field of venepuncture, and to direct resource allocation for improvements toward key steps in the process. Reliability and efficiency of the venepuncture process are defined as: the proportion of successful procedures, and the amount of resources and time used to complete a procedure. The data collection techniques used to learn about the process were multi-phased, making use of structured interviews, an on-line questionnaire, and a literature review, to gain information about the venepuncture process.

A step-by-step venepuncture procedure is laid out, and variations of the procedure found during data collection are presented. In their model, patients are assumed to have a difficulty rating attached to them, either difficult or non-difficult. The difficult group represents patients who are harder to collect blood samples from due to their physical condition. The model assumes that it is harder for staff to perform the procedure on these patients, and uses a wider triangular distribution, than the one for non-difficult patients. Patients were assigned a random value of 0-100 to represent their vein visibility. The probability of failure to collect a blood sample from a given patient is then determined by the group they are in (easy/difficult) and the random number 1-100 assigned to them. Uniform probability distributions are used to model the time taken for each task, with a minimum and maximum value correlating with the collected data set, where all extreme outlier values were discounted. The overall structure of the analysis was to consider 3 variations of the venepuncture process and analyse how many failures of each type occur when staff are following each variation.

Hughes et al. (2000) proposed the use of timed Coloured Petri nets to model a progressive care system. They make use of a modular Petri net so that each of the care units is modelled by a single Petri net module. The system was made up of 7 care units, an adult intensive care unit, a recovery room, an operating room, and four general hospital wards. Patients move around the 4 types of unit as their care progresses, where each unit has a limited number of beds. Two colour sets were used in the model, one to model the type of bed slot available, the other to model patients possible routes through the care system. The model assumes that all patients of the same type make use of the same delay times to simplify how the model works. Such an assumption could be useful when modelling community pharmacies.

Salimifard et al. (2013) used a hierarchical timed coloured Petri net to model a hospital emergency department. Their model was designed to mimic the characteristics of an emergency department in Iran, where data was collected from hospital information systems and interviews. 5 scenarios which varied the resources and protocols within the hospital were simulated against three performance criteria. The first scenario was considered a base case, with 1 general practitioner, 1 specialised practitioner, and 1 admission staff. The first variation considered increasing the number of specialised practitioners to 2, the second variation had 2 specialised practitioners with no general practitioners. The 3rd and 4th variations of the base case involved changing how the procedures were carried out. In these two scenarios the triage and visit stages of the process are carried out at the same time by a GP, rather than sequentially. CPN tools, a publicly available software tool for simulating coloured Petri nets, was used to model the emergency department.

The performance criteria of the emergency department were as follows:

1. Patient waiting time, the mean duration of time spent in the waiting room.
2. Length of stay, the total time spent in the emergency department.
3. Resource utilisation (%), the total busy time of resources compared with total working time.

The emergency department was simulated for a long run of 3 years, and the results of the performance indicators are presented in a way such that all scenario outcomes are shown relative to the base case. Each patient token carries a set of colours; some continuous variables, arrival time, wait time, and process time, some binary variables, such as an indication whether a test is needed, or the results of tests. The outcome of transitions in the high level net of the emergency department modelling for key events such as admission or radiology, are modelled using lower level hierarchical nets. Keeping track of the utilisation rate of staff resources can provide useful information about community pharmacies, a similar concept is used in the CPN (Versions 2, 3, and 4) developed in chapter 6.

3.3.4 Summary

This section of the literature review has examined the literature of the three reliability engineering techniques used throughout this thesis. The focus of the review was on those techniques being applied in healthcare settings.

3.4 Studies based in community pharmacies

Pharmacies have been the setting for a small number of modelling projects. Techniques which have been used to analyse aspects of the dispensing process have included, failure mode and effect analysis (FMEA) (Stojkovic et al., 2017), discrete event simulation (Tan et al., 2009), queuing theory (Bahadori et al., 2014), Markov based models (Vila-Parrish et al., 2012), fault and events tree analysis (Cohen et al., 2012), and Petri nets (Augusto and Xie, 2014). Further work has been done on developing a Human Reliability Analysis framework for community pharmacies, to predict error rates within the dispensing process (Zheng et al., 2017).

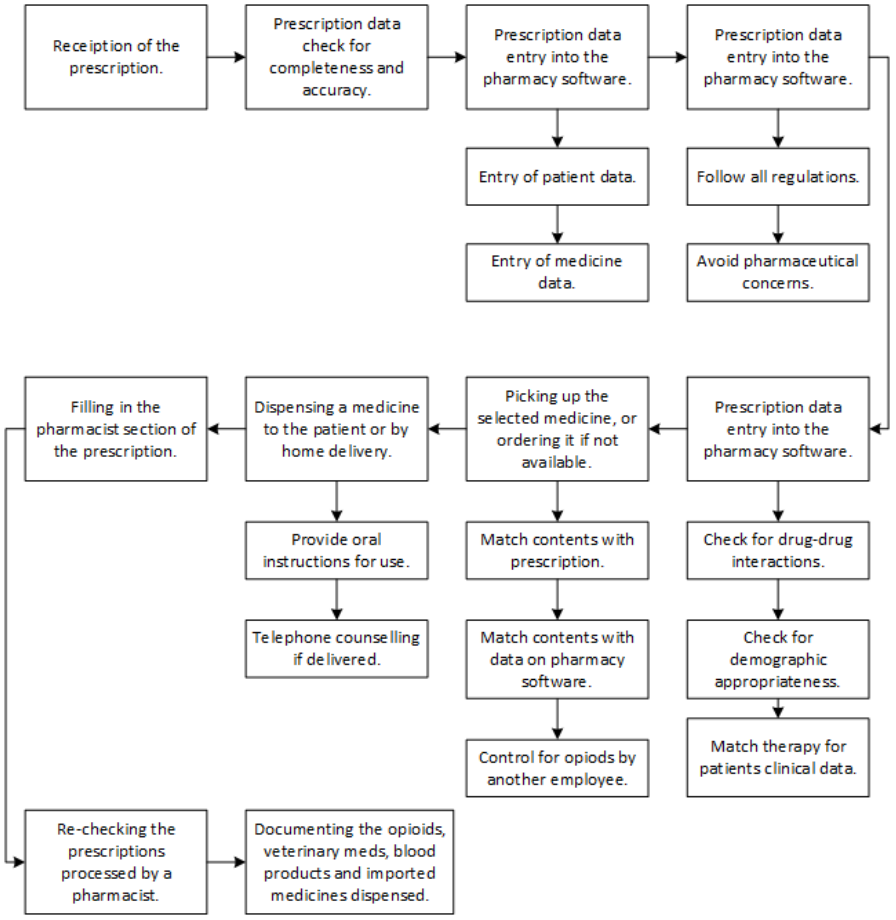
Some of this work based in pharmacies focused on modelling the process of dispensing. If the dispensing process is the focus of modelling, the first step is often to construct a flow chart of how the process is carried out. Different levels of detail are used to produce such a map, depending on the purpose of the project.

3.4.1 FMEA method application

The FMEA project based in German community pharmacies identified 39 failure modes of the dispensing process (Stojkovic et al., 2017). The project used a standard FMEA approach. A team of experts was assembled to map the process, identify potential errors and their causes, rate the risk of each potential failure, and suggest potential corrective actions. The 10 mains steps of their process map is shown in Figure 3.1, with further intermediate steps included.

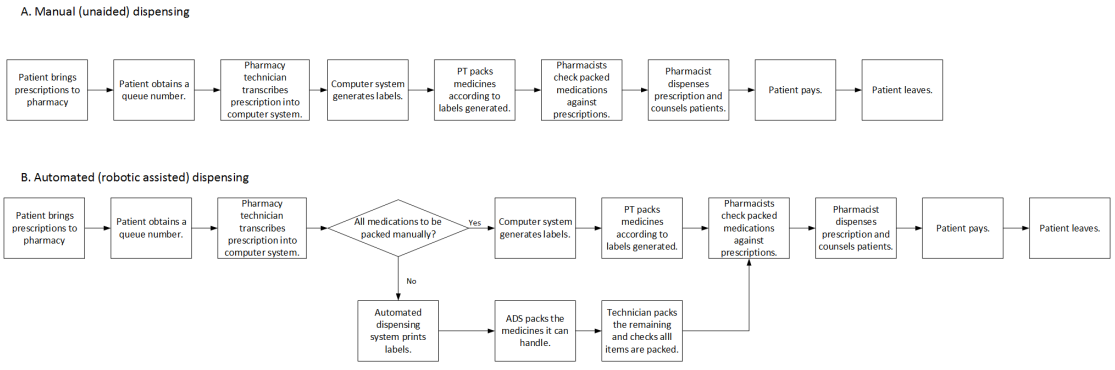
The main causes of failures in the dispensing process were identified as; a lack of standardisation in work processes, work overload, time pressure, distractions, and interruptions. Corrective measures were aimed to remedy the causes of failures. These included standardising the process with additional corresponding staff

FIGURE 3.1: Dispensing process map from Stojkovic et al. (Stojkovic et al., 2017)



training, improving working conditions by increasing the number of staff, and introducing electronic prescribing.

FIGURE 3.2: Dispensing work flows from Tan et al. (Tan et al., 2009)



3.4.2 Discrete event simulation method application

Tan et al. made use of a discrete event simulation methodology to evaluate the impact on waiting times made by the introduction of an automated dispensing machine in an outpatient pharmacy in Singapore (Tan et al., 2009). The paper recognises that patients prefer faster services when collecting their medication, and 30 minutes was used as a target time. Figure 3.2 shows the two different dispensing work flows used to model the process, depending on whether staff are aided by an automated packing robot or working on their own.

Each prescription in the simulation was randomly assigned a number of items following an observed distribution from the study pharmacy, and similarly, each prescription was either tagged as requiring manual dispensing, automated dispensing, or a mixture of both. The time taken to complete tasks was generated from an on site time in motion study using a self reporting data collection technique. Their results indicated that adding a robotic dispensing system would not be sufficient to improve the process to the desired level, they concluded that additional pharmacists were required to widen the bottleneck at the point of handing over prescriptions to patients. The paper demonstrated that process changes can be evaluated at a low cost through computer simulations.

3.4.3 Queuing theory application

Bahadori et al. (2014) used queuing theory to model an outpatient hospital pharmacy. Observations of a military hospital in Tehran were conducted to derive a patient arrival rate, and the rate of receiving services. The 8am - 6pm day was split up into 30 minute intervals to determine how the arrival and service rates changed throughout the day. Their analysis involved varying the number of servers working in the pharmacy, and comparing the queue lengths patients had to wait.

3.4.4 Makov model application

Vila-Parrish et al. develop an inventory control structure for maintaining levels of perishable medicinal products for optimising costs (Vila-Parrish et al., 2012). Although focused on reducing cost, inventory management is also linked to patient care, since shortages of supply can cause errors which arise when dose changes or

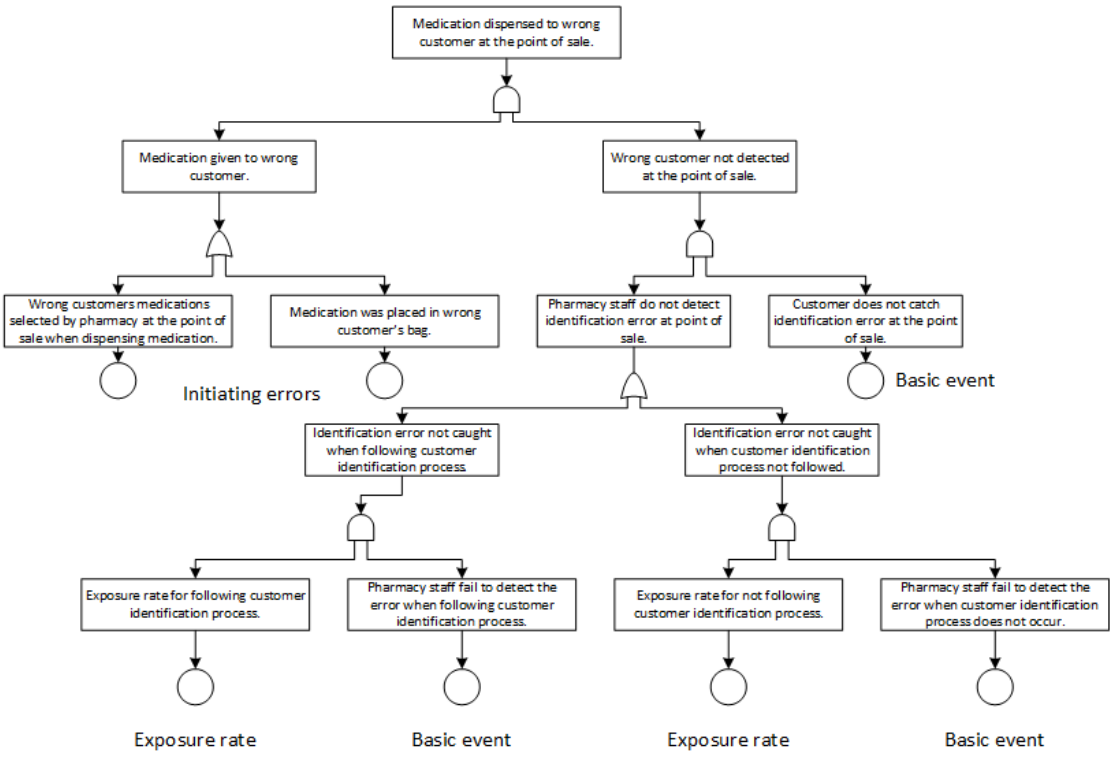
alternative drugs are administered. They aid practitioners responsible for maintaining stock levels of perishable goods by providing inventory optimisation strategies. Their model focuses on the raw materials needed to produce medicines, the degradation of final products ready for patients, and the decisions made about the quantity to buy/produce of each. Sub optimal solutions to these problems can lead to 2 types of error, drug shortages if the inventory is kept too low, or conversely excess inventory levels if too much is ordered. This paper demonstrates another aspect of the community pharmacy dispensing process which can be considered when trying to improve the process.

3.4.5 Fault tree and event tree application

Cohen et al. (2012) performed a socio-technical probabilistic risk assessment (ST-PRA) of the community pharmacy dispensing process. A modelling team constructed fault trees for ten preventable adverse drug events involving six high risk medications. The required input failure rates were generated using a combination of basic equipment failure rates, and expert opinion with reference and guidance from the relevant literature. A variety of sources were used to compile a table outlining error rates for human performance over a range of tasks (Gertman et al., 2005) (Gibson et al., 2006), which was then used to aid the modelling group's error rate estimates. Their estimations for error probabilities began at 0.001, and were adjusted depending on performance-shaping factors. These include the level of training staff have, or working under time constraints. An example of one of their fault tree diagrams is shown in Figure 3.3.

A number of ways of reducing the rate of error for each top event were tested in a sensitivity analysis, where single strategies, and combinations of preventative strategies, were tested for their ability to reduce the top event probabilities. The strategies to improve the safety of the process, such as bar code scanning, or building effective computer alerts into the dispensing procedure, were incorporated into the fault trees, and the resultant drop in top event probability was calculated. The paper showed the utility of quantitative evaluations of the procedure over other purely qualitative methods. It also demonstrated that parts of the dispensing process can be analysed using a fault tree methodology.

FIGURE 3.3: Fault tree diagram from Cohen et al. (Cohen et al., 2012)



3.4.6 Petri net model application to community pharmacy

Augusto and Xie (2014) introduce a Petri net based modelling methodology, MedPRO, designed to address organisation problems within health care systems. MedPRO compiles three alternative views of a medical process, the pathways of care (the ways patients move through the healthcare system), resource activities (how staff are used in healthcare processes, and their various abilities), and organisational dependencies, into a single simulation based model. Each of these views of the system are plotted in UML diagrams, which are easier for non-technical people, involved in the modelling process, to understand. The UML charts are linear process charts mapped out in series, where each event in the process has a set of required resources, and a time for completion. The system then automatically converts these three Unified Modelling Language (UML) descriptions of healthcare systems using an algorithm, into an executable coloured Petri net model.

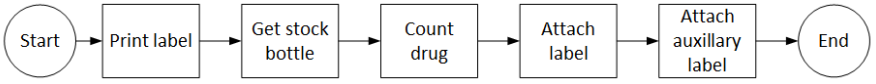
Colours within this framework are used to keep track of resource identities. Each identity has a skill-set (corresponding to types of staff, eg. doctors, nurses, porters, etc.), and each is capable of completing a subset of the available tasks. Tokens of the same colour behave identically when completing tasks. A case study of the

delivery process of medicines for a hospital pharmacy is presented. This process involves transporting medicines from a centralised delivery centre, out to other units in the hospital.

3.4.7 Human reliability analysis

Zheng et al. (2017) used a simple 7 step process map to perform a human reliability analysis of the dispensing process. The aim was to estimate an error rate for the dispensing process as a whole, as well as error rates for the individual stages of the process.

FIGURE 3.4: Dispensing process map from Zheng et al.



Note that this process map doesn't take into account any accuracy checks, since the scope of the project was only to determine the rate of error in the initial part of the process. A cognitive reliability error analysis method (CREAM), was used to describe the procedure, and a set of interacting factors which affect human performance (common performance criteria (CPC)), were used to describe the ways human performance can be affected during a working day. Each CPC was assigned as having a positive, negative, or neutral effect on human performance at each time period, and then by accumulating the effects of CPCs, the process was assigned a quality of performance control mode, which related to an error rate.

Their model estimated that 1% of medications will have the wrong label printed, 2.49% will have the wrong drug, 2.49% will have an incorrect label, and 0.99% will have an incorrect auxillary label. A 7.75% error rate was estimated for the process as a whole. Previous research carried out in hospital pharmacies was cited to estimate the efficacy of pharmacists accuracy checking, which was found to capture 79% of errors (Cina et al., 2006). This meant their estimate for the error rate of the overall process was 1.63%.

3.4.8 Summary

The modelling projects described here have often focused on imporving a single aspect of the dispensing process. For example, the queuing theory and discrete

event simulation projects focused on improving patient waiting times, the human reliability analysis and FMEA projects focused on improving patient safety, and both the Markov based and Petri net modelling projects analysed the distribution and efficiency of resource use throughout a system. It would be desirable to create a model capable of considering many of these aspects, within a single modelling framework. To do so would enable an evaluation of the trade-off between such aspects, for example, analysing how much implementing new safety strategies affects the efficiency of a service. Since both patient safety and waiting times have already been highlighted as two key priorities for community pharmacies, it would be useful to be able to evaluate the relationship between these two aspects.

This section has reviewed the literature of modelling projects being carried out in community pharmacy settings. This has included a range of techniques, including some, such as failure mode and effect analysis (FMEA), or human reliability analysis HRA, which are not used in this thesis.

3.5 Conclusion

This chapter has reviewed the literature of reliability engineering techniques being applied in practice, and reviewed the current knowledge related to the dispensing process in community pharmacies. A motivation for the proposed work has been derived from the literature relating to safety concerns about community pharmacies, and the healthcare sector in general. Models which can evaluate the dispensing process across multiple axes of interest have been identified as missing from the current literature. Developing such a model will be the focus of Chapter 6. This work aims to be another step in the road towards analysing the community pharmacy dispensing process in a similar way to other high risk industries.

Chapter 4

Early analysis of conventional modelling methods

4.1 Introduction

This Chapter demonstrates how two of the modelling methods (Fault Trees and Markov Models) introduced in Chapter 2 can be applied to the dispensing process in community pharmacies. A fault tree is developed for modelling the undesirable top event of a content error being dispensed to a patient. This is used to prioritise events for which improvement efforts should be targeted to produce the greatest impact on reliability. A Markov model of the flow of prescriptions in a community pharmacy is proposed. The steady state performance with increasing members of staff is analysed, and a form of redundancy for accuracy checking is introduced using an accredited checking technician (ACT). Note that this work does not represent the main novel contribution of the thesis, and was conducted at an early stage of the research, as an exploration exercise of the suitability of these traditional modelling techniques.

4.2 Fault Tree Analysis of the dispensing process

The first step of the fault tree analysis is to define the objective of the analysis, i.e. what do we want to gain from completing the FTA? In our case, the objective of the FTA is primarily to see what effects checking procedures have on

dispensing process reliability, how reliable the process is as a whole (in terms of delivering accurate prescriptions to patients), and to evaluate the FTA as a method for studying community pharmacy process reliability. Two fault tree models are developed, and then they combine to produce a single fault tree. A thorough quantitative analysis and interpretation of importance measures is then performed on the resulting fault tree.

The two top events considered as a part of this feasibility test were as follows:

1. A patient receives a prescription which contains incorrect items. (Top event 1)
2. Medication dispensed to wrong customer at the point of sale. (Top event 2)

The top event 2 is an intermediate event of the top event 1. The fault tree for top event 2 was found in literature (Cohen et al., 2012), and the events were populated with the same probabilities from literature. The fault tree for top event 1 was built as a part of this research, by considering the immediate necessary and sufficient conditions for the top event to occur, and working downwards. The dispensing process was used as outlined in Section 2.2.2. Where possible this fault tree was populated with values found in the literature on errors in community pharmacy; some values were assumed if no information was available.

4.2.1 Modelling assumption

One key assumption had to be made to carry out a fault tree analysis of community pharmacies. This section shows that key assumption:

- Independence of basic events
  The occurrence of basic events is assumed to be independent, and therefore, the probabilities of occurrence for cutsets, made up of an intersection of basic events, can be calculated using Equation 2.12.

4.2.2 FTA: A dispensed prescription contains incorrect items

Figure 4.1 shows a fault tree designed to model the event that a patient receives a prescription which contains a content-based error. The event E7 represents the top event found in the literature, and presented in the following Section 4.2.3.

The underlying logic structure of the fault tree can be found using a bottom-up substitution. This is done by using Boolean + to relate events below an OR gate and Boolean * to relate events below an AND gate. The series of Equations 4.1 - 4.6 show how the Boolean logic structure was built from the bottom up.

$$G6 = D + E + F + G + H \tag{4.1}$$

$$G5 = (E + F + G + H + I) * D \tag{4.2}$$

$$G4 = ((E + F + G + H + I) * D) + C \tag{4.3}$$

$$G3 = (((E + F + G + H + I) * D) + C) * B \tag{4.4}$$

$$G2 = (((E + F + G + H + I) * D) + C) * B + A \tag{4.5}$$

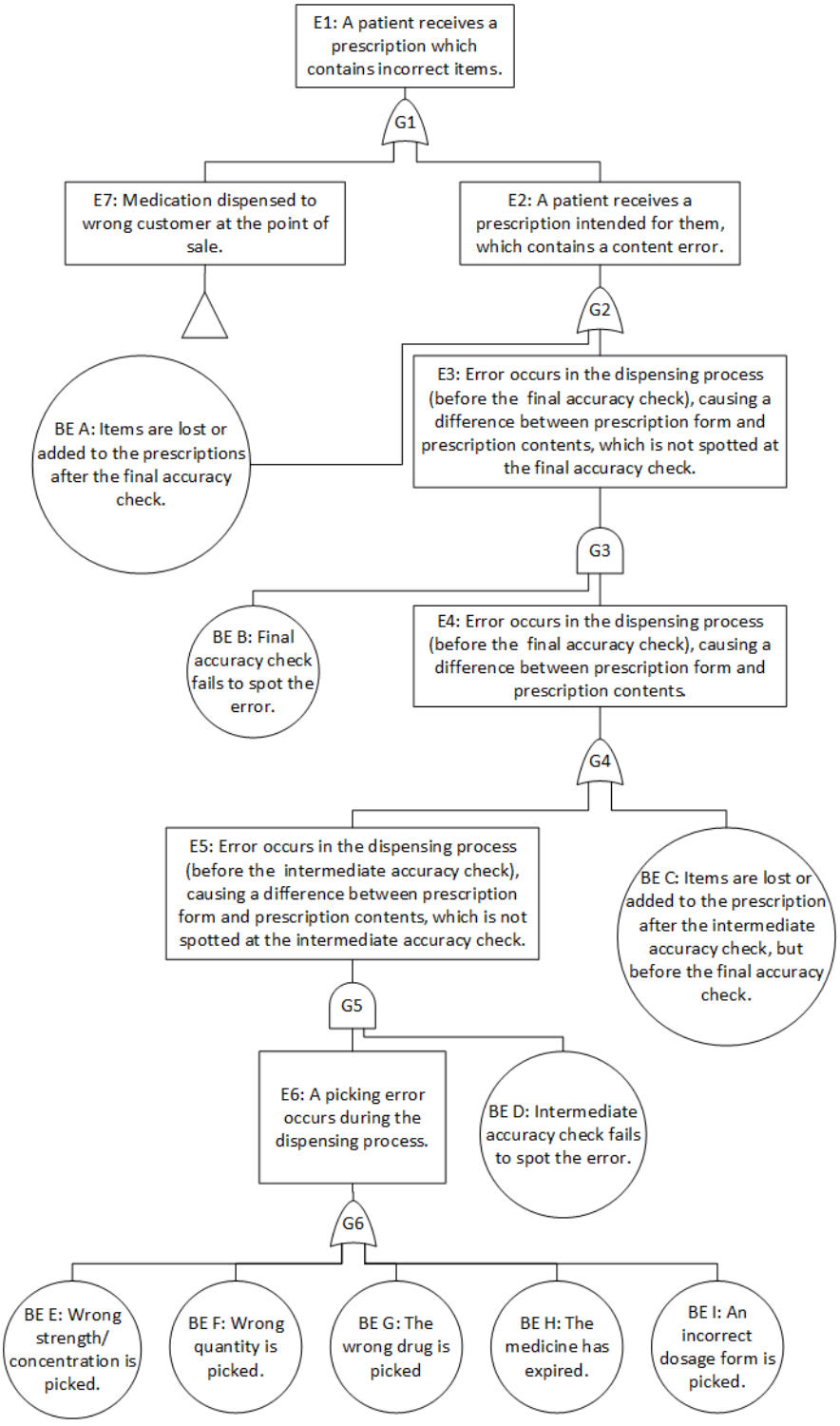
$$G1 = ((((E + F + G + H + I) * D) + C) * B + A) + E7 \tag{4.6}$$

Simplifying the remaining terms from Equation 4.6 gives the Boolean product of minimal cutsets in Equation 4.7.

$$G1 = E7 + A + BC + BDE + BDF + BDG + BDH + BDI \tag{4.7}$$

The tree has 2 low order minimal cutsets involving only 1 and 2 basic events. These are the sets $\{A\}$, and $\{B, C\}$. The event E7 is further developed in Section 4.2.3. The basic event A represents the event where items are added to a prescription after the final accuracy checked has been carried out on it. This could happen if a prescription bag is left open and some medicine falls into it by accident, or if some medicine falls off a shelf landing near to the bag, someone might assume it belongs

FIGURE 4.1: Example FTA of a content error event

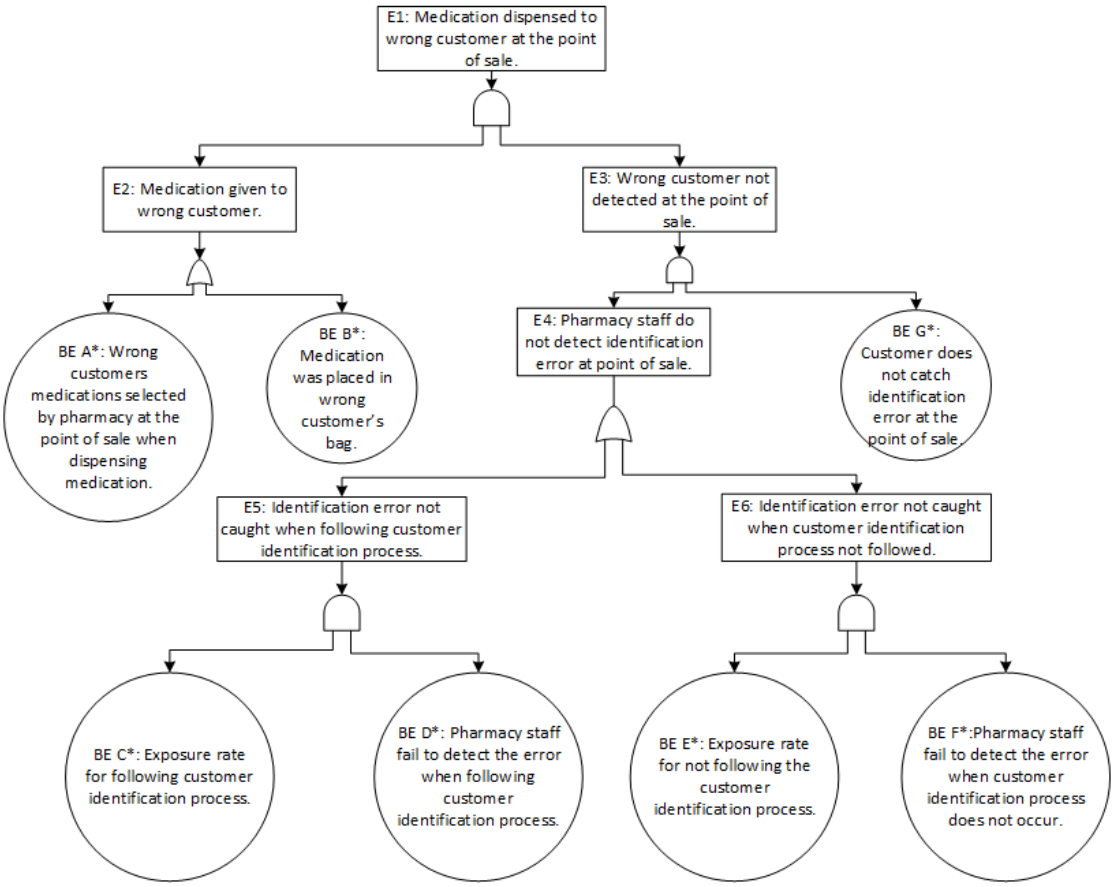


in the bag and place the fallen item into the prescription. The minimal cutset $\{B, C\}$, corresponds to the same event occurring in-between the intermediate accuracy check and the final accuracy check, and the final accuracy check fails to spot the error.

4.2.3 FTA: A prescription is handed out to the wrong person

The products of events on the right hand side of Equation 4.7 are the minimal cutsets of the content error fault tree. The event E7 can be developed further, which can then be presented as a sum-of-product form. Figure 4.2 shows a fault tree modelling the event E7, that a patient is given a prescription which is intended for someone else. The same fault tree found in the literature review by Cohen et al. (Cohen et al., 2012) (See Section 3.4.5) is used to model this top event.

FIGURE 4.2: FTA for a prescription being handed to the wrong person from Cohen et al. (Cohen et al., 2012)



The tree has a simple structure, which can be expressed as shown in Equation 4.8.

$$E1 = E2 * E3 = (A^* + B^*) * G^* * ((C^* * D^*) + (E^* * F^*)) \tag{4.8}$$

Using the distributive law this can be further simplified to give a sum of products representation, shown in Equation 4.9:

$$E1 = (A^* * G^* * C^* * D^*) + (A^* * G^* * E^* * F^*) + (B^* * G^* * C^* * D^*) + (B^* * G^* * E^* * F^*) \tag{4.9}$$

Equation 4.9 says that the event E1 will happen if any one of the 4 Boolean products occur. The 4 products on the right hand side of Equation 4.9 represent the minimal cutsets of the fault tree. They are found to be the sets: $\{A^*, G^*, C^*, D^*\}$, $\{A^*, G^*, E^*, F^*\}$, $\{B^*, G^*, C^*, D^*\}$, and $\{B^*, G^*, E^*, F^*\}$.

Thus, the minimal custsets of the fault tree in Figure 4.1 are as follows: $\{A^*, G^*, C^*, D^*\}$, $\{A^*, G^*, E^*, F^*\}$, $\{B^*, G^*, C^*, D^*\}$, $\{B^*, G^*, E^*, F^*\}$, $\{A\}$, $\{B, C\}$, $\{B, D, E\}$, $\{B, D, F\}$, $\{B, D, G\}$, $\{B, D, H\}$, and $\{B, D, I\}$.

4.2.4 Assigning probabilities

In order to analyse the fault tree, its basic events are assigned probabilities. Where possible, the probabilities of basic events will be assigned using sources from the literature. A study by Cina el al. (Cina et al., 2006) found a hospital pharmacist identified only 79% of prescriptions containing errors. This implies an error probability of 0.21 during the final accuracy check, and for this model it is assumed that the error probability of intermediate accuracy checks are the same. Hence the probability of basic events B and D are assigned to be 0.21.

A dispensing error study by Franklin et al. (Franklin and O'Grady, 2007) found content errors in 1.7% of 2859 prescriptions. Assuming the dispensing process used in that study implemented intermediate accuracy checks and final accuracy checks with error probability of 0.21, the number of prescriptions which contain an error before either accuracy check has taken place can be estimated. In the study, 49 content errors were dispensed, and if the final accuracy check identified 79% of the errors that were presented to it, then there must have been 49/0.21 = 233.3 prescriptions containing errors that were presented to the final accuracy check.

Using a similar calculation for the intermediate accuracy check, implies that 233.3 /0.21 = 1111.1 prescriptions containing content errors were presented at the intermediate accuracy check stage. This implies an estimated 1111/2859 = 38.9%

of prescriptions contained a content error before the intermediate accuracy check. This corresponds to an estimate for content errors, $P(E \cup F \cup G \cup H \cup I) = 0.389$, and subsequently by assuming each type of content error is equally likely, we can estimate each of the individual probabilities (P(E), P(F),... etc.) to be 0.0965 using the full expression of the inclusion exclusion principle. Basic events from the sub tree of event $E7$ (A*, B*,... ,G*), are assigned the same probabilities they were given in (Cohen et al., 2012).

Probabilities of the remaining events (A, and C) have been estimated without reference to the literature since no relevant sources could be found. Events A and C represent items being added to prescriptions after the final accuracy check, or in-between the two checking stages respectively. It is assumed that A is an order of magnitude rarer than C, since prescriptions are usually sealed in a paper bag after the final accuracy check, whereas they are often stored in an open top basket after the intermediate accuracy check. An estimate of the top event probability using the rare event approximation (See Section 2.3.6), is 0.0258. Table 4.1 shows the importance measures for each of the basic events, which were calculated using the Equations in Sections 2.3.6.2 - 2.3.6.5. The top event probability derived from the rare event approximation, was also used repeatedly to calculate importance measures.

Table 4.1: Importance measures: Content error

Basic event	Probability	FV-Absolute	FV-Relative	RRW	RAW	BM
A*	0.003	1.22×10^{-3}	0.047	0.9528	16.69	0.405
B*	0.0004	1.62×10^{-4}	6.29×10^{-3}	0.9937	16.74	0.406
C*	0.5	1.53×10^{-6}	5.94×10^{-5}	0.9999	1.00	2.58×10^{-6}
D*	0.001	1.53×10^{-6}	5.94×10^{-5}	0.9999	1.06	1.55×10^{-3}
E*	0.5	1.38×10^{-3}	0.0535	0.9465	1.05	0.0256
F*	0.9	1.38×10^{-3}	0.0535	0.9465	1.01	1.64×10^{-3}
G*	0.9	1.38×10^{-3}	0.0535	0.9465	1.01	1.64×10^{-3}
A	0.001	0.001	0.0388	0.9612	39.79	1.00
B	0.21	0.0234	0.908	0.0923	4.41	0.111
C	0.01	0.0021	0.0815	0.9185	9.07	0.210
D	0.21	0.0213	0.826	0.1739	4.11	0.101
E	0.0965	4.26×10^{-3}	0.165	0.8348	2.55	0.0442
F	0.0965	4.26×10^{-3}	0.165	0.8348	2.55	0.0442
G	0.0965	4.26×10^{-3}	0.165	0.8348	2.55	0.0442
H	0.0965	4.26×10^{-3}	0.165	0.8348	2.55	0.0442
I	0.0965	4.26×10^{-3}	0.165	0.8348	2.55	0.0442

The accuracy of the rare event approximation decreases when events have larger probabilities, since the error terms are the products of probabilities. Hence, using

the rare event approximation in this case will be overestimating the probability of failure, since 4 of the basic events have probability of occurrence greater than 0.5. The exact value would be difficult to obtain without using computer assisted calculations. However, the minimal cutset upper bound was calculated to be 0.00870 using Equation 2.4.

4.2.5 Interpretation of importance measures

The importance measures provide information about the basic events of the fault tree. They can be used to make recommendations about which basic events should be the focus of improvements. It would be desirable to improve the reliability of all basic events, but in practice, budget or time limitations often require a more focused approach. The following will highlight examples of how importance measures can be used to derive increasingly smaller subsets of basic events which should be prioritised for improvement work.

An example of using relative Fussell-Vesely (FV) importance measures to derive a subset of prioritised basic events, could be to only work on improving events which contribute towards more than 10% of all system failures. In order to implement such a policy, the relative FV values for each basic event can be used, since the relative FV value of a basic event X represents the proportion of failures where the occurrence of basic event X contributed to system failure. Choosing to only improve (reduce the probability of occurrence) basic events which have relative FV values of more than 0.1 would remove from the choice, all basic events that contribute to less than 10% of system failures, leaving only events B, D, E, F, G, H, and I as candidates for improvement.

It may be desirable to further restrict the set of events. The risk reduction worth (RRW) importance measure can be used. For this purpose, the lower the RRW value of a basic event, the more worthwhile it would be to try and implement improvements to reduce it's probability of occurrence. A condition for event selection could be that events should only be improved, if their RRW is less than 0.5 (ensuring the non-occurrence of those basic events would at least halve the probability of top event occurrence). Events that satisfy both the RRW and relative FV conditions stipulated thus far are B and D.

To narrow the two events down to a single event, the RAW or BM importance measures could be used. The BM importance measure of a basic event X represents the difference in top event probability with the probability of X occurring set at 0, and 1. A larger BM value represents a large sensitivity, meaning that there is more potential value from improving the basic event. Of the two candidate events, event B has the larger BM value. This event (the final accuracy check), should be the focus of improvements to safety.

4.2.6 Summary

Completing the FTA has been a useful exercise for modelling failure modes of the dispensing process. The initial implementation has demonstrated an ability to derive a subset of basic events which should be prioritised for improvements to improve the reliability of the dispensing process.

However, calculating importance measures and top event probabilities has relied on the assumption that the occurrence of each of the basic events are independent. This assumption may not be valid. For example, consider the probability of the final accuracy check failing to spot an error. This may not be independent of the basic events related to the picking, because a final accuracy check might be considerably more likely to spot an erroneous prescription, if multiple errors are present, compared to a prescription where only one error is present. Additionally this modelling technique has little scope for considering how prescriptions can become delayed throughout the dispensing process.

4.3 Markov Modelling the efficiency of the dispensing process

This subsection of the thesis presents the development of four Markov models used to model the availability of a community pharmacy. The objective of the modelling is to evaluate the steady state availability of the dispensing process, and to evaluate Markov modelling as a method for studying community pharmacy dispensing. Availability is defined as having a member of staff present and working on each of three key stages of the dispensing process. In the following models,

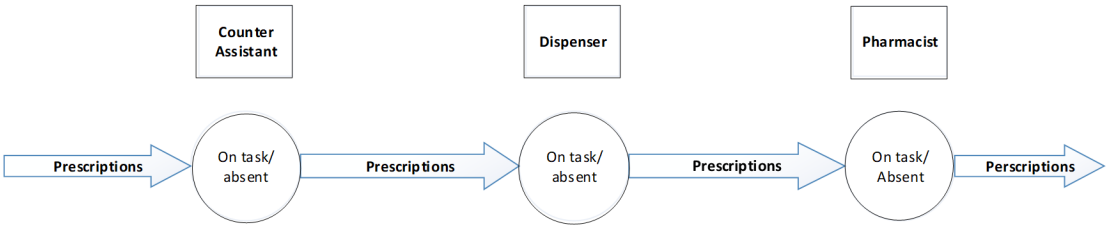
this means having a member of staff present and ready to receive prescriptions from patients, a dispenser dispensing prescriptions, and a pharmacist present to accuracy check prescriptions.

4.3.1 Initial model

The staff roles used in community pharmacy mean that members of staff will typically be assigned one of the main tasks to complete throughout the day. For example, dispensers are often responsible for dispensing, counter assistants receive prescriptions from patients, and pharmacists (and ACT's) are the only members of staff capable of clinical and accuracy checking prescriptions. Using the observation that individual staff may become unavailable throughout the day due to other non-dispensing responsibilities, a Markov model is developed to model community pharmacy availability.

Figure 4.3 shows a prescription flow diagram, which was developed into a Marokov model by considering the way in which different staff move prescriptions through stages of the dispensing process in pharmacies. The process has been simplified to 3 stages, where each member of staff moves prescriptions through their stage and onto the next.

FIGURE 4.3: Example Markov model for modelling availability of community pharmacy dispensing



This system has eight (2^3) possible states. Using the assumption that staff do not change task, staff can be seen to either help with or restrict the flow of prescriptions. The states are generated by having absent or present staff, where the full set of states is all the possible combinations. For example, one state can be generated by having all three staff members present. In this state, prescriptions can flow freely into and out of the pharmacy. Conversely, the situation where all three staff members are absent represents a state where prescriptions cannot move. Intermediate states exist where some staff are present and others are absent. Consider the state (absent, present, present), i.e. the counter assistant is absent, the

dispenser is present, and the pharmacist is present. In this state new prescriptions cannot enter the system, however any prescriptions that have already entered the system by passing through the reception already, can leave the system. The pharmacies' full availability is represented by the state where all members of staff are present. Any state in which at least one of the stages has no present member of staff represent unavailable states.

4.3.2 Modelling assumptions

This section shows the key assumptions used to model the dispensing process in a Markov modelling framework:

1. Staff are assigned either a role or a set of roles. Once assigned to roles they do not change or contribute to the completion of other tasks.
2. Staff assigned more than 1 task split their time equally between the two tasks.
3. The exponential distribution is used to model the times at which staff leave and return to their positions. Which implies further assumptions:
    - The times at which staff leave their positions are independent.
    - The time taken for a member to leave their post is always greater than 0.
    - The rate at which staff take breaks or go absent is constant throughout the day.
    - Two members of staff cannot change their status (present or absent) simultaneously.

4.3.3 Staff configurations

The performance of four different staff configurations will be compared using the Markov modelling framework. The four configurations are as follows:

1. A single pharmacist is responsible for all tasks.

2. A pharmacist checks accuracy, a dispenser is dispensing and receiving.

3. A pharmacist is checking accuracy, 1 dispenser is dispensing, and 1 counter assistant receiving prescriptions.

4. A pharmacist and ACT are checking accuracy, 1 dispenser is dispensing, and a counter assistant is receiving.

For configurations with fewer than 3 members of staff, some staff members will be transitioning between being absent and being on one of the tasks they have been assigned. In the case of the single pharmacist (configuration 1), the pharmacist will be transitioning between being absent, and being present on one of the three tasks. Similarly, for the second configuration, the dispenser will be randomly switching between being absent or on task dispensing/receiving prescriptions. These configurations will not produce a situation where the flow of prescriptions is uninterrupted throughout the pharmacy, but they may give a useful comparison.

4.3.4 System states

This section will be used to describe the states used for each staff configuration. The states used in Markov chains form the backbone of the analysis, as they describe the possible situations that arise in the pharmacy. Every member of staff in each configuration can either be on task or absent. "Absent" here does not mean that they are not in the pharmacy (i.e. didn't show up to work for the day), it is meant to represent a situation where the member of staff is present in the pharmacy but occupied with another non-dispensing related activity at the current moment. Tables 4.2-4.5 show the full set of system states for the 4 staff configurations.

Table 4.2: System states of configuration 1

States	Description	Result on pharmacy
1	Pharmacist absent	(No checking, dispensing, or receiving)
2	Pharmacist on task, receiving prescriptions	(No checking or dispensing)
3	Pharmacist on task, dispensing	(No reception or checking)
4	Pharmacist on task, accuracy checking	(No dispensing or reception)

Table 4.3: System states of configuration 2

States	Description	Result on pharmacy
1	Pharmacist on task, dispenser on task receiving prescriptions	(No dispensing)
2	Pharmacist on task, dispenser on task dispensing	(No reception)
3	Pharmacist on task, dispenser absent	(No dispensing or reception)
4	Pharmacist absent, dispenser on task receiving prescriptions	(No checking or dispensing)
5	Pharmacist absent, dispenser on task dispensing	(No checking or reception)
6	Pharmacist absent, dispenser absent	(No checking, dispensing or reception)

Table 4.4: System states of configuration 3

States	Description	Result on pharmacy
1	All 3 staff on task	(Fully functioning)
2	Pharmacist absent	(No Checking)
3	Dispenser absent	(No dispensing)
4	Counter assistant absent	(No reception)
5	Pharmacist and dispenser absent	(No checking or dispensing)
6	Pharmacist and counter assistant absent	(No checking or reception)
7	Counter assistant absent, dispenser absent	(No dispensing or reception)
8	All absent	(No checking, dispensing or reception)

Table 4.5: System states of configuration 4

States	Description	Result on pharmacy
1	All 4 staff on task.	(Fully functioning)
2	Pharmacist absent, other 3 on task.	(Fully functioning)
3	Dispenser absent, other 3 on task.	(No dispensing)
4	Counter assistant absent, other 3 on task.	(No reception)
5	ACT absent, other 3 on task.	(Fully functioning)
6	Pharmacist and dispenser absent, counter assistant and ACT on task.	(No dispensing)
7	Pharmacist and counter assistant absent. ACT and dispenser on task.	(No reception)
8	Pharmacist and ACT absent, dispenser and counter assistant on task	(No checking)
9	Pharmacist and ACT on task, dispenser and counter assistant absent.	(No dispensing or reception)
10	Pharmacist and counter assistant on task, dispenser and ACT absent.	(No dispensing)
11	Pharmacist and dispenser on task, counter assistant and ACT absent.	(No reception)
12	Counter assistant on task, other 3 absent.	(No checking or dispensing)
13	Dispenser on task, other 3 absent.	(No checking or reception)
14	Pharmacist on task, other 3 absent.	(No dispensing or reception)
15	ACT on task, other 3 absent.	(No dispensing or reception)
16	All absent.	(No tasks being completed)

Each state represents a situation of the prescription flow inside a pharmacy. If there is no staff responsible for one part of the three main dispensing processes (prescription reception, dispensing and accuracy checking), then the model is taken to have no output of that task for the period of time that the process is in that state. These states are the unavailability of the tasks which are not being completed. In some configurations it requires two staff to be absent for this to be the case, namely in configuration 4, as there is an ACT acting as a redundancy for the accuracy checking stage. Both of the staff responsible for a section of the dispensing process must be absent to restrict the flow of prescriptions through their station of work.

In this model, the pharmacy dispensing system changes state whenever an individual member of staff changes their state, between being on task/absent. The Markov modelling requirement (the Markov property) states that if the current state of the system is known, the following state that the system will enter is independent of the history of the process. A way to ensure this property is met is to have transition rates between states governed by exponential probability distributions. The exponential distribution is memoryless (independent of the past), so a process governed entirely by this distribution will satisfy the Markov property. For further information on the exponential distribution, see Section 2.4.7.2.

4.3.5 Transition Rates

There are 8 rates that need to be described for this group of Markov chains. These are the rate at which each member of staff leaves and returns to their posts. Table 4.6 shows the 8 transition rates used, where each rate corresponds to λ in an exponential distribution. The rates are in minutes, if $\lambda = 1/6$ for the rate of pharmacists leaving their post, this corresponds to them leaving their post every 6 minutes on average.

Table 4.6: Staff transition rates

Member of staff	Rate of leaving post	Rate of returning to post
Pharmacist	$P = 1/6$	$p = 1/5$
Dispenser	$D = 1/10$	$d = 1/3$
ACT	$A = 1/15$	$a = 1/3$
Counter assistant	$C = 1/10$	$c = 1/4$

The rates have been chosen such that the pharmacist is taken away from their position in the prescription production chain more frequently and for longer than other staff. This is based on the assumption that pharmacists have more non-dispensing activities, such as advanced services or patient consultation to complete. The rate of return has been set to be lower for the pharmacist, based on the assumption that these tasks will take longer than the shorter breaks taken by other members of staff.

The methodology used to generate results was to first construct the graphs for each staff configuration, and apply the rates of individual staff transitions to produce transition matrices. The technique used for writing the rate transition matrices was inspection of the graphs representing the configurations. The transition rates

are implied by the states that transitions occur between. For example, in configuration 3, the transition rate from state 1 to state 2 is the rate at which a dispenser will switch from being on task to absent.

4.3.6 Results

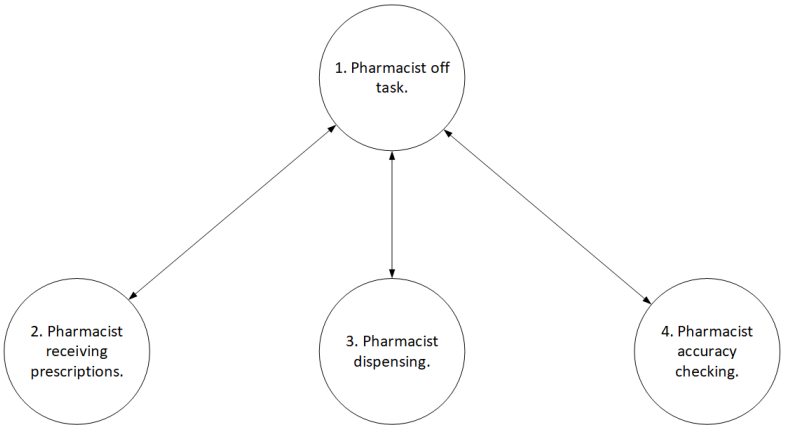
A numerical solving program was coded in Matlab to find the probability of being in each system state as a function of time. The program was run using 4 matrices, each are described in Sections 4.3.6.1 - 4.3.8, along with initial conditions for each model, i.e. all the staff members are present. Note that the algebraic version of the transition matrix for configuration 4 is omitted due to its size, the numerical matrix is given.

4.3.6.1 Configuration 1

Figure 4.4 shows the graph of the Markov model for configuration 1, and Equation 4.10 shows the corresponding rate transition matrix.

$$Q_1 = \begin{matrix} -p & \frac{p}{3} & \frac{p}{3} & \frac{p}{3} \\ P & -P & 0 & 0 \\ P & 0 & -P & 0 \\ P & 0 & 0 & -P \end{matrix} = \begin{matrix} -\frac{1}{5} & \frac{1}{15} & \frac{1}{15} & \frac{1}{15} \\ \frac{1}{6} & -\frac{1}{6} & 0 & 0 \\ \frac{1}{6} & 0 & -\frac{1}{6} & 0 \\ \frac{1}{6} & 0 & 0 & -\frac{1}{6} \end{matrix} \tag{4.10}$$

FIGURE 4.4: Configuration 1

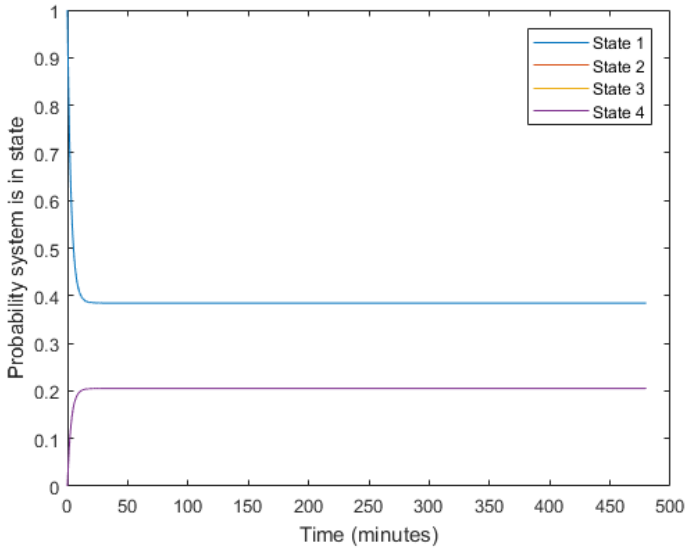


The steady state distribution for configuration 1 is $\pi = (0.38462, 0.20513, 0.20513, 0.20513)$. Being absent has the largest probability, since the pharmacist must return to this state before changing task. The other three states are equal. Over the course of

a day, the pharmacist spends 20.5% of his time on each dispensing related tasks and the rest on non-dispensing related activities, 38.5%.

The numerical solver was run with an initial condition of q0 = [1, 0, 0, 0]. From this state it took 1788 seconds (29 minutes and 48 seconds) for the system to reach a steady state distribution. A graph of the plot of the numerical solution is shown below in Figure 4.5. Only 2 lines appear since 3 of the states had the same steady state probability.

FIGURE 4.5: Probability of system states against time



4.3.6.2 Configuration 2

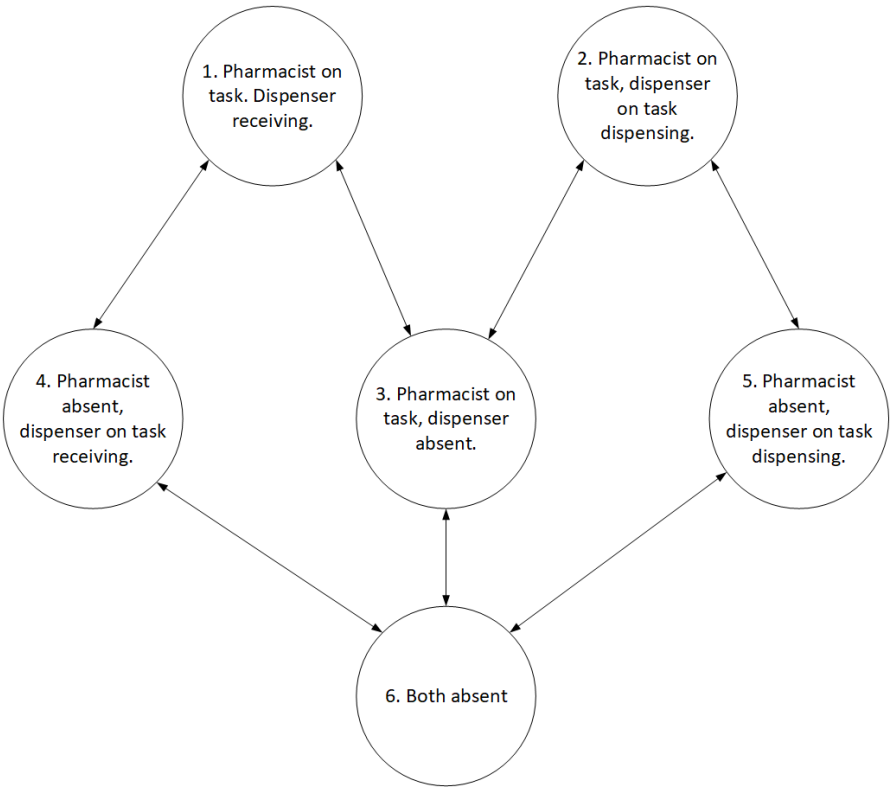
Figure 4.6 shows the graph of the Markov model for configuration 2, and Equation 4.11 shows the corresponding rate transition matrix.

$$Q_2 = \begin{matrix} -(d+p) & 0 & d & p & 0 & 0 \\ 0 & -(d+p) & d & 0 & p & 0 \\ \frac{D}{2} & \frac{D}{2} & -(D+p) & 0 & 0 & p \\ P & 0 & 0 & -(P+d) & 0 & d \\ 0 & P & 0 & 0 & -(P+d) & d \\ 0 & 0 & P & \frac{D}{2} & \frac{D}{2} & -(P+D) \end{matrix} = \begin{matrix} -\frac{4}{15} & 0 & \frac{1}{10} & \frac{1}{6} & 0 & 0 \\ 0 & -\frac{4}{15} & \frac{1}{10} & 0 & \frac{1}{6} & 0 \\ \frac{1}{6} & \frac{1}{6} & -\frac{1}{2} & 0 & 0 & \frac{1}{6} \\ \frac{1}{5} & 0 & 0 & -\frac{3}{10} & 0 & \frac{1}{10} \\ 0 & \frac{1}{5} & 0 & 0 & -\frac{3}{10} & \frac{1}{10} \\ 0 & 0 & \frac{1}{5} & \frac{1}{6} & \frac{1}{6} & -\frac{8}{15} \end{matrix} \quad (4.11)$$

The steady state distribution for configuration 1 is: $\pi = (0.23669, 0.23669, 0.14201, 0.14793, 0.14793, 0.088757)$.

More can be said about the prescription flow in this configuration than in configuration 1. Once the system has reached the steady state, both members are on task

FIGURE 4.6: Configuration 2



for 47.3% of the time (probability of states 1 and 2). Similarly to configuration 1, it is also true for configuration 2 that the pharmacist is off task for 38.5% of system operation. However, in this configuration the pharmacist spends the remainder of their time checking.

Considering how patients are able to hand in prescriptions in this configuration, it can be seen that the dispenser is absent from receiving prescriptions for 61.5% (1 - 0.24 – 0.14) of the time. This is mainly due to only the dispenser being able to receive prescriptions and their work load is being split between two jobs (receiving prescriptions and dispensing). The dispenser is off task for 23.1% of system operation (States 3 and 6). The most undesirable state (State 6), where neither member of staff is working on dispensing related activities, only occurs for 8.8% of the time. For this configuration, at steady state:

P(No checking) = P(states 4, 5 and 6) = 0.38461

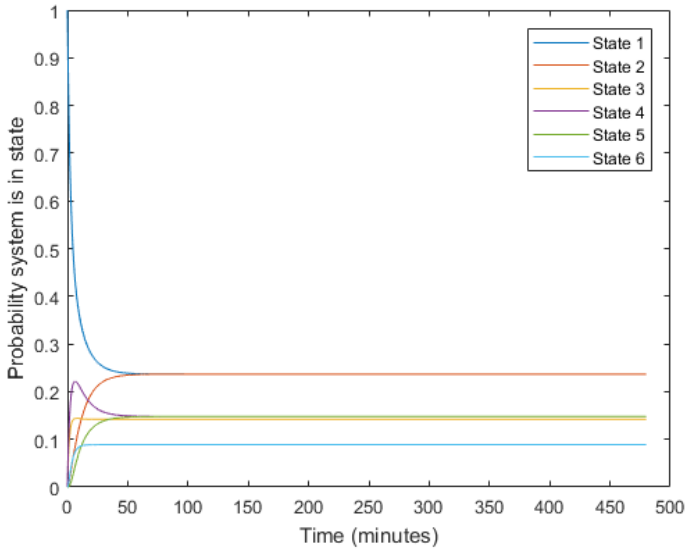
P(No dispensing) = P(states 1, 3, 4 and 6) = 0.769233

P(No reception) = P(states 2, 3, 5 and 6) = 0.61538

Configuration 2 is a better configuration for customers, the probability of their prescription being taken in when the customer walks into the pharmacy is 0.38462 (1-P(No reception)), as compared with 0.20513 in configuration 1. The probability of the other stations being "open" is also higher. Checking accuracy being "open" is 0.61539 in configuration 2, compared to 0.20513 in configuration 1. Similarly dispensing is more likely to be open in configuration 2, 0.230767 in configuration 2 against 0.20513 in configuration 1.

The numerical solver was run with the initial condition $q_0 = [1, 0, 0, 0, 0, 0]$. The system took 5448 seconds (1 hour, 90 minutes and 48 seconds) to reach a steady state. The graph of the probability of each state is shown below in Figure 4.7.

FIGURE 4.7: Probability of system states against time for configuration 2



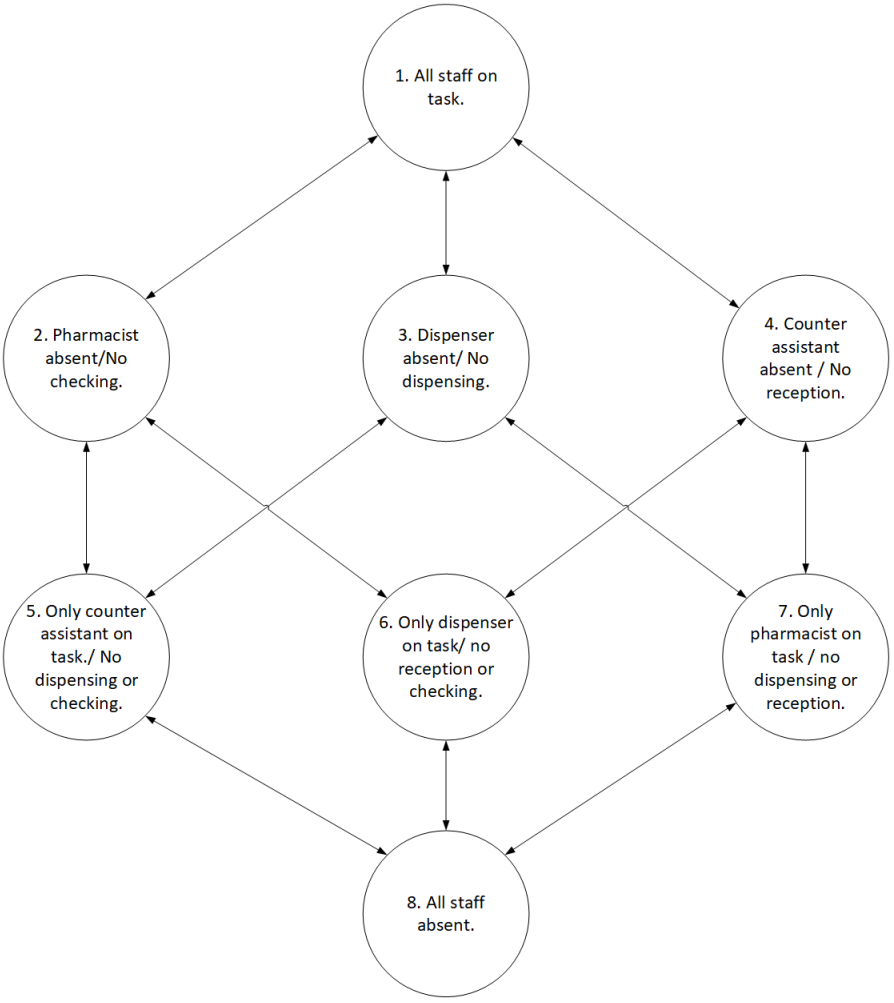
4.3.7 Configuration 3

Figure 4.8 shows the graph of the Markov model for configuration 3, and Equation 4.12 shows the corresponding rate transition matrix.

$$Q_3 = \begin{array}{c} \\ -(p+c+d) \\ P \\ D \\ C \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} \begin{array}{cccccccc} p & d & c & 0 & 0 & 0 & 0 \end{array} \\ \begin{array}{cccccccc} -(P+d+c) & 0 & 0 & d & c & 0 & 0 \end{array} \\ \begin{array}{cccccccc} 0 & -(D+p+c) & 0 & p & 0 & c & 0 \end{array} \\ \begin{array}{cccccccc} 0 & 0 & -(C+p+d) & 0 & p & d & 0 \end{array} \\ \begin{array}{cccccccc} D & P & 0 & -(P+D+c) & 0 & 0 & c \end{array} \\ \begin{array}{cccccccc} C & 0 & P & 0 & -(P+d+C) & 0 & d \end{array} \\ \begin{array}{cccccccc} 0 & C & D & 0 & D & -(p+D+C) & p \end{array} \\ \begin{array}{cccccccc} 0 & 0 & 0 & C & 0 & P & -(C+D+P) \end{array} \end{array}$$

$$= \begin{array}{cccccccc} -\frac{11}{30} & \frac{1}{6} & \frac{1}{10} & \frac{1}{10} & 0 & 0 & 0 & 0 \\ \frac{1}{5} & -\frac{2}{5} & 0 & 0 & \frac{1}{10} & \frac{1}{10} & 0 & 0 \\ \frac{1}{3} & 0 & -\frac{3}{5} & 0 & \frac{1}{6} & 0 & \frac{1}{10} & 0 \\ \frac{1}{4} & 0 & 0 & -\frac{31}{60} & 0 & \frac{1}{6} & \frac{1}{10} & 0 \\ 0 & \frac{1}{3} & \frac{1}{5} & 0 & -\frac{19}{30} & 0 & 0 & \frac{1}{10} \\ 0 & \frac{1}{4} & 0 & \frac{1}{5} & 0 & -\frac{11}{20} & 0 & \frac{1}{10} \\ 0 & 0 & \frac{1}{4} & \frac{1}{3} & 0 & 0 & -\frac{3}{4} & \frac{1}{6} \\ 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{3} & \frac{1}{5} & -\frac{47}{60} \end{array} \tag{4.12}$$

FIGURE 4.8: Configuration 3



The steady state solution was found to be $\pi = (0.33812, 0.21133, 0.10144, 0.13525, 0.063398, 0.084531, 0.040575, 0.025359)$. This is the first configuration to contain

an allocated member of staff for each task. In this configuration, the pharmacist is away from their post, again, for 38.5% of the time (States 2, 5, 6, and 8), and similarly the dispenser is off task for 23.1% of the time. The counter assistant is off task for 28.6% of the systems operation time. If a customer walks into the pharmacy at a random time, when it has reached the steady state probabilities:

P(No checking) = P(State 2 ∪ State 5 ∪ State 6 ∪ State 8) = 0.21133 + 0.063398 + 0.084531 + 0.025359 = 0.3846

P(No dispensing) = P(State 3 ∪ state 5 ∪ state 7 ∪ state 8) = (0.10144 + 0.063398 + 0.040575 + 0.025359) = 0.2308

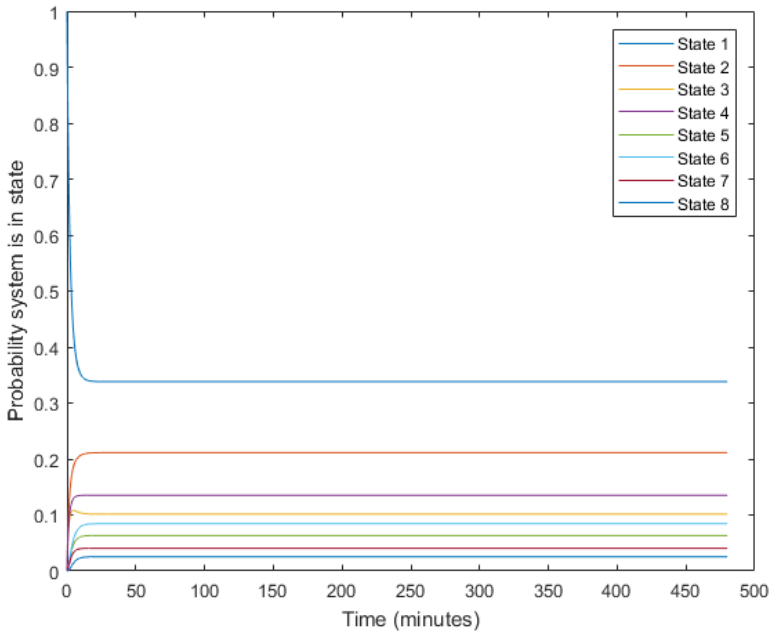
P(No reception) = P(state 4 ∪ state 6 ∪ state 7 ∪ state 8) = (0.13525 + 0.084531 + 0.040575 + 0.025359) = 0.2857

P(All available) = 0.33812

Customers are significantly more likely to be able to hand in their prescriptions when they arrive at the pharmacy in configuration 3, than in 1 or 2. With a member of staff dedicated to each task, the probability that the counter assistant will be at the counter when a customer arrives is 0.7143.

The initial condition was set to be q0 = [1, 0, 0, 0, 0, 0, 0, 0] and when solving the system by approximation it took 1344 seconds to reach a steady state. The graph of the approximations generated is shown below in Figure 4.9.

FIGURE 4.9: Probability of system states against time for configuration 3



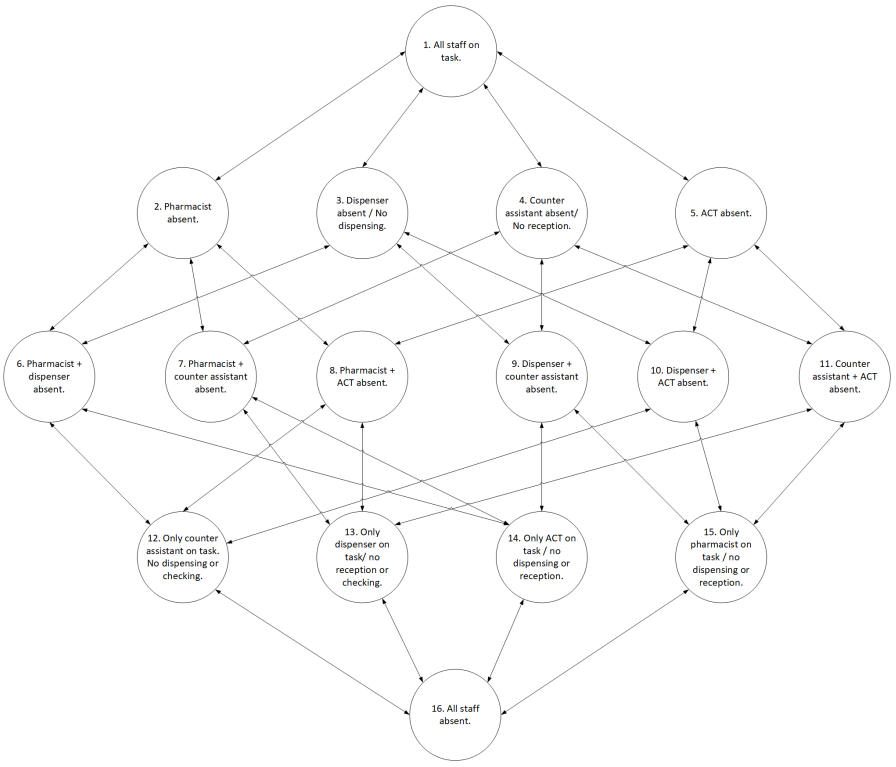
4.3.8 Configuration 4

Figure 4.10 shows the graph of the Markov model for configuration 4, and Equation 4.13 shows the corresponding rate transition matrix.

$$Q_4 = \begin{bmatrix}
-\frac{13}{30} & \frac{1}{6} & \frac{1}{10} & \frac{1}{10} & \frac{1}{15} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{5} & -\frac{7}{15} & 0 & 0 & 0 & \frac{1}{10} & \frac{1}{10} & \frac{1}{15} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & 0 & -\frac{2}{3} & 0 & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{10} & \frac{1}{15} & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & 0 & 0 & -\frac{37}{60} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{10} & 0 & \frac{1}{15} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{3} & 0 & 0 & 0 & -\frac{7}{10} & 0 & 0 & \frac{1}{6} & 0 & \frac{1}{10} & \frac{1}{10} & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{1}{3} & \frac{1}{5} & 0 & 0 & -\frac{7}{10} & 0 & 0 & 0 & 0 & 0 & \frac{1}{15} & 0 & \frac{1}{10} & 0 & 0 \\
0 & \frac{1}{4} & 0 & \frac{1}{5} & 0 & 0 & -\frac{37}{60} & 0 & 0 & 0 & 0 & 0 & \frac{1}{15} & \frac{1}{10} & 0 & 0 \\
0 & \frac{1}{3} & 0 & 0 & \frac{1}{5} & 0 & 0 & -\frac{11}{15} & 0 & 0 & 0 & \frac{1}{10} & \frac{1}{10} & 0 & 0 & 0 \\
0 & 0 & \frac{1}{4} & \frac{1}{3} & 0 & 0 & 0 & 0 & -\frac{49}{60} & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{15} & 0 \\
0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 & -\frac{14}{15} & 0 & \frac{1}{6} & 0 & 0 & \frac{1}{10} & 0 \\
0 & 0 & 0 & \frac{1}{3} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & -\frac{17}{20} & 0 & \frac{1}{6} & 0 & \frac{1}{10} & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{5} & 0 & -\frac{29}{30} & 0 & 0 & 0 & \frac{1}{10} \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{4} & 0 & 0 & \frac{1}{5} & 0 & -\frac{53}{60} & 0 & 0 & \frac{1}{10} \\
0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{3} & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 & -\frac{17}{20} & 0 & \frac{1}{15} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & \frac{1}{4} & \frac{1}{3} & 0 & 0 & 0 & -\frac{13}{12} & \frac{1}{6} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{3} & \frac{1}{3} & \frac{1}{5} & -\frac{67}{60}
\end{bmatrix} \tag{4.13}$$

In this configuration there is an ACT assisting the accuracy checking of prescriptions in the pharmacy. If the pharmacist leaves their post, then the ACT will continue to check in their absence. The steady state solution of this configuration

FIGURE 4.10: Configuration 4



was found to be π = (0.28177, 0.17611, 0.084531, 0.11271, 0.056354, 0.052832, 0.070442, 0.035221, 0.033812, 0.016906, 0.022542, 0.010566, 0.014088, 0.021133, 0.0067625, 0.0042265)

This is the first configuration to contain a redundant element in the system, the accredited checking technician. The results of the inclusion of an ACT into the process, backing up the pharmacist if they ever need to attend to a non-dispensing activity, can be seen by comparing the probabilities that no checking is occurring between the configurations. In configuration 3, the probability of no checking being done when a customer walks in was 0.3846. Now in configuration 4:

P(No checking) = P(states 8, 12, 13 and 16) = 0.0641

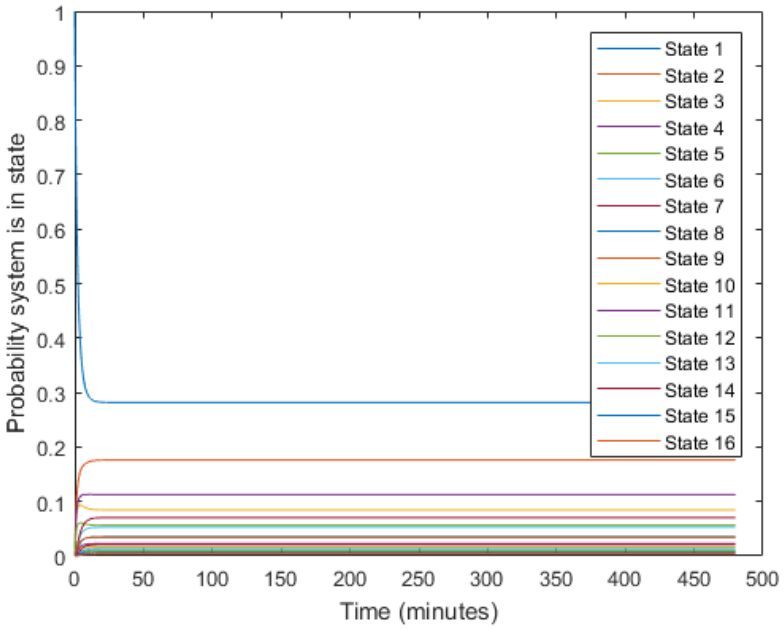
P(No dispensing) = P(states 3, 6, 9, 10, 12, 14, 15 and 16) = 0.2308

P(No reception) = P(states 4, 7, 9, 11, 13, 14, 15 and 16) = 0.2857

This is a considerable improvement between the two configurations. The down-time (time there is no-one accuracy checking prescriptions) of the checking process has been reduced by 83.3%. By adding redundant members of staff into the system at other points in the process, similar reductions could be made. The same initial

condition was used for the approximation (starting with all staff on task), and the graph of results is shown below in Figure 4.11.

FIGURE 4.11: Probability of system states against time for configuration 3



4.3.9 Summary

This modelling has demonstrated that Markov models can be useful for analysing the impact of adding redundancy into systems, as presented in configuration 4. The main limitation of Markov modelling is the restriction of adhering to the memoryless property. This requires that all the transitions between states be controlled by exponential distributions, which may not be the best variable to use if in field data suggests alternative distributions are preferable (See Chapter 5 for examples of this).

An additional problem with the Markov model presented in this Chapter is the inability to incorporate failures at an individual prescription level. The failures in the Markov model explored, come in the form of members of staff being away from their post, and the resultant effect this has on the pharmacies ability to dispense. These types of failures are of interest, but it would be much better if both types of failures could be analysed within the same modelling framework. In Chapter 6 Coloured Petri nets are used to model a community pharmacy. The

model includes how individual prescriptions can become delayed, and also tracks how each member of staff spends their time.

In conclusion, Markov models are useful for modelling substructures of the overall pharmacy system, but are not very flexible, and may therefore struggle to model a version of the dispensing process with many concurrent processes.

4.4 Conclusion

This Chapter has presented two attempts to apply well-known reliability engineering techniques to the process of community pharmacy dispensing. These exercises acted as feasibility studies for the two reliability models, fault trees and Markov models. Some similar work completed as a part of these studies was found to have already been done in the literature, notably the fault tree presented in (Cohen et al., 2012).

This initial modelling illustrates that, Fault trees and Markov models can be used to model some aspects of the dispensing process. However, both modelling frameworks may not be well suited to model both reliability and efficiency aspects simultaneously. Markov modelling is limited in this respect, since the number of states required to model a system grows exponentially as the number of components increases, and fault trees are more suited to reliability. Based on this initial exploration of the two techniques it was decided that this thesis would focus on Petri Nets, as they are a more suitable modelling framework.

Chapter 5

Data Collection and Interpretation

5.1 Introduction

This chapter explains and interprets in field data, collected from 4 UK community pharmacies. A mixed methods investigation using observational data collection and semi-structured qualitative interviews was undertaken as part of this thesis, with the aim of informing reliability models of community pharmacies. Ethical approval for the study was granted by the Nottingham Medicine School on the 24th of August 2017, which lasted for a year from the date it was granted. Quantitative data was collected by observing the community pharmacy dispensing process and timing how long six key stages of the dispensing process take. For 4 stages, label generation, picking, applying labels, and final accuracy check, the number of items in the prescription being processed was also recorded. Qualitative interviews were carried out with staff working in the pharmacies to gain an understanding of the dispensing process from the perspective of practitioners. The interviews produced a number of potential adjustments to the model developed in Chapter 6 to make the model match up with practitioners' understanding of the process.

This Chapter is presented as follows: Section 5.2 gives a brief outline of each of the sites, Section 5.3 presents the findings of the quantitative timings of the process, and Section 5.4 gives an analysis of the qualitative interviews. Section 5.5 gives clear implications of the previous two sections on the CPN model developed in Chapter 6, and Section 5.6 concludes the Chapter.

5.2 Site overview

After the observation period at each pharmacy, observational profiles were written describing the key characteristics of the site. These included: information about when the study took place, the number of staff working and their qualifications, the method of dispensing observed, and an outline of any key events that took place during the observations. This data was collected with the aim of informing models of community pharmacy dispensing. Where possible, the observational profiles were written shortly after the observation period took place. The full observational profiles are included in Appendix A.

Table 5.1: Overview of 4 community pharmacy sites

Pharmacy	Type	No. Staff	Setting	Observation length
A	Large Multiple	3-6	Co-located	3 days
B	Large Multiple	3-8	Residential	4 days
C	Independent	3-8	Residential	5 days
D	Independent	4-7	Commercial	5 days

Table 5.1 provides key information about the sites visited. The designation of a community pharmacy as either a Large Multiple or an Independent is a reference to the size of the business the store is a part of. A pharmacy which is a member of a large multiple is one store which is run by a larger headquarters responsible for multiple stores around the country, whereas an independent pharmacy is a single store owned and run by a pharmacist. The number of staff seen working in each store refers to the minimum and maximum number of staff seen to be working in the pharmacy at any time. The variations in staff levels were due to people going for lunch breaks, or stores employing fewer people during quieter parts of the day. Three types of setting were identified; co-located signifies the pharmacy was located in very close proximity to a general practitioner care facility, residential settings were those where stores were mainly surrounded by houses, and a commercial setting indicates that the surrounding area was mostly used by other businesses. For additional details about the pharmacy sites visited, see the observational profiles included in Appendix A.

5.3 Quantitative data collection and analysis

5.3.1 Introduction

This section presents the interpretation of the quantitative data collected from the 4 sites. The desired outcome is to be able to specify which probability distributions best describe the time to complete a stage of the process. These can be used to accurately simulate pharmacy staff completing dispensing tasks. Furthermore, the analysis will investigate how the number of items contained in prescriptions affects these distributions. Quantitative timing data was collected for up to six tasks in each pharmacy (also the same stages as in Chapter 6, Figure 6.1). These were:

1. Prescription reception/initial interaction
2. Label generation
3. Picking medicines
4. Applying labels
5. Accuracy checking
6. Handing over/storing medication.

Tasks from 2 till 5 were assumed to be dependent on the number of items in a prescription, so when collecting timing data for each of these tasks, the number of items in the prescription being processed was also collected. This was done either by observing the script being used and counting the number of items, or in some cases co-operative staff would call out how many items were included in the prescription that they were working on. All statistical analysis of quantitative data was carried out in R (R Core Team, 2017), using the library fitdistplus (Delignette-Muller and Dutang, 2015). Data sets were tested for potential fits using Q-Q plots for 6 common probability distributions: normal, uniform, exponential, lognormal, Weibull, and gamma.

For the purposes of the analysis 6 categories of prescription were considered: these were prescriptions with 1, 2, 3, and 4 items, medium sized prescriptions (5-8 items), and large prescriptions (8 or more items). For the purposes of this analysis, each category of prescription will be referred to as a segment of the data.

5.3.2 Method: data collection

The data collection technique for the duration of each task in the dispensing process was observation and stop watch timing. Staff were observed working in the pharmacy during working hours, the stop watch would be started when an event was initiated, and ended when an event was completed. Short descriptions of the timing process for each stage are given below.

5.3.2.1 Prescription reception/initial interaction

If a customer is waiting at the counter, the timing begins when the counter assistant (or other member of staff) begins an interaction with the customer. When the interaction ends and the customer is no longer in need of attention, the timing ends. Data collection of this stage can include more actions than handing in a prescription, because it is difficult to identify the intention (or need) of a customer waiting at a community pharmacy counter before any interaction takes place, and customers commonly do multiple actions at once. The phenomena included were: customers picking up their pre-dispensed EPS prescriptions, customers buying non-prescription medicine or other products from the shop, customers who knew the counter assistant engaging them in conversation, customers handing in prescriptions to be dispensed, or any possible combination of the above actions. Other phenomena were included since it was very common for some kind of interaction to take place between the customer and pharmacy staff at each encounter. The timing of prescription reception represents how long it takes for an initial interaction between a patient and member of staff to conclude, for any of the above activities.

5.3.2.2 Label generation

Data collection of this stage of the process mostly focused on EPS prescriptions since these were done consecutively, making it simpler to identify when staff were using the computer to generate labels, rather than using it for other general purposes. Timing began when a dispenser picked up a prescription form, it included all the time spent inputting commands into the computer, and ended when the prescription form was placed back into a basket, along with the printed labels. The labels for a single prescription are all printed off item by item. A strip of

labels emerges from the label printer, becoming longer as more labels are printed. When labels for all the items in the prescription have been printed, the whole strip is torn off, and folded up inside the prescription form. The action of ripping off the strip of labels provided an audible indication to stop the stopwatch.

5.3.2.3 Picking medicines

Data was collected on the time taken to pick medicines by observing dispensers as they pick the items for prescriptions. The timing begins when a dispenser picked up a basket containing a prescription form with corresponding labels, and ends when the dispenser either passed the full basket over to the pharmacist's work bench, or placed it in a designated area where interim picked prescriptions were stored.

Items for a prescription are fetched in batches, of two or three items at a time, or one at a time, depending how large or how many of each of the items there are. The size is important since dispensers may only carry a limited amount of objects around a pharmacy. This limitation meant that sometimes multiple trips around the pharmacy were needed to gather all the items. This occasionally caused the timing process to be stopped prematurely, due to the observer assuming all the items had been gathered when they had not. These early stoppages, when identified, were not recorded.

5.3.2.4 Applying labels

Data was collected on the time taken to apply labels to medicine boxes by observing dispensers as they apply labels to items in prescriptions which they have picked. The timing begins when a dispenser picked up a basket containing a prescription form with corresponding items and labels in, and ends when the dispenser got to the end of the set of label stickers.

Labels are applied to the items in prescriptions one at a time, until all the items have been labelled. During this stage any split packs must be sorted out. Split packs arise when a prescription form has a number of tablets different from the default amount included in boxes. Splitting packs can prolong the process significantly and it caused many timings to be stopped, because initially the researcher was unsure what was going on. When the split pack process was understood, packs

that had to be split were not timed, since the process is very different to normal labelling and it was not considered to be a part of the labelling stage.

5.3.2.5 Accuracy checking

Data was collected on the time taken to accuracy check prescriptions by observing pharmacists or ACTs as they accuracy checked prescriptions. The timing would begin when a prescription was picked up to be checked, and would end when the items in the prescription were closed inside a white paper bag, and a label stuck on the outside.

Items in a prescription are checked one at a time. If a pharmacist is completing the check, it is often the case that a clinical check is done at the same time. During observations it was difficult to know if pharmacists were carrying out both checks simultaneously, or if only the accuracy checks were being performed. It could have been the case that some of the timings were how long it took to complete both checks together, rather than *only* the accuracy check. This was not a problem when observing ACTs, since they are only qualified to accuracy check. Both were recorded since there was a limited time to collect data, and in some pharmacies ACTs were completing the vast majority of accuracy checking.

5.3.2.6 Handing over/storing medication

Data was collected on the time taken to finalise the dispensing process. This was done by timing how long it took the pharmacist to either handover completed prescriptions to patients, or the time taken to place prescriptions in the storage area, in cases where patients were not waiting in the pharmacy to collect their prescription.

5.3.3 Method: Data analysis

This section describes the methods used to analyse the quantitative data collected during the observations. The data was initially entered as data frames in R, where each of the 6 tasks, highlighted in Section 5.3.1, had their own data frame for each pharmacy. During the analysis each site was treated independently, and

one additional analysis was done using combined data from pharmacies A and B, to produce a larger data set. This was done since the pharmacies were both members of the same large multiple, and had similar work patterns. The other two pharmacies were treated as separate sites since they were entirely independent stores. The analysis completed for each segmentation of the data was a 5-stage process, shown below:

1. Remove outliers, and generate descriptive statistics.
   Outliers were removed using the inter-quartile range method (remove data points further than 1.5*IQR (Inter quartile range) from the mean). Descriptive statistics of the mean and standard deviation were generated for each segmentation of the data.

2. Plot QQ plots for 6 common distributions.
   The QQ plots are used as a checking tool in this analysis to see whether the result produced by the distribution fitting analysis aligns with the Q-Q plots (See 2.4.1).

3. Determine best fitting parameters for each of the 6 distributions being tested, using the method of maximum likelihood estimation.
   A distribution fitting analysis was carried out for each segmentation of the data. This involved estimating optimal parameters to fit the data, for each of the six common probability distributions described in Section 2.4.7. Parameters for each candidate distribution were estimated using the method of maximum likelihood (See 2.4.2).

4. Identify which distribution is the best fit for the data, using the Akike Information Criterion (AIC) (Akaike, 2011), and check whether the Q-Q plots confirm this choice.
   The fitted distributions are compared using AIC (Akaike Information Criterion) values to determine which distribution is the most suitable (See Section 2.4.3).

5. Use a bootstrap sampling method to derive 95% confidence intervals for the estimated parameters.
   Uncertainties of the parameters estimated during the previous steps of the distribution fitting can be attained using bootstrap sampling (Delignette-Muller and Dutang, 2015) (See Section 2.4.6).

For brevity, the full set of graphs and statistics produced during this process are only presented for 3 segments of the data, the prescription reception in Pharmacy A, label generation of 1 item prescriptions in Pharmacy A, and label generation of large prescriptions in Pharmacy A. The results for the other stages are presented in a condensed table form in Section 5.3.9. Below each of the 5 steps are described in more detail.

5.3.4 Probability distributions

Six commonly known continuous probability distributions were used in the distribution fitting analysis of this study (See Section 2.4.7). Each subset of the data was tested for goodness-of-fit against these distributions using the method outlined in Section 5.3.3. These were chosen since they had either been used as assumed variables in the initial modelling (Chapter 4), or they are commonly occurring variables in other reliability engineering applications.

5.3.5 Analysis examples

To demonstrate how the data was analysed, the full analysis of three segments of the data are shown: prescription reception from Pharmacy A, label generation of single item prescriptions in Pharmacy A, and Label Generation of large prescriptions in Pharmacy A. The analysis of all other segments of the data was carried out in a similar manner to this. The results of the analysis for all of the quantitative timings collected are presented in table form in Section 5.3.9. The following subsection describes each of the five steps used to analyse the quantitative date. Note that confidence intervals (step 5) were not calculated for all parameters, but the method is presented.

5.3.6 Example 1: Prescription reception in Pharmacy A

A full set of quantitative data was collected for prescription reception stage in Pharmacy 1, 160 observations of 1 variable data, i.e. how long it took for the initial interaction between patient and staff to conclude.

FIGURE 5.1: Histogram of prescription reception

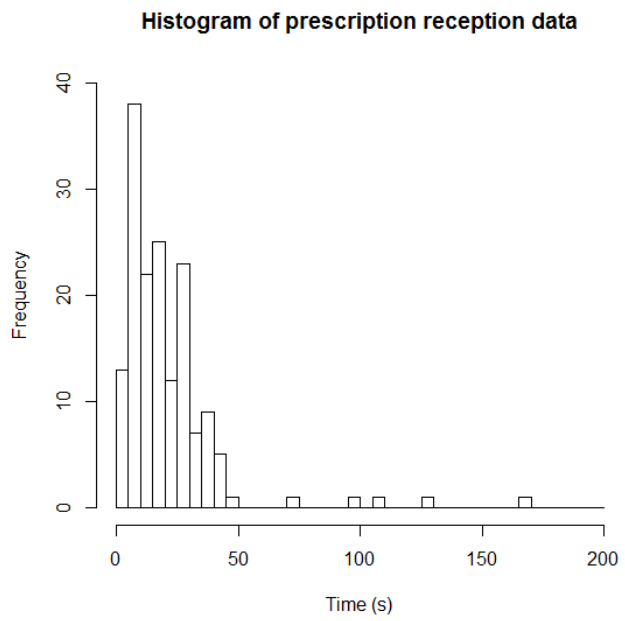


Figure 5.1 shows a histogram of the full data set, and Table 5.2 shows a summary of the data.

TABLE 5.2: Summary of prescription reception data: outliers removed

Pharmacy	Min	Q1	Median	Mean	Q3	Max
1	2.000	8.475	16.150	20.907	27.400	170.000

Figure 5.1 shows that the majority of observations were in the 0-50 second range, with only 5 observations occurring beyond 50 seconds. The mean time was 20.9 seconds. The 5 largest values were removed since they were classified as outliers, and the new mean time with outliers removed was 17.9 seconds.

Distribution fitting analysis of the time taken to receive prescriptions was conducted on the data set, Figure 5.2 shows the theoretical densities of the 6 distributions fitted to the data. The six AIC values associated with each distribution are shown in Table 5.3. Note that an AIC value cannot be calculated for the Uniform distribution, since the likelihood function is not twice differentiable.

The fitted Gamma distribution produced the lowest AIC value of the 5 distributions tested. To test the fit of each of the distributions, QQ plots for the data plotted against each distribution are shown in Figure 5.3. The main purpose of Figure 5.3 is to check whether the uniform distribution appears to be a better fit than the Gamma distribution. The best fitting Gamma probability density

FIGURE 5.2: Set of fitted distributions

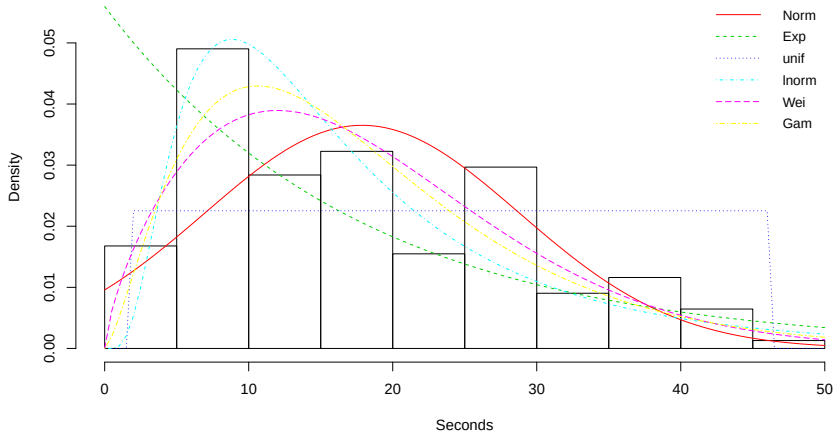


TABLE 5.3: AIC values of fitted distributions

Distribution	AIC value
Normal	1185.1
Exponential	1205.8
Uniform	N/A
Log-normal	1159.9
Weibull	1151.8
Gamma	**1151.6**

function, a P-P plot, and a comparison of the empirical and theoretical CDFs are plotted in Figure 5.4.

Therefore, the time taken to receive prescriptions is best modelled by a random variable X such that, $X \sim Gamma(\alpha, \beta)$, $\alpha = 2.4622352$ and $\beta = 0.1377484$.

FIGURE 5.3: Multi distribution Q-Q plots

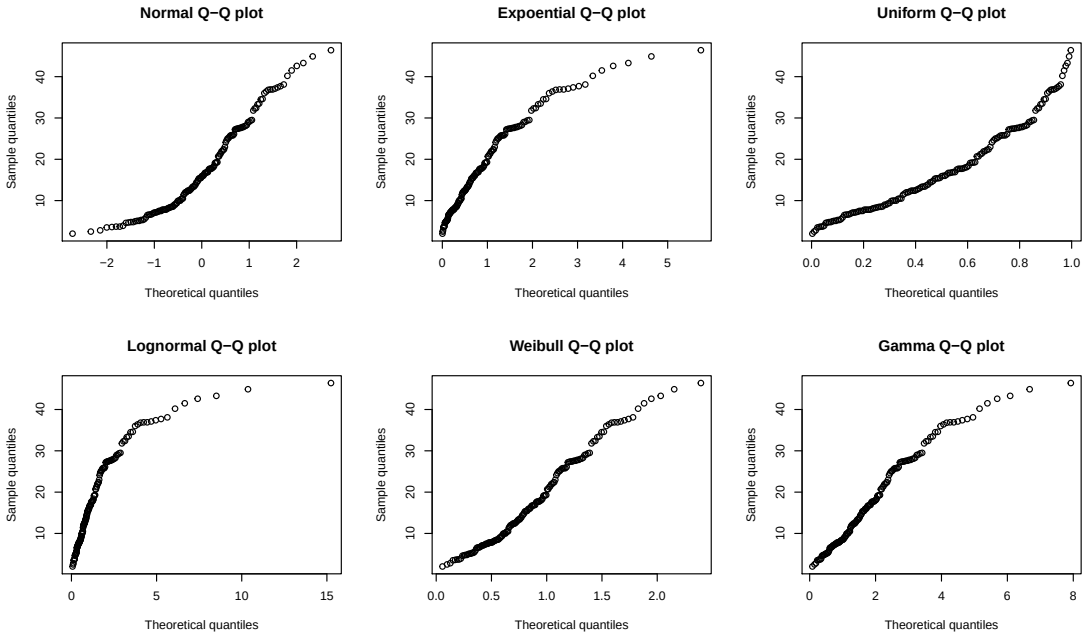
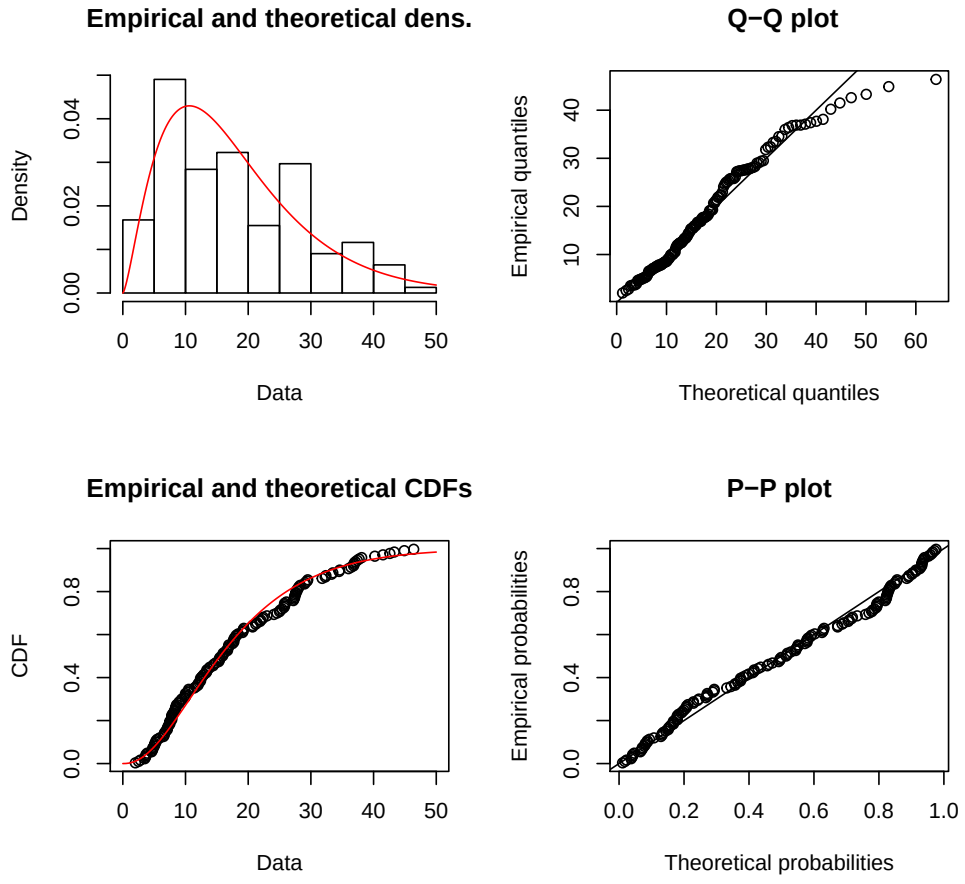
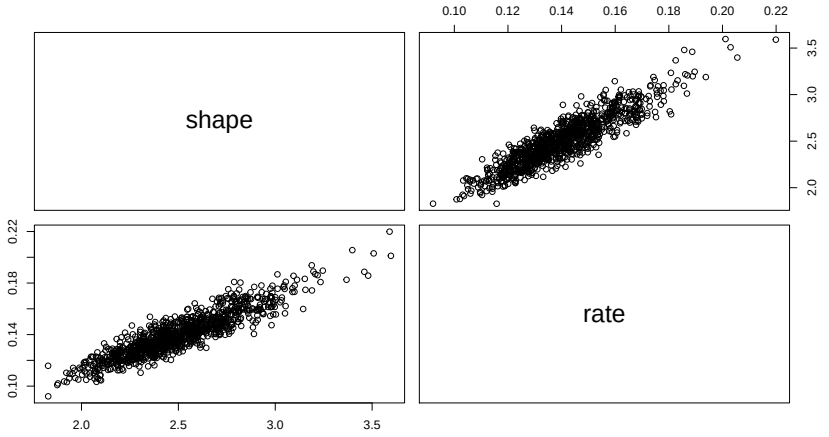


FIGURE 5.4: Distribution fitting analysis



1000 samples of the Gamma distribution's parameters were obtained by producing bootstrap samples from the observed data set. These provide 95% confidence interval estimations for the values of α and β. Figure 5.5 shows the results of the sampling.

FIGURE 5.5: Bootstrapped values of parameters



The 95% confidence intervals for the estimated parameters are:
[2.0188330, 3.1007944] for α and [0.1116018, 0.1799003] for β.

5.3.7 Example 2: Label Generation of 1 item prescriptions in Pharmacy A

This example demonstrates how the distribution fitting analysis was performed for stages where the data was segmented by the number of items in a prescription, whereas in Example 1 the stage of prescription reception was assumed to be independent of the number of items. There were 69 recorded observations of label generation of 1 item prescriptions in Pharmacy A. Outliers were removed by removing all data points further than 1.5IQR away from the mean, this removed 6 data points in total.

FIGURE 5.6: Histogram of prescription reception

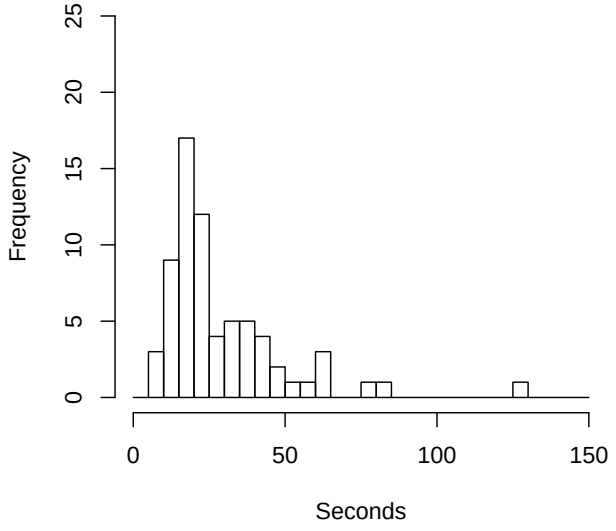


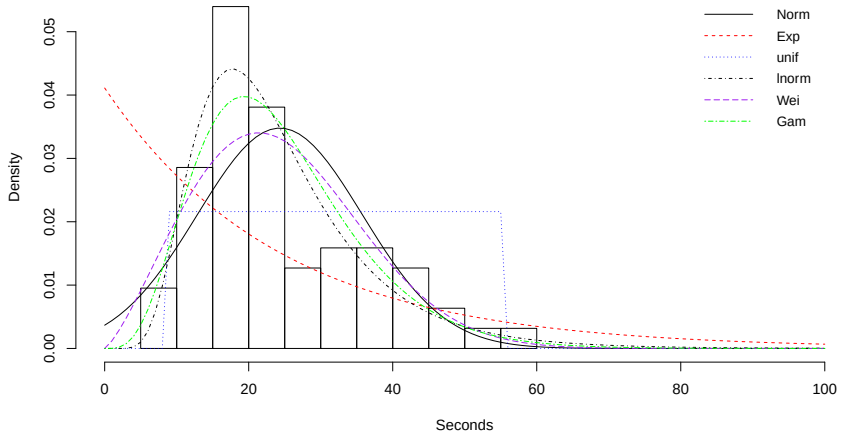
Figure 5.6 shows a histogram of the 69 observations of 1 item label generation recorded in Pharmacy A, and Table 5.4 shows a summary of the data once outliers had been removed.

TABLE 5.4: Summary of 1 item label generation

Pharmacy	Min	Q1	Median	Mean	Q3	Max
A	8.9	16.5	20.7	24.3	31.9	55.2

Distribution fitting analysis of the time taken to generate labels for 1 item prescriptions was conducted on the data set. Figure 5.7 shows the theoretical densities of the 6 distributions fitted to the data. The six AIC values associated with each distribution are shown in Table 5.5. As in 5.3.6, an AIC value cannot be calculated for the Uniform distribution, since the likelihood function is not twice differentiable.

FIGURE 5.7: Set of fitted distributions for 1 item label generation



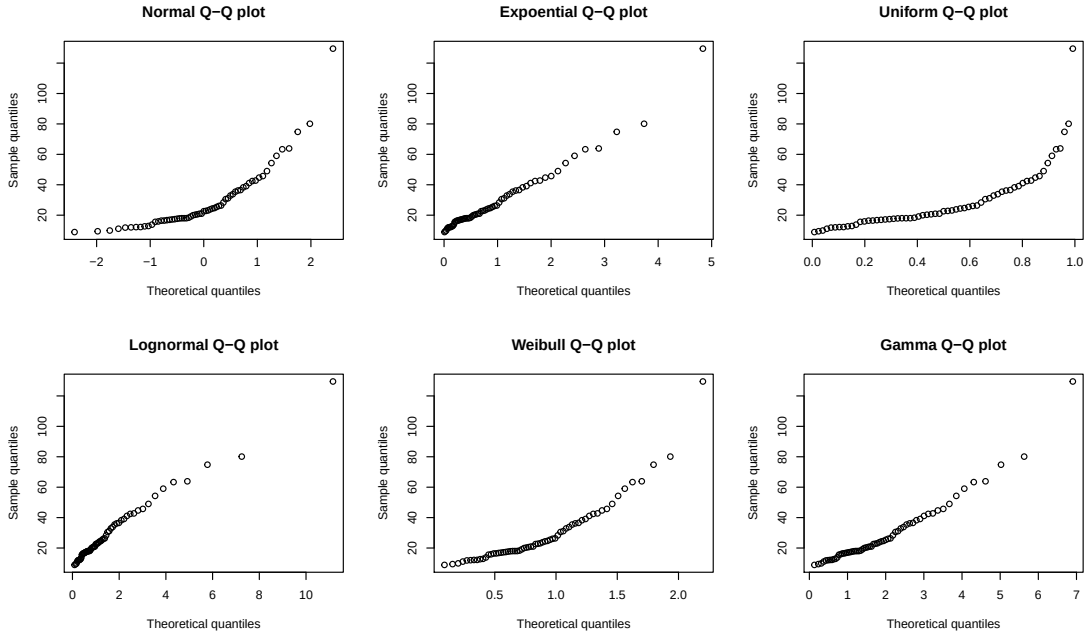
The fitted lognormal distribution produced the lowest AIC value of the 5 distributions tested. To test the fit of each of the distributions, QQ plots for the data plotted against each distribution are shown in Figure 5.8. The main purpose of Figure 5.8 is to check whether the uniform distribution appears to be a better fit than the lognormal distribution. The best fitting lognormal probability density function, a P-P plot, and a comparison of the empirical and theoretical CDFs are plotted in Figure 5.9.

TABLE 5.5: AIC values of fitted distributions

Distribution	AIC value
Normal	490.2
Exponential	530.1
Uniform	N/A
Log-normal	**473.5**
Weibull	481.7
Gamma	475.9

Therefore, the time taken to generate labels for 1 item prescriptions is best modelled by a random variable X such that, $X \sim lognormal(\mu, \sigma)$, $\mu = 3.0857021$ and $\sigma = 0.4590996$.

FIGURE 5.8: Multi distribution Q-Q plots for 1 item label generation: Pharmacy A



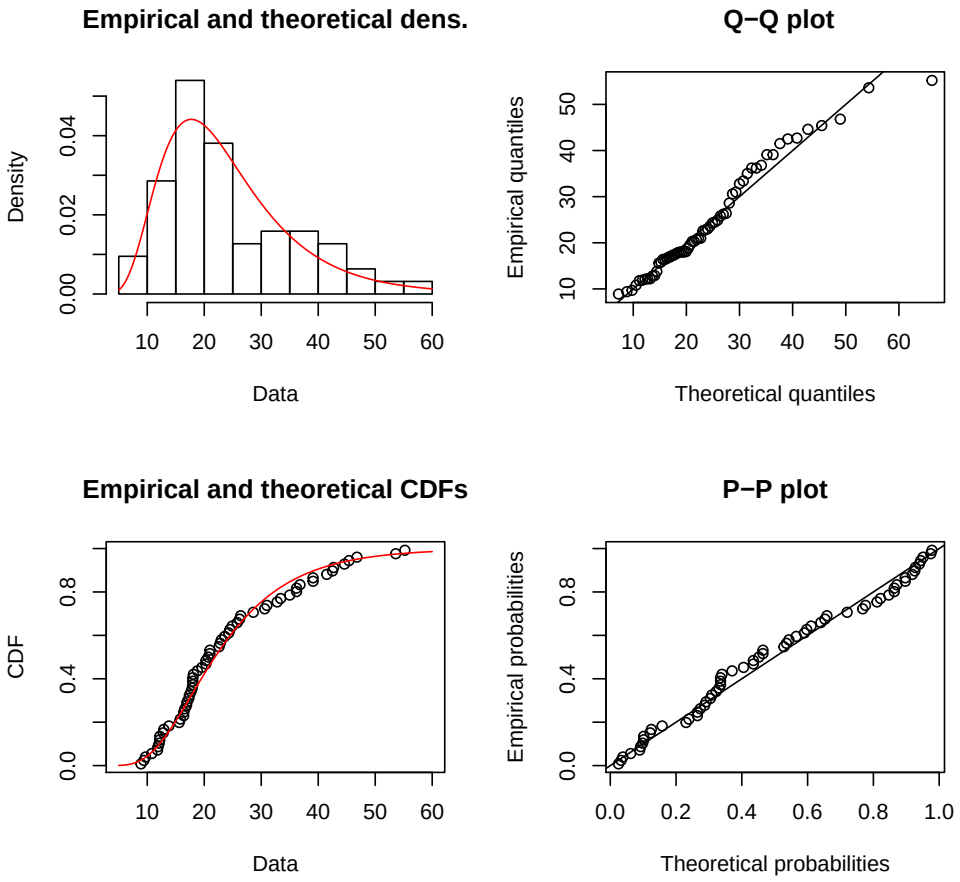
1000 samples of the lognormal distribution's parameters were obtained by producing bootstrap samples from the observed data set. These provide 95% confidence interval estimations for the values of μ and σ. Figure 5.10 shows the results of the sampling.

The 95% confidence intervals for the estimated parameters are:
[2.9650126, 3.1997809] for μ and [0.3761913, 0.5280642] for σ.

5.3.8 Example 3: Label Generation of large prescriptions in Pharmacy A

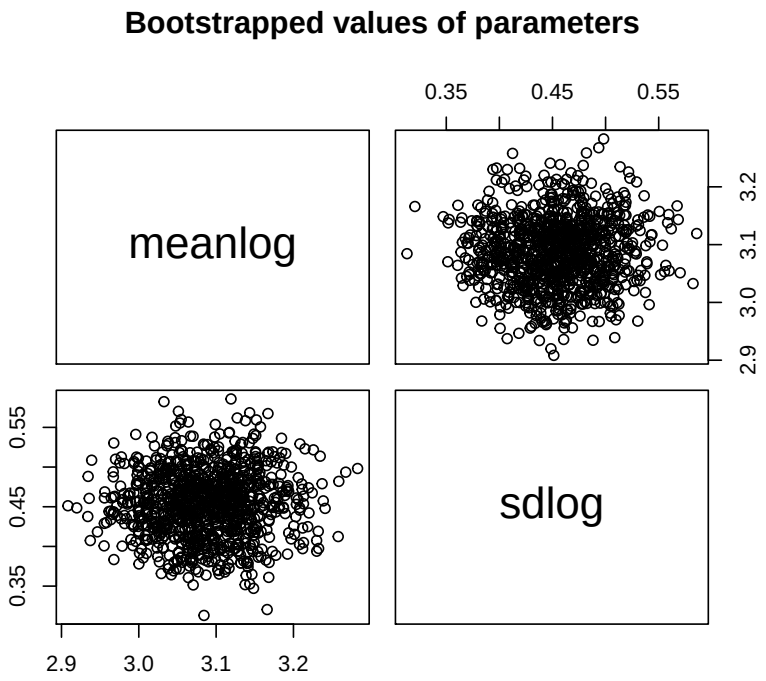
This example demonstrates how distributions were generated in cases where a small number of observations were recorded. Only 4 observations of larger prescriptions (those containing more than 8 items), were observed in Pharmacy A. This is considered too small a dataset to conduct a distribution fitting analysis. The threshold number of observations considered sufficient to conduct an analysis was 15. This value of 15 was assumed to be sufficient, based on the amount of data that was collected for a large number of segments, and the applicability of the law of large numbers.

FIGURE 5.9: Distribution fitting analysis for 1 item label generation: Pharmacy A



In this case, where few observations were collected, the distribution type is assumed to be the same as the for the segment below in size. In this case, the distribution type for large prescriptions is assumed to be the same as that for medium sized prescriptions, the lognormal distribution. Best fitting parameters for the data can be estimated using the method of maximum likelihood.

FIGURE 5.10: Bootstrapped values of parameters



5.3.9 Results

The results of the data analysis performed following the example case outlined in Section 5.3.5 are presented here. The analysis for each pharmacy is presented seperately, and then the analysis of the data collected in pharmacies A and B (stores from the same large multiple) is also presented. Note, the distribution for any segments of the data which received less than 15 observations were assumed to be the same as the segment below. New parameters were estimated for the assumed distribution using any data that was collected. Each task is assigned a best fitting distribution for each size of prescription. Note that timings have been grouped together by prescription size, and the frequency of each size prescription varied in the pharmacies.

5.3.9.1 Pharmacy A

Table 5.6 shows a summary of the data analysis performed on a combined data set from Pharmacy A.

TABLE 5.6: Summary of quantitative data analysis for Pharmacy A

Task	No# items	N	Mean	S.D.	Distribution	Parameter 1	Parameter 2
Prescription reception	General	160	17.9	20.9	$X \sim Gamma(\alpha, \beta)$	2.46	0.138
Label generation	1	69	29.1	20.4	$X \sim lognormal(\mu, \sigma)$	3.19	0.57
	2	32	35.1	16.4	$X \sim lognormal(\mu, \sigma)$	3.46	0.45
	3	18	47.7	26.5	$X \sim lognormal(\mu, \sigma)$	3.75	0.47
	4	20	46.2	20.2	$X \sim Gamma(\alpha, \beta)$	5.57	0.12
	Medium	17	78.7	40.4	$X \sim lognormal(\mu, \sigma)$	4.25	0.49
	Large	4	155.9	88.3	$X \sim lognormal(\mu, \sigma)$	4.87	0.67
Picking	1	65	18.7	20.0	$X \sim lognormal(\mu, \sigma)$	2.48	0.92
	2	37	15.0	11.7	$X \sim lognormal(\mu, \sigma)$	2.51	0.59
	3	21	27.2	21.4	$X \sim lognormal(\mu, \sigma)$	3.09	0.65
	4	13	24.2	10.9	$X \sim lognormal(\mu, \sigma)$	3.09	0.44
	Medium	20	43.9	29.4	$X \sim lognormal(\mu, \sigma)$	3.57	0.67
	Large	4	69.9	23.2	$X \sim lognormal(\mu, \sigma)$	4.18	0.32
Applying labels	1	44	13.8	11.8	$X \sim lognormal(\mu, \sigma)$	2.37	0.70
	2	35	34.9	30.8	$X \sim lognormal(\mu, \sigma)$	3.26	0.73
	3	16	27.3	11.0	$X \sim lognormal(\mu, \sigma)$	3.23	0.39
	4	17	49.9	39.6	$X \sim lognormal(\mu, \sigma)$	3.68	0.65
	Medium	35	72.9	44.8	$X \sim lognormal(\mu, \sigma)$	4.12	0.57
	Large	13	169.5	122.8	$X \sim lognormal(\mu, \sigma)$	4.88	0.77
Accuracy checking	1	54	31.7	19.6	$X \sim lognormal(\mu, \sigma)$	3.33	0.48
	2	36	34.2	15.7	$X \sim lognormal(\mu, \sigma)$	3.44	0.41
	3	18	48.3	21.9	$X \sim lognormal(\mu, \sigma)$	3.79	0.41
	4	16	41.4	12.3	$X \sim Gamma(\alpha, \beta)$	11.7	0.28
	Medium	29	76.9	31.2	$X \sim lognormal(\mu, \sigma)$	4.27	0.36
	Large	7	142.1	89.4	$X \sim lognormal(\mu, \sigma)$	4.77	0.63
Handover/store	General	160	34.2	33.1	$X \sim lognormal(\mu, \sigma)$	3.15	0.87

5.3.9.2 Pharmacy B

Table 5.7 shows a summary of the data analysis performed on a combined data set from Pharmacy B. Note that no data was collected for the prescription reception or the handover/store stages of the process from pharmacy B. This was due to the layout of the pharmacy, the reception area was not visible from the main data collection area.

Table 5.7: Summary of quantitative data analysis Pharmacy B

Task	No# items	N	Mean	S.D.	Distribution	Parameter 1	Parameter 2
Prescription reception	General	0	N/a	N/a	N/a	N/a	N/a
Label generation	1	81	25.5	15.5	$X \sim lognormal(\mu, \sigma)$	3.09	0.53
	2	25	45.1	25.0	$X \sim lognormal(\mu, \sigma)$	3.69	0.48
	3	15	73.5	49.0	$X \sim lognormal(\mu, \sigma)$	4.13	0.55
	4	16	87.2	51.7	$X \sim lognormal(\mu, \sigma)$	4.32	0.54
	Medium	17	115.0	58.2	$X \sim lognormal(\mu, \sigma)$	4.65	0.40
	Large	6	192.2	64.1	$X \sim lognormal(\mu, \sigma)$	5.19	0.39
Picking	1	73	22.4	35.9	$X \sim lognormal(\mu, \sigma)$	2.69	0.77
	2	28	26.7	13.7	$X \sim Gamma(\alpha, \beta)$	3.93	0.15
	3	17	31.6	14.4	$X \sim Gamma(\alpha, \beta)$	5.25	0.17
	4	15	45.1	20.7	$X \sim lognormal(\mu, \sigma)$	3.71	0.45
	Medium	23	72.7	40.6	$X \sim lognormal(\mu, \sigma)$	4.16	0.49
	Large	4	151.3	69.8	$X \sim lognormal(\mu, \sigma)$	4.93	0.42
Applying labels	1	69	22.7	20.8	$X \sim lognormal(\mu, \sigma)$	2.79	0.79
	2	33	29.7	16.6	$X \sim Weibull(\lambda, k)$	1.93	33.6
	3	14	56.1	41.3	$X \sim Weibull(\lambda, k)$	1.45	62.2
	4	13	52.0	23.5	$X \sim Weibull(\lambda, k)$	2.52	58.7
	Medium	26	116.5	73.2	$X \sim lognormal(\mu, \sigma)$	4.59	0.58
	Large	5	195.5	80.2	$X \sim lognormal(\mu, \sigma)$	5.21	0.36
Accuracy checking	1	79	39.7	42.2	$X \sim lognormal(\mu, \sigma)$	3.44	0.62
	2	32	57.6	38.6	$X \sim lognormal(\mu, \sigma)$	3.87	0.58
	3	16	71.0	36.2	$X \sim lognormal(\mu, \sigma)$	4.14	0.49
	4	7	62.8	16.1	$X \sim lognormal(\mu, \sigma)$	4.11	0.25
	Medium	21	105.1	68.0	$X \sim lognormal(\mu, \sigma)$	4.50	0.53
	Large	5	223.1	158.9	$X \sim lognormal(\mu, \sigma)$	5.25	0.50
Handover/store	General	0	N/a	N/a	N/a	N/a	N/a

5.3.9.3 Pharmacy C

Table 5.8 shows a summary of the data analysis performed on a combined data set from Pharmacy C.

Table 5.8: Summary of quantitative data analysis Pharmacy C

Task	No# items	N	Mean	S.D.	Distribution	Parameter 1	Parameter 2
Prescription reception	General	160	28.0	25.2	$X \sim lognormal(\mu, \sigma)$	3.02	0.79
Label generation	1	101	22.8	21.3	$X \sim lognormal(\mu, \sigma)$	2.91	0.58
	2	29	27.6	7.06	$X \sim Gamma(\alpha, \beta)$	15.66	0.57
	3	22	46.7	24.4	$X \sim lognormal(\mu, \sigma)$	3.74	0.42
	4	5	50.8	14.3	$X \sim lognormal(\mu, \sigma)$	3.90	0.25
	Medium	3	47.0	8.7	$X \sim lognormal(\mu, \sigma)$	3.84	0.10
	Large	0	N/a	N/a	N/a	N/a	N/a
Picking	1	106	18.8	21.8	$X \sim lognormal(\mu, \sigma)$	2.55	0.81
	2	25	35.2	20.5	$X \sim lognormal(\mu, \sigma)$	3.39	0.59
	3	12	44.3	27.1	$X \sim lognormal(\mu, \sigma)$	3.58	0.71
	4	12	49.8	49.1	$X \sim lognormal(\mu, \sigma)$	3.61	0.74
	Medium	5	101.4	66.4	$X \sim lognormal(\mu, \sigma)$	4.40	0.71
	Large	0	N/a	N/a	N/a	N/a	N/a
Applying labels	1	70	19.6	17.9	$X \sim lognormal(\mu, \sigma)$	2.60	0.89
	2	38	44.1	32.4	$X \sim lognormal(\mu, \sigma)$	3.53	0.73
	3	25	55.5	48.3	$X \sim lognormal(\mu, \sigma)$	3.84	0.52
	4	13	57.2	27.1	$X \sim lognormal(\mu, \sigma)$	3.91	0.61
	Medium	14	159.4	145.1	$X \sim lognormal(\mu, \sigma)$	4.83	0.63
	Large	0	N/a	N/a	N/a	N/a	N/a
Accuracy checking	1	90	20.3	16.6	$X \sim lognormal(\mu, \sigma)$	2.73	0.77
	2	35	30.6	19.2	$X \sim Gamma(\alpha, \beta)$	2.95	0.10
	3	10	49.6	23.0	$X \sim Gamma(\alpha, \beta)$	5.18	0.10
	4	14	60.3	23.3	$X \sim Gamma(\alpha, \beta)$	6.67	0.11
	Medium	10	88.8	40.1	$X \sim Gamma(\alpha, \beta)$	5.12	0.06
	Large	1	67.7	N/a	$X \sim Gamma(\alpha, \beta)$	5.15	0.06
Handover/store	General	160	41.9	51.8	$X \sim lognormal(\mu, \sigma)$	3.25	0.98

5.3.9.4 Pharmacy D

Table 5.9 shows a summary of the data analysis performed on a combined data set from Pharmacy D.

Table 5.9: Summary of quantitative data analysis Pharmacy D

Task	No# items	N	Mean	S.D.	Distribution	Parameter 1	Parameter 2
Prescription reception	General	160	20.9	20.9	$X \sim lognormal(\mu, \sigma)$	2.73	0.78
Label generation	1	67	32.6	34.8	$X \sim lognormal(\mu, \sigma)$	3.26	0.59
	2	38	41.4	14.8	$X \sim lognormal(\mu, \sigma)$	3.67	0.33
	3	22	56.5	29.2	$X \sim lognormal(\mu, \sigma)$	3.93	0.45
	4	18	55.9	25.7	$X \sim lognormal(\mu, \sigma)$	3.94	0.39
	Medium	13	72.6	21.5	$X \sim lognormal(\mu, \sigma)$	4.25	0.26
	Large	2	244.7	81.2	$X \sim lognormal(\mu, \sigma)$	5.47	0.24
Picking	1	90	20.2	18.2	$X \sim lognormal(\mu, \sigma)$	2.74	0.71
	2	27	42.2	29.7	$X \sim lognormal(\mu, \sigma)$	3.53	0.64
	3	21	47.0	16.5	$X \sim lognormal(\mu, \sigma)$	3.80	0.32
	4	8	50.4	23.7	$X \sim lognormal(\mu, \sigma)$	3.82	0.44
	Medium	10	81.3	14.9	$X \sim lognormal(\mu, \sigma)$	4.38	0.19
	Large	4	114.3	19.2	$X \sim lognormal(\mu, \sigma)$	4.70	0.16
Applying labels	1	91	27.8	24.1	$X \sim lognormal(\mu, \sigma)$	3.03	0.68
	2	28	41.5	17.2	$X \sim Gamma(\alpha, \beta)$	6.09	0.15
	3	19	55.8	22.2	$X \sim Gamma(\alpha, \beta)$	6.68	0.12
	4	7	89.3	33.8	$X \sim Gamma(\alpha, \beta)$	9.67	0.11
	Medium	13	94.8	27.2	$X \sim Gamma(\alpha, \beta)$	12.5	0.13
	Large	2	140.9	60.1	$X \sim Gamma(\alpha, \beta)$	10.6	0.07
Accuracy checking	1	68	24.8	11.4	$X \sim Gamma(\alpha, \beta)$	4.56	0.18
	2	38	39.0	21.9	$X \sim lognormal(\mu, \sigma)$	3.54	0.49
	3	17	52.4	17.1	$X \sim lognormal(\mu, \sigma)$	3.91	0.30
	4	11	72.3	31.6	$X \sim lognormal(\mu, \sigma)$	4.20	0.41
	Medium	21	104.4	42.7	$X \sim Gamma(\alpha, \beta)$	6.70	0.06
	Large	5	196.5	74.4	$X \sim Gamma(\alpha, \beta)$	8.73	0.04
Handover/store	General	160	49.0	43.6	$X \sim lognormal(\mu, \sigma)$	3.55	0.85

5.3.9.5 Pharmacy A & B

Table 5.10 shows a summary of the data analysis performed on a combined data set from pharmacies A and B.

5.3.10 Discussion

Pharmacies A and B were both members of the same large multiple chain. Comparing the mean duration for each segment of the data across both pharmacies shows that, in all but one of the segments, the average duration to complete a stage was longer in Pharmacy B. The one segment where Pharmacy B was faster was applying labels to prescriptions containing two items. Pharmacy A may have been quicker since it was a co-located pharmacy, which was more geared towards dealing with walk-in patients. Whereas Pharmacy B had a greater focus on dispensing nursing home trays, which meant there was less focus on dealing with walk-ins.

Pharmacies C and D were unrelated independent pharmacies. Pharmacy C was generally the faster of the two, and for all but 4 segments of prescription size,

Table 5.10: Summary of distribution fitting analysis

Task	No# items	N	Mean	Distribution	Paramater 1	Parameter 2
Prescription reception	N/a	160	17.9	$X \sim Gamma(\alpha, \beta)$	2.46	7.25
Label generation	1	150	27.2	$X \sim lognormal(\mu, \sigma)$	3.14	0.549
	2	57	38.0	$X \sim lognormal(\mu, \sigma)$	3.53	0.454
	3	33	54.6	$X \sim lognormal(\mu, \sigma)$	3.88	0.487
	4	36	59.5	$X \sim lognormal(\mu, \sigma)$	3.96	0.515
	Medium	34	90.8	$X \sim Gamma(\alpha, \beta)$	5.41	16.8
	Large	10	177.7	$X \sim Gamma(\alpha, \beta)$	4.48	39.7
Picking	1	138	18.7	$X \sim lognormal(\mu, \sigma)$	2.58	0.851
	2	65	20.0	$X \sim lognormal(\mu, \sigma)$	2.79	0.653
	3	38	27.0	$X \sim Gamma(\alpha, \beta)$	3.99	6.76
	4	27	45.7	$X \sim lognormal(\mu, \sigma)$	3.42	0.540
	Medium	43	56.0	$X \sim Gamma(\alpha, \beta)$	3.06	18.28
	Large	5	131.2	$X \sim Gamma(\alpha, \beta)$	4.00	32.8
Applying labels	1	113	19.2	$X \sim lognormal(\mu, \sigma)$	2.63	0.787
	2	68	32.3	$X \sim lognormal(\mu, \sigma)$	3.24	0.673
	3	30	40.8	$X \sim lognormal(\mu, \sigma)$	3.47	0.658
	4	30	50.8	$X \sim lognormal(\mu, \sigma)$	3.75	0.599
	Medium	61	91.5	$X \sim lognormal(\mu, \sigma)$	4.32	0.615
	Large	18	176.7	$X \sim Gamma(\alpha, \beta)$	2.59	68.5
Accuracy checking	1	133	32.2	$X \sim lognormal(\mu, \sigma)$	3.35	0.486
	2	68	45.2	$X \sim lognormal(\mu, \sigma)$	3.65	0.539
	3	34	59.0	$X \sim lognormal(\mu, \sigma)$	3.96	0.484
	4	34	47.9	$X \sim Gamma(\alpha, \beta)$	8.66	0.181
	Medium	50	83.4	$X \sim lognormal(\mu, \sigma)$	4.34	0.409
	Large	12	175.8	$X \sim lognormal(\mu, \sigma)$	4.97	0.629
Handover/store	N/a	160	34.2	$X \sim lognormal(\mu, \sigma)$	3.15	0.874

Pharmacy C had lower average task durations. The 4 segments in which Pharmacy D was faster were: prescription reception, picking for 4 item prescriptions, applying labels to 2 item prescriptions and medium sized prescriptions. Pharmacy C may have been faster due to the pharmacies general attitude towards waiting times. Many of the staff expressed that they took pride in dispensing prescriptions quickly to deliver the best care for patients.

Comparing the average durations recorded in pharmacies A and C, pharmacy A was faster for 15 of the segments, mainly from the stages of picking, and applying. Pharmacy C was faster in 8 segments which were mostly in the stages of labelling and accuracy checking.

Table 5.11 shows the average duration for each of the tasks in pharmacies which were either part of the large chains, or independent pharmacies.

The independent pharmacies had shorter average durations for 13 of the 26 total segments of the data. The independent pharmacies were faster for: 5 of the 6 label generation segments, picking single item prescriptions, applying labels to

large prescriptions, and accuracy checking all sizes of prescription, except medium sized prescriptions.

5.3.11 Summary

This section has shown the method of quantitative data collection used to collect data from 4 UK community pharmacies on the time taken to dispense prescriptions at each stage of the process. The distribution fitting method used to analyse the data was shown, and three examples of applying the method to subsections of the data were given to illustrate how the analysis was performed. The pharmacies were each analysed separately, and all results are presented in table form in Section 5.3.9. The data from Pharmacies A and B are combined to produce a larger data set, which was also analysed.

Table 5.11: Comparing independents and large multiples

Task	No# items	Avg large multiples (s)	Avg independents (s)
Prescription Reception	N/a	17.9	24.5
Label generation	1	27.2	26.7
	2	39.5	35.4
	3	59.4	51.6
	4	64.4	54.8
	Med	96.9	67.8
	Large	177.7	244.7
Picking	1	20.7	19.4
	2	20.0	38.8
	3	29.2	46.0
	4	35.4	50.0
	Med	59.3	88.0
	Large	110.6	114.3
Applying labels	1	19.2	24.2
	2	32.4	42.9
	3	40.7	55.6
	4	50.8	68.4
	Med	91.5	147.0
	Large	176.7	140.9
Accuracy checking	1	36.5	22.2
	2	45.2	35.0
	3	59.0	51.4
	4	47.9	65.6
	Med	88.7	99.4
	Large	175.9	175.0
Handover/store	N/a	34.2	45.5

5.4 Interview Analysis

Qualitative semi-structured interviews were carried out in each of the pharmacies with the aim of understanding practitioners' views of the dispensing process, and making sure the modelling framework reflects these views. A further aim of the interviews is to give the models' predictions credibility among pharmacy professionals. From a pragmatic perspective, the model will only be useful if it is supported and understood by key stakeholders in the field of community pharmacy, which will only be possible if it is reflective of their views and understanding of the dispensing process. Towards this goal, findings were used to influence the model with a goal of making it more representative of the practice of dispensing from the perspective of those working in the field. The Coloured Petri Net (Version 4), developed in Chapter 6 has some features, such as timing distributions derived from real data and intermediate accuracy checks, supported by the results of this study.

5.4.1 Method

Qualitative semi-structured interviews were carried out with all consenting members of staff at each of the sites. An interview guide with three key themes was used as a starting point for understanding the key features and procedures in everyday practice, and open ended questions were also included in the interviews to allow the participants to express their beliefs and feelings. The three key themes on the initial interview sheet were: observation clarification, errors and near misses, and task management/prioritisation.

Follow-up questions were used as a way to understand and explore topics that were either observed during the observation period, or brought up by the interviewees during interviews. A common follow-up question was to clarify the meaning of specific terms, acronyms, or explanations used by the participants when describing their work environment. If a misunderstanding did arise, examples of this type of follow up question included: "What is the red zone?", "Hang on, I don't quite understand.", or "What are those?". Each interview lasted between 10 minutes to half an hour. All interviews were audio recorded.

The qualitative interviews were transcribed and thematically analysed. The qualitative analysis software Nvivo 12 was used to aid the analysis, and code each of

the interviews into an initial set of nodes (Bryman, 2012). Nodes are representative of topics and themes covered during the interviews, where each section of the interview is placed into a node. The initial set of nodes was constructed during the transcription stage of the analysis, by noting down any recurrent patterns or topics that came up. Throughout the coding stage of the analysis, additional themes were constructed for any sections of the interviews that did not fall into one of the initial nodes.

5.4.2 Results

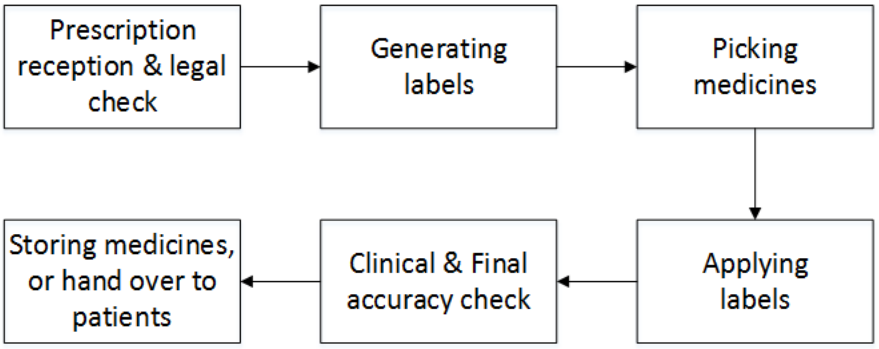
Interviews were conducted using a semi-structured interview guide, with three main topics of discussion: observation clarification, errors, and task prioritisation. Initial themes were noted down during the transcription stage of the analysis. Throughout the coding of the interviews, emergent themes were constructed to include all the data collected. After finishing the analysis, 6 principal themes were identified, and these are shown below:

1. Process Validation
   Asking practitioners about how they view the process, does our understanding match up with theirs?
2. Errors/Frequency of errors
   How do practitioners think about errors, and the frequency of errors occurring in community pharmacy?
3. Inefficiencies
   What processes are considered to be inefficient in the day to day running of pharmacies?
4. Stress factors
   Exploring which factors or conditions in pharmacies which may be more likely to produce errors?
5. Safety procedures
   Exploring the systems and patterns of behaviour used to respond to and prevent errors.
6. Waiting times
   Practitioners views on acceptable waiting times for patients in community pharmacy.

5.4.2.1 Process Validation

One key factor to assess in terms of process validation was whether our understanding of the dispensing process derived from the literature, was in accordance with practitioners' views of the dispensing process. To try and ascertain this, the researcher would, either describe or show, interviewees a flow chart similar to that of Figure 5.11 (a six stage abstraction of the dispensing process), and ask them about their thoughts of the abstraction.

FIGURE 5.11: The main stages of dispensing



Interviewees had a generally positive response to the abstraction of the dispensing process, although some respondents also made clear that the process was not rigid, and gave examples of some exceptions that can arise. One exception that came up frequently was that printing labels, and picking medicines can be completed in a flexible manner. The following quotes highlight how sometimes the medicines can be picked into baskets, before the labels are printed, and how practitioners were familiar with the abstraction of the dispensing process:

> "Yep. They can be. There's no hard and fast rule for that really. ...Say if we've got quite a few walk-ins and somebody needs to help, then they'll probably pick before the labels are being printed." (Pharmacy A – ACT)

> "So, is the reception [Flowchart Figure 5.11] where you get the prescription? ..And then label, how long it takes to make up the labels? Picking medication, taking it out the drawers... ...applying labels, obviously just sticking the labels on, and making sure it's the right medication, and accuracy check, just checking what you've done,

putting it in the basket neatly and then handing over." (Pharmacy D – Dispensing Technician)

5.4.2.2 Errors/Frequency of errors

Near misses were seen as a generally common occurrence in pharmacies by practitioners, and 2 methods of near miss recording were encountered. The following quote demonstrates that near misses are not a rare event, and something that happens on a daily basis. The following comment was made by a Locum pharmacist. Note that a Locum is a person brought in to temporarily fill the role, who might work 1 or 2 days a week, as opposed to a full time pharmacist.

> "On a daily basis. Yeah. Near misses happen every day to be honest. We used to record on a hardcopy, which recently our company stopped, and they asked us to go online... And being honest sometimes we don't log it, we should log it, but we don't get chance. When we used to have hardcopy, it's quick. Within one minute I can fill it out. Now, if, let's say I'm doing three or four near misses... I'm there for an hour." (Pharmacy B – Locum pharmacist)

Interviewees were asked their views on the primary dispensing task error rates used in the CPN model developed in Chapter 6 (0.03, 0.05, and 0.06). Reponses were mixed, although the general response was that this near miss rate was too high. The quotes below demonstrate the difference in views.

> "1 in 500 I'd say in terms of, it's tricky to put a number on it but 1 in 20 is too high." (Pharmacy C – Dispensary manger)

> "I think it is around 1-2 in 100. It is because most of the items are very common items, you will not make an error at all because you dispense about 5 of the items each day." (Pharmacy D – Pharmacist owner)

An alternative view was encountered in Pharmacy B. The pharmacist was very involved in the error reporting process, and the pharmacy dispensed a high volume

of trays for nursing homes. It seemed like the trays for nursing homes had a higher probability of including near misses. As well as a high percentage of their dispensing being trays, there was also some newer or relief members of staff working in the store, that could have been contributing to this alternative view of the prevalence of near misses. The quote below shows how some pharmacies can have very different rates of near misses occurring.

> "Yeah. So, when I started here in January last year, there was not really much near miss reporting, or it wasn't consistent. So when we started looking at it, we were hitting, some weeks up to 50, and I don't think that covers everything, so 50 from just all the general dispensing, and then we do lots of trays as well, so I think generally in this pharmacy which is doing roughly around, just over 2000 items a week, we hit about 100 near misses, of different types?" (Pharmacy B – Regular Pharmacist)

A pharmacist in an independent pharmacy brought up the idea that their experienced dispensing staff were very reliable, and unlikely to make many near misses, but the same confidence was not expressed when talking about trainee pre-registration pharmacists.

> "Every hundred items they might get something wrong, like a strength or a form, subtle nuances, but they're very good in there. Dispensers particularly, pre-regs, maybe not so." (Pharmacy C – Pharmacist owner)

5.4.2.3 Inefficiencies

Inefficiencies within the dispensary could be causing unnecessary delays, leading to longer waiting times, and an increased workload for pharmacy staff. While observing the process it was considered important to try and identify any areas of the process that seemed inefficient, and get practitioners views on these phenomena. Any processes that seemed excessively lengthy or slow, were investigated and explored during interviews. 2 common inefficiencies seen to some degree across all the 4 sites were; dispensing split packs, and having to unmake prescriptions which patients who had ordered via EPS never came to collect.

Split pack dispensing occurs when the amount of medicine prescribed to a patient does not conform to the manufacturer's default dosage. An example could be, a patient is prescribed 50 paracetamol tablets, but the packs they come in only contain 32 tablets. In such a case, the dispenser dispensing such a prescription has to construct a brand new white box, place 32 tablets from one pack of paracetamol inside, and then use scissors to cut out the extra 18 tablets from a second pack. This significantly increases the time taken to dispense, adding to waiting times, and the workload of the pharmacy. Views were mixed about split packs, and some of the more experienced staff viewed them as less of a problem, and were almost accepting that split packs would always be a part of the dispensing process. The following quotes show practitioners expressing their frustrations with the extra difficulties of dispensing split pack medications.

> "It's easier when it says 28 tablets and you only get 28 tablets in a box so you don't have to split, that's a lot quicker, rather than having to split the boxes, or even getting new cartons out." (Pharmacy D – Dispenser)

> "They're a pain. Split packs are a massive pain, at my old place I'd say you'd have about 15-20 a day... cetirizine packs... They come in packs of 30, so when they dispense 28 it's a pain. ...give them the extra 2 and it would be so much quicker. ...if it's a split pack I'd say it adds on about a minute or 2 extra on top of the dispensing process." (Pharmacy C – Pre Reg Pharmacist)

Of those who were more reconciled with the necessity of split packs, there were mixed motivations. One key reason for the acceptance was due to the requirements of addiction patients, who are required to be on shorter term supplies than a standard 28 days regime. The following quotes show practitioners more comfortable with the need for split pack dispensing. The case is made for addiction patients to be given split packs, and the case is a strong one.

> "That is in the ideal perfect world, but it ain't gonna happen. Ever. Never going to get it. Yeah, it makes life easier. It's quicker. Quicker dispensing, yeah, original packs, much quicker. But we don't live in an ideal world. So, if you go in any pharmacy, walk in the dispensary, you

> will see split packs. Multiple, independent, you will see it." (Pharmacy C – Pharmacist owner)

Another inefficiency common to all the sites was the phenomenon of patients not coming in to collect their EPS prescriptions. EPS prescriptions can be ordered from outside the pharmacy, after which the pharmacy will get the prescription ready and bagged up for when patients come in. However, there is only so long a pharmacy can keep dispensed prescriptions waiting on the shelves for patients to come and collect them. Sites would allow a 1-2 month collection period, where after the allotted time, if a patient had not come in to collect their prescription, it would be taken apart and the medicine would be put back onto the shelves. This is a clear inefficiency in the dispensing process, and the quote below outlines this process.

> "Every Friday, I'll get someone to cleanse the retrieval system... they'll go through first and find out any from the previous week that have had the L put on them, which means we've sent out a letter the week before... because their date was 4 weeks ago, from Fridays date. So, they send the letter out to them, they then stay on the shelf for a further week, so... they've then got seven days to come and collect it. The next Friday, they all then get taken off, and they get broken back down again." (Pharmacy A – ACT)

5.4.2.4 Stress factors

We wanted to hear from practitioners which working conditions they found to be the most likely to produce errors. These environmental factors that can lead to poor performance were classified as stress factors. A wide range of stress factors were identified throughout the interviews, including: the pharmacy being busy, being short of staff, working with inexperienced staff, being distracted while dispensing, seeing other people in the pharmacy becoming stressed, and managing situations with agitated patients. It was noticed that independent pharmacies may be more exposed to this particular category of stress factor because they had less, or no, extra cover for when people were sick. The large multiple store had a group of dispensers which would act as short term locums to fill for people when they were unable to come to work, although this presented a new set of problems

when people were working in a store for the first time, and not knowing where thing were. The following quotes show how being low on staff can lead to rushing processes, with the potential consequences of more errors.

> "Next week we're not going to have the full complement of staff, and I know that there's certain tasks which we are going to have to rush, and obviously if you're rushing you're not paying full attention." (Pharmacy A – Relief Pharmacist)

> "Yeah, when it's busy it does make you more stressed, so you tend to rush more, and make more mistakes... I think it depends on the mood that you're in, if you're tired you could make mistakes." (Pharmacy D – Apprentice Dispenser)

> "If you're short staffed then you're all stuffed aren't you. There is no cover for when people are off sick, that's all when we're off." (Pharmacy C – Dispenser)

Being busy was another frequently reported factor that can lead to a stressful working environment. This is in some ways linked to being low on staff, since having less staff will lead to more work for each person. However, busy days can arrive even if a full complement of staff is present. The quotes below show how busy working conditions are often attributed to be one of the major stress factors in community pharmacies.

> "When you've got an absolute load of stuff to do, and you're rushing and you're stressing. That's when mistakes happen. Especially when things start to get a bit too much and there's people all waiting... and then the phones ringing... That's when mistakes are made." (Pharmacy C – Pre reg pharamcist)

> "It's very variable, across the weeks, sometimes we can be very busy, sometimes we can be quiet. When some of the staff go on their lunch then it will be a headache. This will increase the risk of us making error, so it will really require us to keep ourselves calm and not to rush, and hope that the patient will understand too." (Pharmacy D – Pharmacist owner)

Those with extra responsibilities for checking prescriptions, or managing the store, such as managers or pharmacists, also pointed out that people asking questions and distracting them while they're working can have a negative impact on the outcome of critical tasks.

> "Everything is down to human error... Being the manager and listening to everything that's going on, acknowledging customers... There's a lot going on, the phones ringing, someone'll ask me "can you do this?" It's a really really busy, intense environment." (Pharmacy A – Manager)

> "You've seen how we operate here at the minute. It's quite an open planned pharmacy, so whilst I'm dispensing I'll be having to listen out to a lot of different things, and very often, my process of checking gets disrupted." (Pharmacy A – Regular pharmacist)

As well as being busy or understaffed, other practitioners raised the potential issues of working with less experienced staff. These types of comments were heard more commonly in pharmacies B and D. In pharmacy B, is seemed to be relatively common for relief or locum staff to be working for short periods of time. They would not be familiar with the store they were working in, due to the transient nature of the work. On the other hand, Pharmacy D had a larger complement of trainee staff than was seen elsewhere, and it was recognised that at least initially during the training period, this would require an extra use of concentration and resources.

> "Everyone makes mistakes yeah, but sometimes you know, people who are less well trained, and they're trying to catch their work, and they're trying to zip through it. You know what I'm saying. Sometimes, newbies, like new dispensers, they're more prone to make mistakes." (Pharmacy B – Locum pharmacist)

> "But relief members of staff, so if we had a dispenser coming in to cover, they could potentially make mistakes on not knowing, like, certain brands for certain patients and different little things like that. They wouldn't know like the permanent staff would." (Pharmacy B – Healthcare assistant)

> "Maybe new staff, because in this pharmacy. You can see most of us are still in the training process, which may increase the demand to the pharmacist to supervise, not everyone but most of the others which may make the whole process more demanding for the pharmacist, and to slow down the process to a certain extent." (Pharmacy D – Pre Reg Pharmacist)

Another factor that was highlighted that can make a community pharmacy a high pressure environment is the nature of the work taking place. Receiving medication to treat illnesses is a vital service for people suffering with illnesses, who may feel additional vulnerability based on their situation. The delivery of medication to resolve their health issues is very important to patients. If something is to go wrong, this can potentially lead to a higher pressure environment for those working in the pharmacy.

> "It's their medication and quite a lot of patients can be emotional and upset in regards to the medication they're having to take, their medical history and they may be short tempered. You need to learn to empathize with them, and understand where it is that they're coming from. That they're not personally attacking you, but it's the situation they're in at the moment." (Pharmacy A – Counter assistant)

Finally a dispenser reported how stressful factors can cascade and spread, from one team member to another, causing more added stress within the pharmacy.

> "I must admit when everyone's getting a bit stressed, it kind of stresses you out. Because you're trying to stay calm, and thinking "It's fine we'll do it." But if people start getting agitated with each other and trying to rush you, that doesn't help the situation." (Pharmacy D – Dispenser)

5.4.2.5 Safety procedures

The UK national patient safety agency design for patient safety report (NPSA, 2007) makes reference to intermediate accuracy checks completed by dispensers

while they are working. To further investigate how prevalent these checks were, and at which points in the process they occur, questions were asked whether non-pharmacist staff were also responsible for performing non-formal accuracy checks throughout the dispensing process. It was commonly found that dispensers were doing checks at multiple points throughout the dispensing process. The following three quotes demonstrate how pharmacists from both independent and large multiples, and even less senior members of staff are all well used to the idea of dispensers frequently checking and taking responsibility for the accuracy of their work:

> "I hope they are. Because they need to be responsible for their own work. So when somebody dispenses a prescription, they need to, take responsibility and ownership of what's been dispensed. In terms of the actual product, and the strength, and the quantity." (Pharmacy A – Relief pharmacist)

> "Yeah, yeah, yeah, totally. You expect the dispensers as they're picking, labelling, they'll check and confirm it. And also you'll see that, at the end of the process, I ask the dispensers to then bag up, and I don't ask them to go through it as if the same level you would a checking level, but I do ask them to keep an eye on it, and cast an eye on it, and count the items in and check it in." (Pharmacy C - Owner pharmacist)

> "Slow down, everybody slow down when they're dispensing. Which is difficult because we've all got a lot to do on a daily basis. Everyone needs to just take their time and check everything thoroughly. (Pharmacy B – Healthcare assistant)

Keeping a record of past errors can be a good way to plan and prevent errors of a similar type from repeatedly occurring (Reason, 2000). Understanding how community pharmacies record their errors, and plan to prevent repeat occurrences was another piece of the puzzle with regards to safety procedures. Each pharmacy made an effort to record and monitor errors by keeping a near miss record. In the large multiples, this near miss record was kept on an online system that was stored at the company HQ, whereas the independent stores kept a physical record

of near misses. Practitioners from multiple pharmacies reported that recording the occurrence of near misses was not always possible due to busy working conditions, and the added pressure that can come with people waiting in store to collect their prescriptions. A further negative incentive to record near misses was highlighted by a pharmacist from one of the large multiple stores. If they as a single pharmacy push to record all their errors diligently, it can make their store look bad in comparison to other stores who record errors with less vigour. The quote below highlights difficulties either preventing or discouraging the recording of near misses.

> "You need drivers. You need people in that branch that are committed to safety. Because we now get reports of how many near misses are logged... and we still have [other] stores that have very minimal, and you know that's not right. You've got us, that are right at the top of the table every month... ...we now look like we're the worst branch for errors, and actually if people looked at their own processes a bit more, they might find that we're not." (Pharmacy B – Regular pharmacist)

5.4.2.6 Waiting time

We asked practitioners their view on acceptable waiting times for patients visiting community pharmacies, because understanding the requirements in terms of speed of service is an important aspect to consider when modelling the dispensing process. There is a strong link between patient waiting time, and customer satisfaction (Afolabi, 2003). Two main ideas were presented in response to these questions. The first being a typical estimate of a period of time that might be considered acceptable to patients, typically between 5 and 10 minutes. Allowances were made for large prescriptions containing many items, they might take longer, usually by an increment of time per extra item. The second point that was stressed by participants was that it was important to manage patient expectations with regard to waiting times. If they knew a prescription was going to take longer than 10 minutes, it was considered a priority to make the patient aware of the fact. The patient can then potentially make plans to return later to collect, or at least be forewarned before a long wait. The quotes below highlight the first of these two ideas.

"I would say, generally I would say no more than, probably like 10 to 15 minutes, general time." (Pharmacy A – Counter Assistant)

"I think 10 minutes. I think unless they bring in a 6, 7, page prescription... this morning we were doing them in 5 minutes." (Pharmacy B – Healthcare Assistant)

"I think the staff really need to communicate with the pharmacist about waiting times, because it depends on them. If I've been told I've got 5 trays to check, and then there's a waiter (a patient waiting in the pharmacy for a prescription), and I've been told those trays are going on delivery in 10 minutes... then I'd say to the staff, 'Look, I'm sorry but the waiters are going to have to be an hour'." (Pharmacy B – Locum Pharmacist)

"Well we've been taught that it's within 5 minutes. They need to be out that door, in, out." (Pharmacy C – Dispenser)

5.4.3 Summary

The method and results of the qualitative interviews have been demonstrated with supporting quotations extracted from the transcribed audio recordings. These will be combined with findings from the quantitative data in the next section, to outline the main implications on the CPN model.

5.5 Modelling implications

This section will review some of the main findings of the on site studies, identify key areas that any modelling framework developed could include, to produce more accurate simulations, and ensure that the perspectives of community pharmacy practitioners are represented in the model.

5.5.1 Transition timings

Having collected a sizeable amount of data about how long each stage of the dispensing process takes to complete, the CPN model built in Chapter 6 was updated to use the distributions determined by real data. Version 4 of the CPN model was updated to use the distributions estimated from the combined data set of Pharmacies A and B.

5.5.2 Intermediate accuracy checks

When building the initial CPN of the dispensing process, it was unclear how prevalent the use of intermediate accuracy checks were in the dispensing process. Not all of the checking strategies investigated used intermediate accuracy checks. The interviews have confirmed that it is standard practice for intermediate accuracy checks to take place throughout the dispensing process.

5.5.3 Near miss rate

The initial primary dispensing task error rate used (0.05, (Cohen et al., 2012)), was considered to be too high by interviewees. However, factoring in the effect of intermediate accuracy checks completed by dispensing staff into the near miss rate will reduce the effective rate of near misses. For example, if it is assumed that a dispenser checks their own work after completing a task, and they identify any errors they have made 9 times out 10 (as in the CPN in Chapter 6), the near miss rate of 0.05 is reduced to 0.005 (1 in 200). This reduced figure matches more closely practitioner's perceptions of the rate of near misses they experience in practice.

5.5.4 Near miss reporting

A recurrent theme throughout the interviews while discussing safety procedures was that of the process of recording near misses. Interviewees often pointed out that it can take time to log and record near misses as they happen. This type of activity was not modelled in any of the work in the thesis. In future work, the model could accommodate this phenomenon by taking the dispenser who committed the

error to an inactive node of the CPN for a period of time whenever a near miss is found by the pharmacist, where the time spent in this node is representative of the time taken to deal with the near miss reporting system. Alternative near miss reporting systems, the local hard copy, and the centralised internet based reporting system could be compared using the ACO optimisation framework presented in Chapter 7 by including an additional layer for the type of near miss reporting used.

5.5.5 Waiting times

Interviews give us an understanding of practitioners' beliefs about reasonable waiting times for patients using community pharmacy services. These ranged from 5-15 minutes for a single item prescription, and much longer for patients requiring large multiple item prescriptions. A key mitigating factor used by practitioners when waiting times are large is to inform patients of the expected wait. This manages the expectations of the patient, and reduces any potential inconvenience. A waiting time of over 15 minutes was the classification used for a delayed prescription in the CPN models used in Chapter 6. However, this time limit applied for all sizes of prescription. An adjustment whereby prescriptions which contain more items may take a longer time to complete without being considered delayed may make the model more representative of real practice.

5.5.6 Split packs

Split packs were found to be a common inefficiency in the dispensing process. They could be included into the CPN model by randomly allocating some of the prescription tokens to be split packs. These tokens would be differentiated by a new colour, and the time required to dispense them would be increased. This could be done by using different 'split pack' timing distributions to control the duration of the initial stages of dispensing.

5.5.7 Variable staff

The complexity of the model could be expanded to include different staff types. An experience rating could be given to staff within the pharmacy that effects their

proficiency at each of the tasks. An example of how this could be implemented would be considering how the performance of a team of experienced practitioners is affected when a new inexperienced member of staff is introduced. Inexperienced staff could take longer to complete each of the stages of dispensing, incur a higher error rate while completing tasks, or a combination of the two. This could also include adding the potential for modelling how the number of staff working in a pharmacy varies throughout a day, rather than using a fixed number of staff.

5.5.8 Stocking responsibility

During the observation period, a pharmacist was only seen helping to replace stock onto shelves on one occasion (See Appendix A). The CPN model developed in Chapter 6 allows for all staff to complete this task. It was very common for this task to be the responsibility of a dispenser or multiple dispensers, and the model may better reflect real practice if this task was assigned to be completed by dispensers only. A reflection of this is seen in the non-flexible work pattern given to some set-ups in Section 7.6.1.

5.5.9 Summary

This section has outlined the main modelling implications which were recognised as a result of the study of 4 UK community pharmacy sites. These changes, if implemented, will make the CPN model developed in Chapter 6 more representative of real practice. Key aspects, such as split packs, and the different types of near miss reporting, would not have been considered for modelling without conducting the on-site visits.

Of the modelling implications derived from interviews with staff, some are already implemented in the CPN model shown in Chapter 6, while others are suggested for future work. The modelling implications implemented in the CPN developed in this thesis are as follows: transition timings informed by data, the non-flexible work pattern given to pharmacists, and the use of intermediate accuracy checks used in the dispensing process. All other modelling implications found during interviews are to be implemented in future work.

Many of the modelling implications raised throughout the analysis of these interviews were feasible findings, which would improve a model of the dispensing process. The model in developed Chapter 6 was developed before these on-site studies took place. It was known that timings of the dispensing process would be collected, and the model was developed in such a way that it would be possible to change the distributions attached to transitions to match the real data. However the modelling implications derived from interviews were an unknown quantity before the on-site studies went ahead, and it was therefore difficult to plan for how the findings might impact upon the model design. It is for this reason that the modelling implications highlighted during the interview analysis are not included in the model in Chapter 6. Making changes to the model based on the interview analysis was out of the scope of the project. Findings from the interviews which are present in the model were added based on the literature review, such as intermediate accuracy checks, and the interview analysis has only served to confirm their presence.

5.6 Conclusion

This Chapter has described the process of data collection and analysis of 4 on site studies conducted at UK community pharmacies. The studies consisted of 3 parts: compiling an observational profile of each pharmacy, collecting and analysing quantitative timings of the dispensing process to produce distributions with estimated parameters to model the data, and qualitative interviews with staff. Results for each site for each of the 3 types of analysis have been presented, and a set of modelling implications has been described. The modelling implications outline a set of changes that could be implemented in the CPN developed in Chapter 6 to make the model reflect the findings of these on site studies more accurately.

Chapter 6

A Coloured Petri Net modelling approach for the process of community pharmacy dispensing

6.1 Introduction

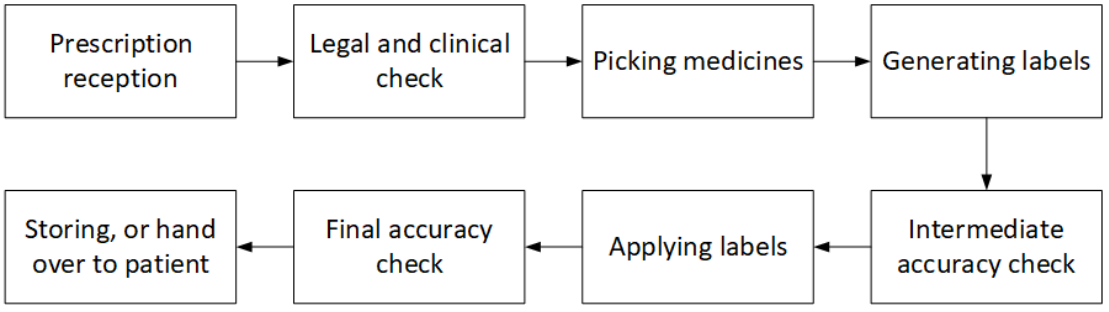
This section outlines the development of a timed Coloured Petri net (CPN) model for simulating the daily operations of a community pharmacy using a manual dispensing method (instead of the automated method (Franklin et al., 2008)). The CPN aims to mimic the behaviour of a team of practitioners as they dispense prescriptions, and complete the required non-dispensing pharmacy tasks of a community pharmacy. 4 versions of the Petri model net are developed, where each iteration improves upon the previous version. Version 1 is a non-coloured Petri net, which was developed initially and did not generate enough detailed information about the process. Version 2 of the model incorporated colours into the modelling framework, which allowed for much easier data collection, and detailed modelling of errors. Version 3 involved incorporating self-concurrency into transitions, which allowed the removal of many parallel sub-systems from the CPN. This reduced the number of places and transitions in the model considerably, allowing for faster run times. Further changes were made in Version 4, where the timing distributions used by transitions were informed by the data collection in Chapter 5. The final Version of the model can be simulated to generate 5 efficiency performance indicators, and 3 safety indicators, of how effective a particular pharmacy set-up is

at completing the required tasks. Version 4 is the final version of the CPN model most representative of real pharmacy practice, it is this Version that is eventually used in the optimisation in Chapter 7.

6.2 The initial PN model: Version 1

Initially a Petri Net approach was proposed for modelling the community pharmacy dispensing process. A Petri Net approach provides a graphical representation of dynamic processes in a discrete event simulation framework, and the flexibility of the technique was desirable for this application. A basic process map to use as a starting point for the modelling process was derived from the description of the dispensing process in Section 2.2.2. Figure 6.1 shows the initial process map. This map was later reduced to a 7 stage process, since modelling the legal and clinical check was considered to be outside the scope of the modelling.

FIGURE 6.1: Process map of the community pharmacy dispensing process

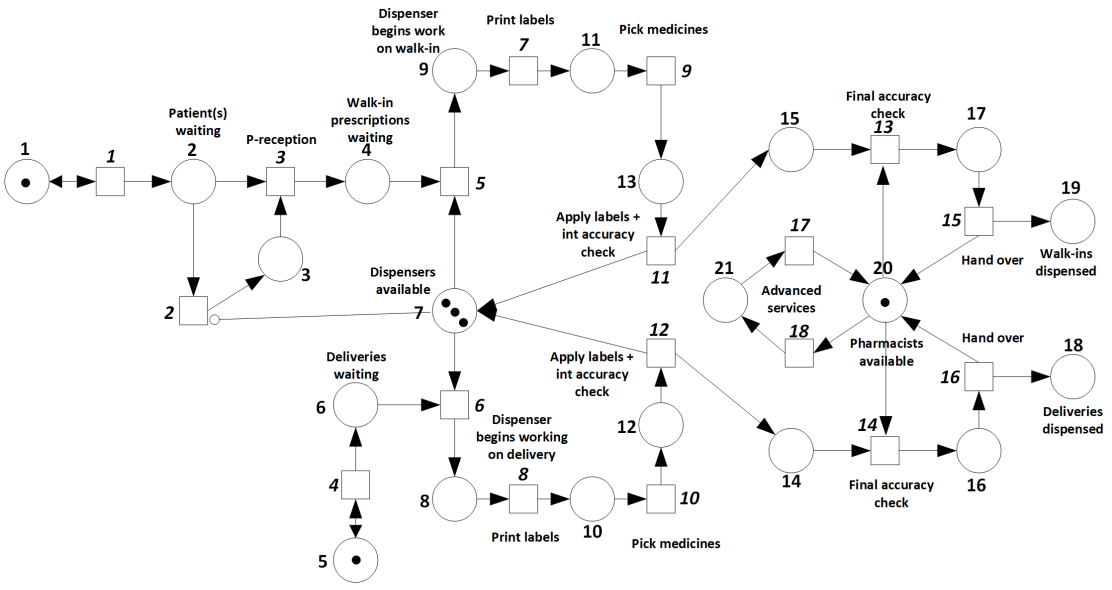


As well as using the process map, known restrictions about which tasks can be completed in the pharmacy by whom were included in the model. For example, pharmacists are the only member of staff capable of completing final accuracy checks, and the set of advanced services. Dispensers are restricted to receiving prescriptions from patients as they arrive at the pharmacy, and completing the 4 primary dispensing tasks; picking, labelling, intermediate accuracy checks, and applying labels to prescriptions.

An initial Petri Net model was constructed, focusing on the physical tasks which move prescriptions through the pharmacy as they are completed. Figure 6.2 shows the initial PN of Version 1. The model was built by considering how a single prescription moves around a pharmacy, and then incorporating the way resources are used in dispensing a prescription. The net dealt with two types of prescriptions,

deliveries and walk-ins, which enter the pharmacy through two constantly enabled transitions (transitions 1 and 4 respectively). Walk-in prescriptions are prioritised over delivery prescriptions, since walk-in customers are waiting in the store to collect their prescriptions and, therefore, require a higher degree of urgency. This prioritisation was done by including small deterministic delays on transitions 5 and 6, with the delay on transition 5 being smaller than the one on transition 6. This prioritisation of transition 5 means that if a walk-in and a delivery are both waiting to be dispensed, the walk-in will be dispensed first. A similar firing control was employed between transitions 13 and 14, to, again, give priority to walk-in prescriptions at the checking stage of the process. The prescriptions are dispensed by dispensers, before being passed on to be checked by a pharmacist. As well as checking, the pharmacist is also responsible for completing advanced services.

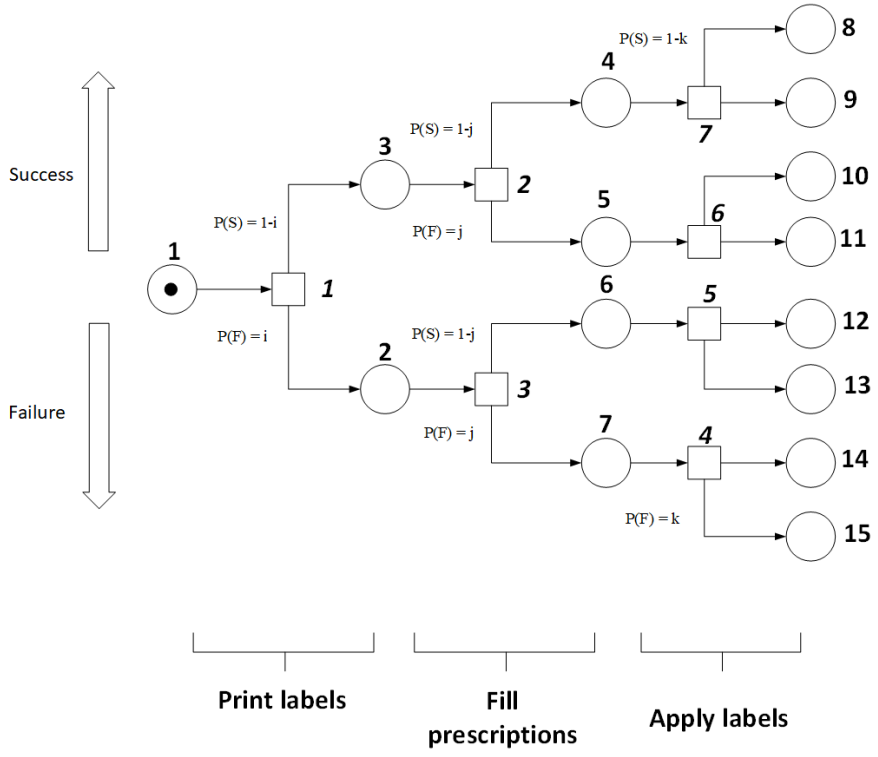
FIGURE 6.2: Initial Petri Net model of a community pharmacy



The model could be simulated and run over a period of time to determine how many prescriptions were dispensed in a working day for variable levels of staff. However, more detailed results were harder to obtain. For example, knowing how long it took to dispense a prescription is a key piece of information, given how sensitive customers can be to waiting times. However, it was initially unclear how to collect this data from the model. A proposed approach to gathering this information was to time the delay between successive firings of transition 1, and transition 15, with the difference being the time taken to complete the prescription.

However, there were some behavioural features of this PN that were not very representative of real practice. In practice, pharmacists are capable of helping with the primary tasks of the dispensing process, such as receiving prescriptions from customers, printing labels, etc. In this non-coloured framework, introducing pharmacists to the initial section of the Petri net was not feasible, since it would be impossible to distinguish between the two types of worker. Another issue was to do with dispensers working concurrently. It can be seen in Figure 6.2 that if two dispensers were to begin working on walk-in prescriptions at the same time, they would not actually be working concurrently. One would have to finish printing labels first before the second could begin printing theirs. This can be seen by modelling the behaviour of the PN if two tokens were placed on place 9. Finally, the introduction of errors into the model was a further difficulty in the non-coloured PN modelling framework. A proposed approach to model errors within this framework was to use conditional transitions. These transitions would have a failure rate, and multiple outcomes. When firing, a sample of a Bernoulli random variable would determine the outcome of the action. A diagram of how this proposition may have been implemented is shown in Figure 6.3.

FIGURE 6.3: Conditional transitions for modelling failures



If i, j, and k, are the probability of failure for each of the printing labels, filling, and applying labels stages of the dispensing process respectively. Figure 6.3 shows

how conditional transitions may be used to model failures for these initial stages. Let each transition have a Bernoulli random variable attached X_i, such that when a transition fires, the outcome of the Bernoulli variable determines which place tokens are placed into. Bernoulli samples returning 1 indicate failures, and conversely Bernoulli samples returning 0 indicate successful completion. For example, if the Bernoulli random variable attached to transition 1, X_1, returns 1 when transition 1 fires, then a token is placed into place 2. If instead X_1 returns 0, when transition 1 fires, then a token is placed into place 3. A similar principle governs the firing of the other transitions, where a successful outcome of the Bernoulli trial, places a token into the place above the firing transition, and a failed Bernoulli trial places a token into the place below a transition. The places 8-15 each describe a unique state prescriptions may be in, after undergoing these 3 stages of dispensing.

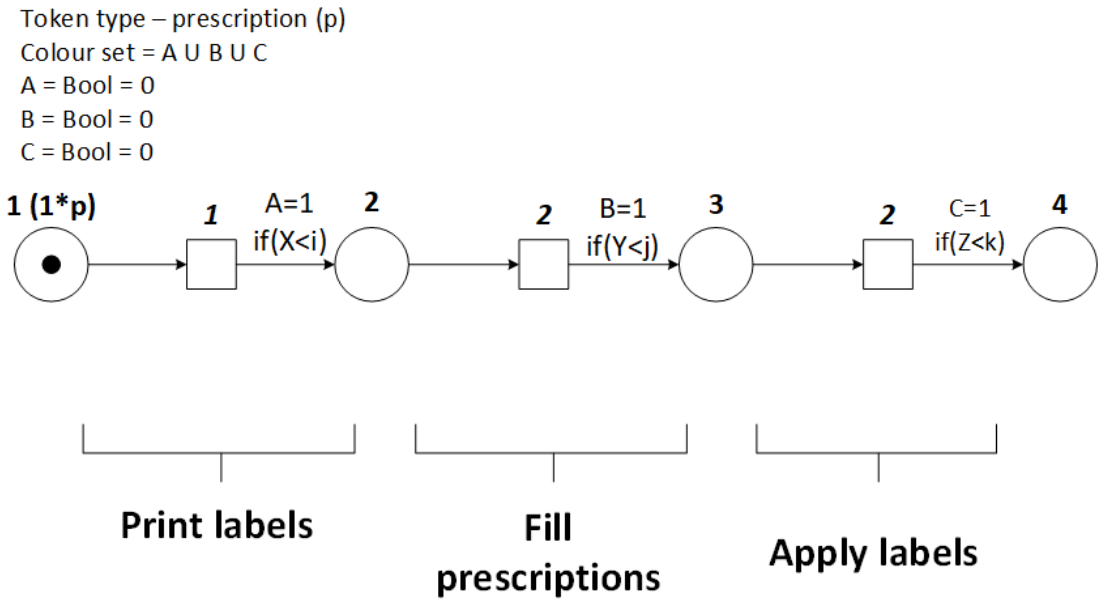
It is clear that the number of places required to model failures using this method grows with the number of fallible stages in the process, since every fallible stage requires at least two potential output places. Another issue is that modelling multiple dispensers completing these tasks concurrently would require two symmetric subnets of this form, which would again increase the number of transitions and places required. Although accuracy checking stages are included in Figure 6.3, these are not fallible in this iteration. Making them fallible with multiple outcomes would add further places and transitions.

After this initial exploration of the Petri Net methodology as a modelling framework for the dispensing process, it was proposed that coloured Petri Nets may be a more effective framework. An example of how nets can be simplified can be seen by considering the modelling of failures in Figure 6.3. This behaviour can be condensed into a much simpler representation with the use of colours, while maintaining identical behaviour. Introducing colours means its no longer necessary to have places representing the outcome (success or failure), for stages of the process. This information can instead be stored on tokens in the form of token colours.

Again failures are modelled using Bernoulli random variables with the same failure rates. However, instead of changing the output place of transitions, now token colours are modified by the outcome of these Bernoulli trials. Figure 6.4 shows how the introduction of colours can reduce the number of places and transitions needed to model these first three stages of dispensing. Prescription tokens are given a colour set consisting of 3 Boolean variables. The variables are initially set to 0, to indicate no failure present. As the prescriptions are moved through the

pharmacy, failures are introduced by changing the corresponding Boolean variable to a number 1 if a failure occurs during a stage of the dispensing process.

FIGURE 6.4: Coloured tokens for modelling failures



Before the introduction of colours, 15 places, and 7 transitions were required to model three stages of the dispensing process. However, by using colours to store information about the outcomes of stages in the process, only 4 places and 3 transitions are needed. Therefore, a CPN framework is proposed in the next section.

6.3 A Coloured Petri Net Framework: Version 2

This section presents the development of a Coloured Petri Net model of the community pharmacy dispensing process. It improves upon the previous model by making use of coloured tokens to enable a much more detailed analysis of token behaviour, detailed modelling of failures, and the ability to model different types of staff member. Additional non-dispensing tasks, such as lunch breaks for dispensers and stock management, as well as the ability to model concurrent working, are also introduced into the model.

6.3.1 Building the model

Version 2 was designed to simulate a community pharmacy using a manual dispensing method with variable numbers of dispensers and pharmacists. The staff are now capable of working in parallel to complete prescriptions. Figure 6.5 shows the full graph of the CPN used to model the process.

6.3.2 Token types

Combinations of colour sets are used to define a set of distinct token types. Version 2 uses three token types: prescription (p), worker (w), and basic tokens (e), to represent the resources used in the process. The worker token type has one colour field used to indicate the role of the worker token (as either dispenser or pharmacist), and a further set of colours which correspond to the tasks they are allowed to complete. As an example, dispenser tokens have colour fields corresponding to 4 activities, primary dispensing, lunch breaks, idle, and receiving prescriptions. The prescription token type, (p) uses a larger set of 8 colours, which represent:

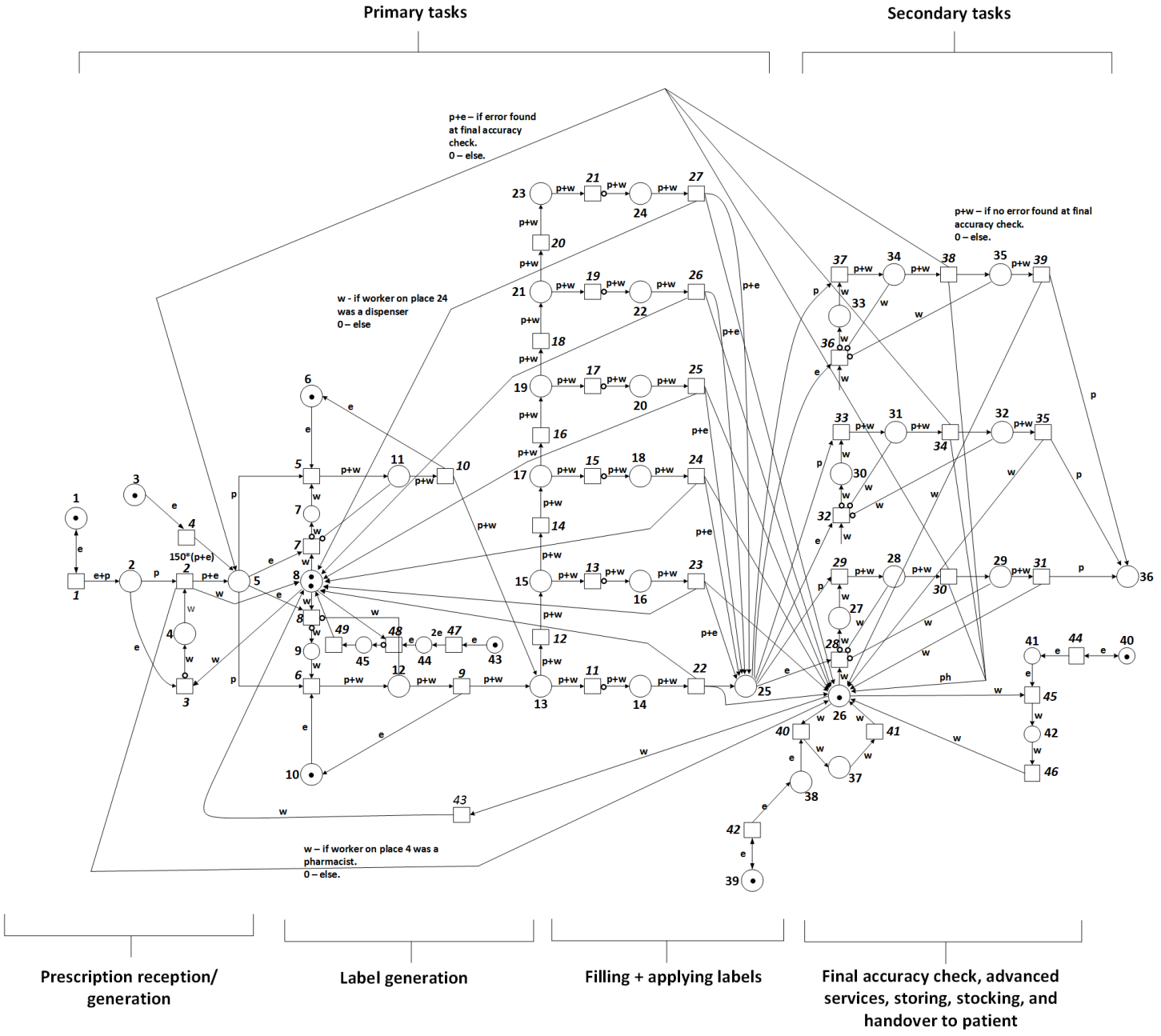
1. Delivery or walk-in
2. The time taken to dispense the prescription
3. Number of iterations to complete the prescription, if faults occur.
4. The overall outcome of the prescription (near miss, dispensing error, or completely correct)
5. The number of items in the prescription
6. Labels
7. Contents
8. Label application

Basic tokens (e) have no colours attached to them, and they are used in the CPN to permanently enable transitions, model the availability of labelling stations, and count the number of times a pharmacist has completed an advanced service.

6.3.3 Places and transitions

Table 6.1 lists each place, along with a description of the place, and the token types that are allowed to occupy each place. Places are used for different purposes

FIGURE 6.5: A coloured Petri net for modelling community pharmacy dispensing



in the net. Some are used to hold resources until they are needed, others show at which stage prescriptions are in the dispensing journey. Alternatively, some places are used to generate net tasks, by keeping a transition constantly enabled. The behaviour is governed by the general principle that staff resources wait in place 8 and place 26 for tasks to arrive in the pharmacy. When tasks are generated, staff are allocated to complete tasks using semi-instantaneous transitions, with very small delays. After completing a given task, staff are returned to either place 8 or place 26. Dispensers wait in place 8 and pharmacists in place 26.

TABLE 6.1: Places

Place	Description	Token Type
1	Walk-in prescription generator.	e
2	Customer waiting at counter.	p + e
3	Delivery prescription generator.	e
4	Member of staff is on the counter ready to receive a prescription.	w
5	Prescriptions waiting to be dispensed.	p + e
6, 10	Labelling stations available.	e
7, 9	A member of staff is choosing a prescription, to complete primary tasks.	w
8	Staff available to complete primary dispensing tasks.	w
11-12	A member of staff is generating the labels of a prescription.	w + p
13, 15, 17, 19, 21 and 23	These places are used to separate staff into parallel work streams.	w+p
14, 16, 18, 20, 22, and 24	Staff are assembling, and applying labels to a prescription.	w + p
25	Prescriptions waiting for secondary dispensing tasks to be completed.	p + e
26	Pharmacists available to complete secondary dispensing tasks.	w
27, 30, 33	A pharmacist is allocated to complete secondary tasks for a prescription.	w
28, 31, 34	Pharmacist is completing a final accuracy check.	w + p
29, 32, 35	Pharmacist is handing over medicines to a patient/storing a delivery for collection later.	w + p
36	All completed prescriptions.	p
37	Advanced service being completed.	w
38	Advanced service waiting to be completed.	e
39	Advanced services task generator.	e
40	Stocking task generator.	e
41	Stocking task waiting to be completed.	e
42	Stocking task being completed.	w
43	Dispenser lunch break generator.	e
44	Lunch break is ready to be taken.	e
45	A dispenser is on their lunch break.	w

Table 6.2 shows a list of transitions, their distribution used for transition timings, a small description of what a transition controls, and a letter Y in the 4th column designates some transitions as processor transitions (events which are affected by the number of items in a prescription). For example, generating labels is a processor transition, since it will take longer to generate a set of labels for a prescription containing a large number of items. However, choosing a prescription

to work on is not a processor task, since this should not depend upon the number of items. The distributions for this version were generated through discussion with experts, and do not use the real data.

Table 6.2: Transitions

Transition	Description	Timing (seconds)	Processor (Y/N)
1	Walk-in generation.	Exponential(0.0033)	N
2	Dispenser receives prescription.	Uniform(30, 60)	N
3	Available staff member moves to counter.	Deterministic(ϵ)	N
4	Delivery generation.	Deterministic(6000)	N
5-6	Staff member chooses a prescription to begin working on.	Uniform(5, 10)	N
7-8	Allocating a worker to complete a primary task.	Determinstc(ϵ)	N
9-10	Label generation (processor)	Deterministic(15)	Y
11-21	Spreaders.	Deterministic(ϵ)	N
22-27	Filling and applying labels.	Normal(50, 10)	Y
28, 32, 36	Pharmacist is allocated to complete secondary dispensing tasks.	Deterministic(ϵ)	N
29, 33, 37	Pharmacist chooses a prescription, to begin completing secondary dispensing tasks.	Uniform(10, 15)	N
30, 34, 38	Pharmacist carries out a final accuracy check.	Uniform(5, 10)	Y
31, 35, 39	Pharmacist hands out prescription to patient, with counselling.	Exponential(0.025)	N
31, 35, 39	Pharmacist stores delivery prescription to be collected later.	Exponential(0.05)	N
40	Allocate pharmacist to complete an advanced service.	Deterministic(ϵ)	N
41	Pharmacist completes an advanced service.	Uniform(300, 600)	N
42	Advanced service task generator.	exponential(0.00006)	N
43	Pharmacist moves to complete a primary dispensing task.	Deterministic(10)	N
44	Generate a stocking task to be completed.	Deterministic(6600)	N
45	Allocate pharmacist to complete a stock management task.	Deterministic(ϵ)	N
46	Pharmacist finishes a stock management task	Uniform(300, 900)	N
47	Begin triggering of lunch break for dispenser.	Deterministic(7200)	N
48	Allocate a dispenser to to take a lunch break.	Deterministic(ϵ)	N
49	Dispenser finished their lunch break.	Deterministic(3600)	N

Some transitions are representative of community pharmacy dispensing tasks (e.g. 2, 9-10, 22-27, etc.), for example labelling, filling or new prescriptions entering the system, while other transitions (e.g. 3, 7-8, 11-21, etc.) are for controlling the way tokens are allocated around the net, and are not representative of any real task. The latter are used to separate tokens into parallel streams. The CPN models a pharmacy in which up to 6 people may assemble prescriptions in parallel, and up to 3 people may accuracy check simultaneously. This is a reflection on the relative size of the pharmacy being modelled. If a specific pharmacy is capable of having more or less people work simultaneously on similar tasks, the model is able to accommodate this by adding or removing these parallel structures.

Transition timings were generated through in depth discussions with two experienced community pharmacists. All interviewees had previous experience working in community pharmacy dispensaries. The interviews generated the estimates of the mean duration of each stage, while the distribution types have been assumed.

Where appropriate, the Poisson distribution was used to model the arrival of patients at the pharmacy for walk-in prescriptions and advanced services. Note that in-field data collection and analysis is presented in Chapter 5, and it is used to in form Version 4 of the CPN model in Section 6.8.

6.3.4 Model assumptions

A key feature of the model is the distinction between primary and secondary tasks, as shown in Figure 6.5, and between the two types of staff, dispensers and pharmacists. The model designates each task as either primary or secondary, where any member of staff may complete primary tasks, but *only* pharmacists may complete secondary tasks. This is implemented by using transition bindings on the set of input transitions of place 25. If the staff token which enabled one of these transitions was a dispenser, they are sent back to place 5. However, if the staff token used to enable one of the transitions was a pharmacist, they are moved to place 26. All secondary tasks require resources (or tokens) from place 26 instead. Each staff tokens is differentiated by a colour field, which indicates whether the token is a dispenser, or a pharmacist. As well as this distinction, the model uses further assumptions about staff behaviour, and pharmacy resources. The assumptions about staff behaviour are as follows:

- Staff complete tasks in an identical way, i.e. the same probability distributions are used to determine how long tasks take to complete, and error probabilities are the same.
- Primary dispensing tasks include: receiving prescriptions, generating labels, assembling prescriptions and applying labels. Secondary tasks include: final accuracy checking prescriptions, completing advanced pharmacy services, stock management, handing prescriptions over to patients (along with patient counselling) and storing medicines for delivery.
- Pharmacists *prioritise* completing secondary dispensing tasks, but they may also contribute to completing primary tasks. If a pharmacist is in place 26 with no secondary tasks to complete, after a 10 second delay, the pharmacist will instantaneously move to complete a primary task.

Assumptions about labelling stations, pharmacy opening hours, and prescriptions, are given below:

- There are 2 labelling stations (A labelling station is a computer linked to a small printer, capable of producing labels).
- When a customer arrives in the pharmacy, the first member of staff available to complete primary tasks moves over to the counter to receive the prescription. If multiple staff are available, the member of staff that has been idle for the longest time receives the prescription.
- Once a member of staff begins generating labels for a prescription, they continue working on the same prescription until the labels have been applied, i.e. the same person will generate labels, collect the contents, and apply the labels consecutively.
- Walk-in prescriptions are prioritised over delivery prescriptions, and there is a "first come first served" order of completion.
- The pharmacy is open from 9am to 5pm.
- 150 delivery prescriptions arrive at the pharmacy in a single large bulk of orders, at 10am, 1 hour after the pharmacy opens.
- Walk-in prescriptions and advanced services arrive with increments of exponential distributions.

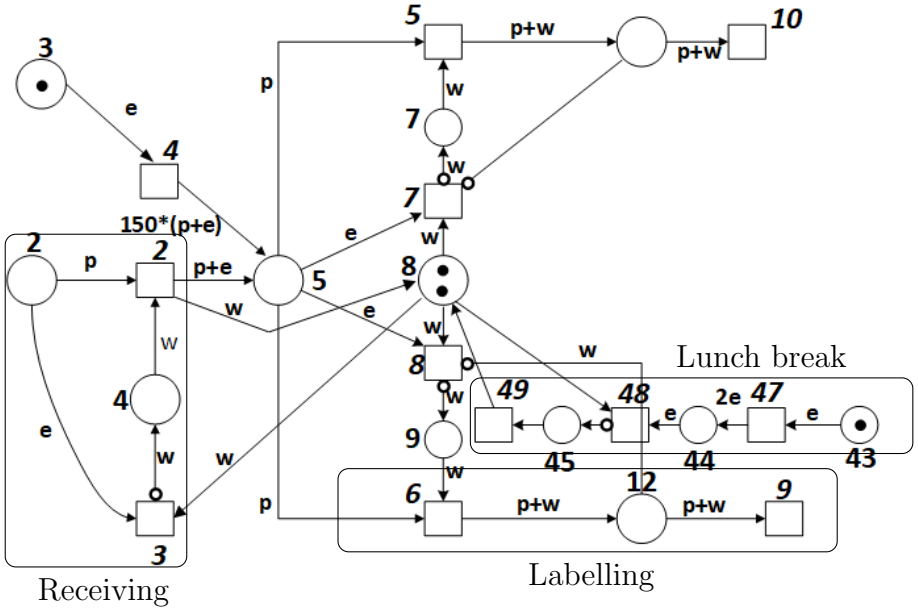
6.3.5 Resources

Simulations begin with no prescriptions in the pharmacy. The first transition to fire in the net will be one of 3 potential transitions: a patient will arrive requiring a prescription (transition 2), a patient will arrive requiring an advanced service (transition 42), or if no patients arrive within the first hour, the delivery of prescriptions at 10am will be the first transition to fire (transition 4). Dispensers and pharmacists are initially placed in places 8 and 26 respectively. The only resource modelled by basic tokens, are the labelling stations used to create and print labels. These are represented in Figure 6.5 by the basic tokens on places 6 and 10.

6.3.6 Primary task allocation

Staff tokens wait in an 'available' place, until a task needs to be completed. Once a task becomes available, the member of staff who has been waiting the longest is allocated to complete that task. Figure 6.6 is an enlarged part of Figure 6.5, showing workers waiting to be allocated to various primary tasks.

FIGURE 6.6: Task Allocation



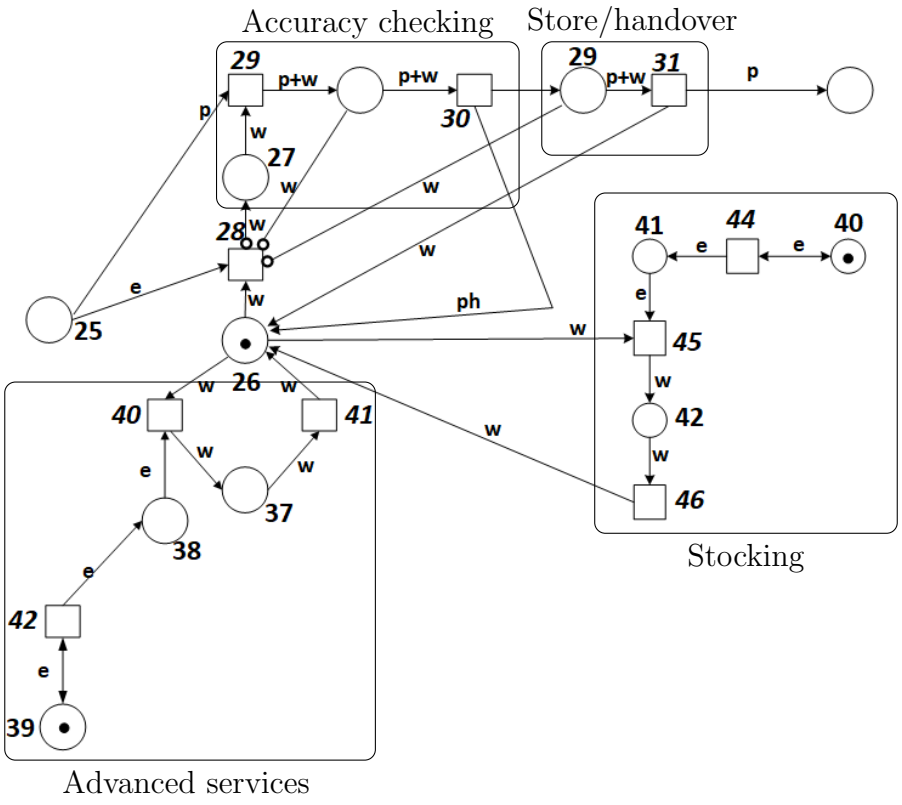
As tasks arrive, workers are allocated using hierarchical instantaneous transitions. The hierarchy controls which task workers are allocated to if there are multiple tasks needing completion. The sizes of the transition's delays are used to settle firing disputes. Figure 6.6 shows the allocation transitions for three primary dispensing tasks which require worker resources.

The hierarchy of primary tasks from the top priority to the lowest priority is as follows: receiving prescriptions, lunch break, and starting to dispense a prescription. Thus transition 3 is given priority over transition 48, which is given priority over transitions 7 and 8. This is done in the model by using deterministic transitions with very small delays of increasing size, to settle firing conflicts. Receiving prescriptions is the top priority, since patients waiting at the counter need to be seen quickly. Lunch breaks are next in the list, above starting to dispense prescriptions, since these occur infrequently compared to dispensing, and these tasks may never be completed if dispensing were prioritised, since the constant stream of dispensing would deny staff a break.

6.3.7 Secondary task allocation

Figure 6.7 shows the net section used to allocate pharmacists to secondary tasks. Similarly to the transitions used to allocate staff to primary dispensing tasks, the

FIGURE 6.7: Secondary tasks



transitions for secondary tasks use a hierarchy determined by their frequency. The hierarchy from highest to lowest is as follows: advanced services, stock management, followed by completing secondary dispensing tasks. This order is chosen so that infrequent tasks are prioritised over more frequent ones. Again deterministic transitions with small delays of increasing size are used to settle firing disputes when multiple transitions enable simultaneously.

6.3.8 Failures

The occurrence of failures is modelled in the CPN using Bernoulli random variables. The processes of label generation, prescription assembly and label application introduce a probability of a labelling error, contents error, or label application error respectively. The failure probabilities used for each stage can be found in Table 6.3. These probabilities were taken from Cohen et al (Cohen et al., 2012). It was then assumed that label generation may be twice as likely to fail as applying completed labels to prescription boxes.

Table 6.3: Error Probability

Task	Description	Probability (%)
1	Labelling	0.06
2	Filling	0.05
3	Label Application	0.03
4	Final Accuracy check	0.05

The probability of errors is introduced to the CPN using arc inscriptions, which have mostly been omitted from Figure 6.5 due to space restrictions. The final accuracy check outcome is conditional on the state of the prescription being checked. If a prescription is error free, it is assumed that the prescription passes through the final accuracy check to the next stage of the process.

If the prescription being checked contains an error, the pharmacist attempts to spot the error at the final accuracy check. If an error is spotted, the prescription is sent back to place 5 to be dispensed again. If a prescription containing an error is checked, and the error is not found, the erroneous prescription will continue through the dispensing process. These are classified in the results as dispensing errors.

6.3.9 Item modelling

Each prescription is assigned a random number of items, by sampling from a Geometric(0.35) distribution. This distribution has a mean of 1/0.35, and has been assumed, based on the following reasoning. Assume a patient goes to a doctor and is given a prescription. The first assumption is that there will be at least 1 item in the prescription. The second assumption is that prescriptions containing more items are less likely to occur than those containing fewer.

The number of items in a prescription affects the duration of the processor stages of the dispensing process (indicated as processors in Table 6.2). To simulate this, transitions representing processor stages sample from their distribution a number of times equal to the number of items in the prescription being processed, and the sum of those samples is used as the delay.

6.4 Task generation

To model a community pharmacy, there needs to be some consideration of how prescriptions, and other non-dispensing tasks enter the pharmacy. Tasks can be separated into two distinct categories, repeated tasks and one off tasks. Walk-in prescriptions, which enter the pharmacy as customers bring them in throughout the day are repeated tasks, along with advanced services and other non dispensing tasks which need to be completed periodically throughout a day of operation (e.g. stock management). Delivery prescriptions, which all arrive at the pharmacy at the same time once a day, are a one-off task.

Task generation is controlled by simple structures in the Petri net. One-off and repeated tasks are both controlled by a similar net substructure containing two places and a transition. Repeated tasks are generated by a place containing 1 basic token, connected by a double edged arc to a transition (Figure 6.8 a). While one off tasks are generated using a similar structure, however the edge connecting place to transition, only runs in one direction (see Figure 6.8 b).

In Figure 6.8 a., the double edged arc ensures that a token is permanently on p1, and thus tokens are fed into place p2 at a rate controlled by increments of the delay timing of transition t1. In 6.8 b., after transition t1 has fired once, 150 tokens (representing a one off task) are placed onto place p2, and the token on p1 is removed. This ensures that only one instance of the task is generated. If a task needs to be completed a fixed number of times, one can use the structure of 6.8 b. with a number of tokens on p1.

FIGURE 6.8: Task generation

a. Repeated task generation

b. One-off task generation

In both instances, colours have been included to the arc running from t1 to p2 to indicate which type of prescription is being generated, walk-in or delivery. In the repeated task case, only a single walk-in prescription arrives into p2. Whereas in the one off task generation, a large batch of 150 delivery prescriptions arrive simultaneously. Note that in Figure 6.8 the arc weight P(WI) indicates that a walk-in prescription is placed onto output places when the connected transition fires. P(D) indicates the same but for delivery prescriptions.

Examples of these task generation substructures can be seen in Figure 6.5, place 1 and transition 1 repeatedly feed randomly timed walk-in prescriptions into the Petri net. Place 3 and transition 4 are used to generate 150 delivery prescriptions 1 hour after the pharmacy opens, and other similar structures are used to generate the other community pharmacy tasks.

6.4.1 Task classification

Table 6.4 below shows a full list of activities which are required to be completed by the community pharmacy being modelled, whether the task is related to the dispensing process, and whether the task is considered to be a repeated task, or a one-off.

Table 6.4: Community pharmacy activity classification

Activity	Dispensing related (Y/N)	Repeated/One off
Walk-in prescriptions	Y	Repeated
Delivery prescriptions	Y	One off
Stock management	N	Repeated
Advanced services	N	Repeated
Lunch hour	N	One off

To include all the potential non-dispensing activities, such as time spent on staff training, housekeeping, and non-professional encounters was not feasible for this version of the model. The model currently includes the completion of prescriptions, stock management tasks, and advanced services. Further versions of the CPN could be expanded to include a wider variety of tasks, to be more in line with the work sampling literature. From the perspective of the dispensing process, the reason why staff are taken out of the process for periods of time is not relevant.

Dispensing delivery and walk-in prescriptions, can be further broken down into 6 key steps, receiving, label generation, contents picking, label application, final

accuracy checking and handing over to the patient with counselling or storing for delivery. A seventh step, the intermediate check can be included, however it was not known during the initial modelling stage how commonly used such checks were. The process then becomes a 6 or 7 stage process, where the legal and clinical check are not considered, and the intermediate check is optional. Table 6.5 below shows a breakdown of who is qualified to complete each stage, and whether the stage of the process will take longer for bigger prescriptions.

Table 6.5: Dispensing stages

Stage	Staff qualified to complete stage	Size dependent (Y/N)
Receiving	All	N
Label generation	Dispensers, ACTs, pharmacists	Y
Contents picking	Dispensers, ACTS, Pharmacists	Y
Applying labels	Dispensers, ACTs, Pharmacists	Y
Final accuracy checking	ACTs or Pharmacists	Y
Handing over to patient	Pharmacists	N
Storing for delivery	All	N
Patient counselling	Pharmacists	N

As well as parts of the dispensing process being limited to only pharmacists, this version of the model assigns pharmacists to complete all the other non dispensing activities, which include stock management and advanced services. An alternative work pattern, where pharmacists do not contribute to the completion of primary dispensing tasks, is explored in Section 7.6.

6.4.2 Results

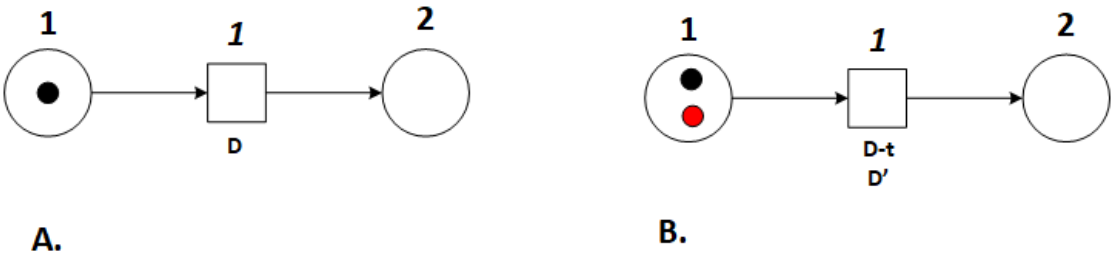
Results are generated from the CPN by inspecting place 36. The number of prescription tokens present in the place indicates how many prescriptions have been completed throughout the day. Further information about how long each prescription took to be completed, how many dispensing errors or near misses were present can be derived from inspecting the token colours. Since this is an intermediate version of the CPN developed to model the dispensing process, the full detailed sets of results generated for this iteration will not be presented here. If the reader desires, in depth presentations of the results generated by this version of the CPN model can be seen in (Naybour et al., 2018b), and (Naybour et al., 2018a). Both of these papers derived results using this version of the CPN.

6.5 A second iteration of the CPN: Version 3

After designing the first iteration of the CPN model, it was proposed that it may be possible to simplify the net structure by incorporating self concurrent transitions into the modelling framework. Such transitions are in the framework presented by Jensen (Jensen, 1996), and it was proposed that by including this property, a number of symmetric subnets within the graph of Figure 6.5 would be modelled by a single net instead. The simpler representation would mean that fewer places, transitions, and arcs would be required to model the pharmacy, and it was hypothesised that this would enable faster simulation times, as well as give a simpler graphical representation.

Self concurrent transitions may be enabled more than once. Figure 6.9 below demonstrates the principle. Initially in A., the transition is enabled a single time by the black token. The transition will fire after a delays of D, moving the black token from place 1 to place 2. However, imagine that if before a delay of D has passed, another token enters place 1 after a delay of t. Then the transition samples the distribution attached to it again, this time binding to the second token. In B. the transition now has two delay times attached to it, $D - t$, and D'. Note that it is possible that $D' < (D - t)$, in which case the red token would reach place 2 before the black token.

FIGURE 6.9: Self concurrency



Improvements to the way staff are allocated to tasks were also proposed. Instead of deterministic transitions with small delays of increasing size, instantaneous transitions (with a delay of 0), each with a priority parameter to determine which transition has priority in cases of firing conflicts, were proposed. These are improvements upon the previous method because the small delays of variable size being used before were not modelling any particular action, they only delayed staff

moving to complete tasks by a small amount of time. The technique has been used before (Marsan, 1990) in a Generalised Stochastic Petri Net framework, and the same principle has previously been applied to CPNs.

Additionally, since the model was being redesigned, some further improvements to the behaviour of staff and task allocations were made to the CPN. The stock management task was changed to a primary task, able to be completed by all staff, since this more closely matches the reality in practice. Furthermore, the way pharmacists move to complete primary tasks was updated. In Version 2, a pharmacist would move to complete a primary task after 10 seconds of inactivity while waiting for a task to complete. In Version 3, pharmacists move to complete a primary task instantaneously, but only if there are no dispensers available to complete a primary task at that moment. This was done by introducing an inhibit arc from place 6 to transition 22 in Figure 6.10. This change stops pharmacists moving to the primary area of the pharmacy at the beginning of simulations. They remain waiting to complete a secondary task, unless the primary area of the pharmacy becomes empty.

It was also thought that it may be desirable to evaluate the demand upon the pharmacy. Two new places were added to the CPN with the sole purpose of counting the number of arrivals of walk-in prescriptions, and patients requiring advanced services. These are places 26 and 27 in Figure 6.10. Aside from the exceptions mentioned in this section (6.5), all other assumptions about staff behaviour, error rates, resources, items, failures, and prescription modelling remain as described for Version 2 of the model.

6.5.1 Places and transitions

The updated CPN, which makes use of self-concurrent transitions, can be seen in Figure 6.10. The structure is similar to that seen in Figure 6.5, however the parallel lanes have been removed. This has simplified the model substantially. There are 18 fewer places, 28 fewer transitions, and 81 fewer arcs. Although, this would rise to 20 fewer places and 83 fewer arcs if the new places (26 and 27) and arcs used for counting arrivals in Figure 6.10 were discounted.

Table 6.6 describes each place in the new CPN. The behaviour is governed by the same general principle that staff resources wait in places 6 and 16 for tasks to

FIGURE 6.10: A coloured Petri net for modelling community pharmacy dispensing

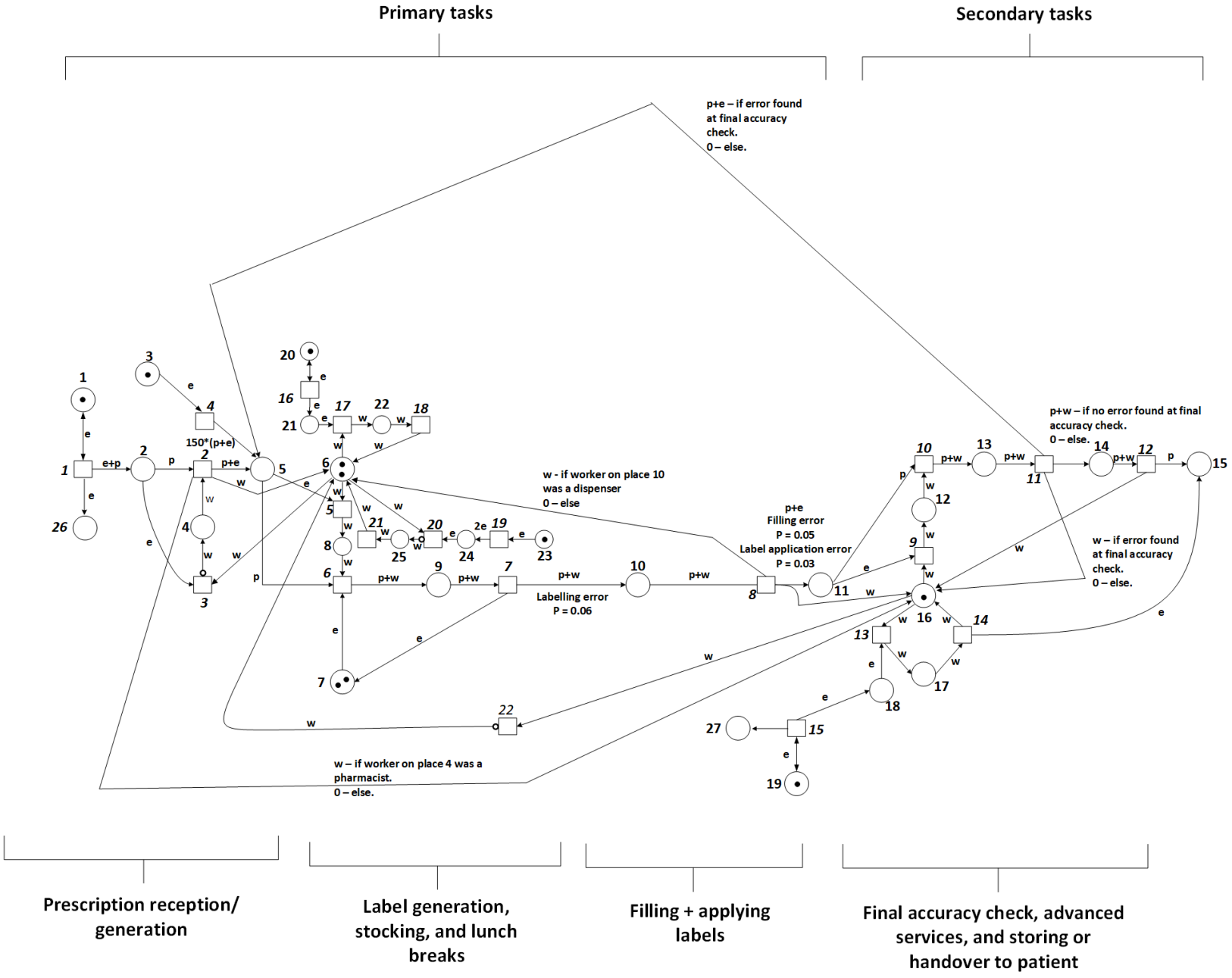


Table 6.6: Model places

Place	Description	Token Type
1	Walk-in prescription generator.	e
2	Customer waiting at counter.	e
3	Delivery prescription generator.	e
4	Member of staff is on the counter ready to receive a prescription.	w
5	Prescriptions waiting to be dispensed.	p + e
6	Staff available to complete primary dispensing tasks.	w
7	Labelling stations available.	e
8	A member of staff is choosing a prescription, to complete primary tasks.	w
9	A member of staff is generating the labels of a prescription.	w + p
10	Staff are assembling, and applying labels to a prescription.	w + p
11	Prescriptions waiting for secondary dispensing tasks to be completed.	p + e
12	A pharmacist is allocated to complete secondary tasks for a prescription.	w
13	Pharmacist is completing a final accuracy check.	w + p
14	Pharmacist is handing over medicines to a patient/storing a delivery for collection later.	w + p
15	All completed prescriptions.	p
16	Pharmacists available to complete secondary dispensing tasks.	w
17	Advanced service being completed.	w
18	Advanced service waiting to be completed.	e
19	Advanced services task generator.	e
20	Stocking task generator.	e
21	Stocking task waiting to be completed.	e
22	Stocking task being completed.	w
23	Dispenser lunch break generator.	e
24	Lunch break is ready to be taken.	e
25	A dispenser is on their lunch break.	w
26	Arrivals counter.	w
27	Advanced services counter.	

arrive in the pharmacy. When tasks are generated, staff are allocated to complete tasks using instantaneous transitions with priority parameters to determine which tasks are prioritised. After completing a task, dispensers are returned to place 6, pharmacists to place 16.

Table 6.7: Model transitions

Transition	Description	Timing (seconds)	Processor (Y/N)
1	Walk-in generation.	Exponential(0.0033)	N
2	Dispenser receives prescription.	Uniform(30, 60)	N
3	Allocate an available staff member to the counter.	Deterministic(0)	N
4	Delivery generation.	Deterministic(3600)	N
5	Allocate a worker to complete a primary task.	Determinstc(0)	N
6	Staff member chooses a prescription to begin working on.	Uniform(5, 10)	N
7	Label generation (processor)	Deterministic(15)	Y
8	Filling and applying labels.	Normal(50, 10)	Y
9	Pharmacist is allocated to complete secondary dispensing tasks.	Deterministic(0)	N
10	Pharmacist chooses a prescription, to begin completing secondary dispensing tasks.	Uniform(10, 15)	N
11	Pharmacist carries out a final accuracy check.	Uniform(5, 10)	Y
12	Pharmacist hands out prescription to patient, with counselling.	Exponential(0.025)	N
12	Pharmacist stores delivery prescription to be collected later.	Exponential(0.05)	N
13	Allocate pharmacist to complete an advanced service.	Deterministic(0)	N
14	Pharmacist completes an advanced service.	Uniform(300, 600)	N
15	Advanced service task generator.	exponential(0.00006)	N
16	Generate a stocking task to be completed.	Deterministic(6600)	N
17	Allocate a member of staff to complete a stock management task.	Deterministic(0)	N
18	Worker finishes a stock management task	Uniform(300, 900)	N
19	Begin triggering of lunch break for dispenser.	Deterministic(7200)	N
20	Allocate a dispenser to to take a lunch break.	Deterministic(0)	N
21	Dispenser finished their lunch break.	Deterministic(3600)	N
22	Pharmacist moves to complete a primary dispensing task.	Deterministic(10)	N

Table 6.7 describes each transition in the new CPN. Transition delay distributions and parameter values have been kept consistent with the values already used in the previous CPN, aside from changing the small deterministic delays to instantaneous transitions. The distributions are still informed from conversations with experts in this Version.

6.5.2 Token creation and continuity

Figure 6.10 uses a non-standard Petri net notation. This was used to simplify the appearance of the model. A guiding principle of the community pharmacy CPN is that tokens are generally conserved throughout the firing of transitions. This can be seen by comparing the number of input prescription tokens, with the number of output prescription tokens seen on each transition. Generally there is equality of input weights and output weights of resource (worker, and prescription) tokens.

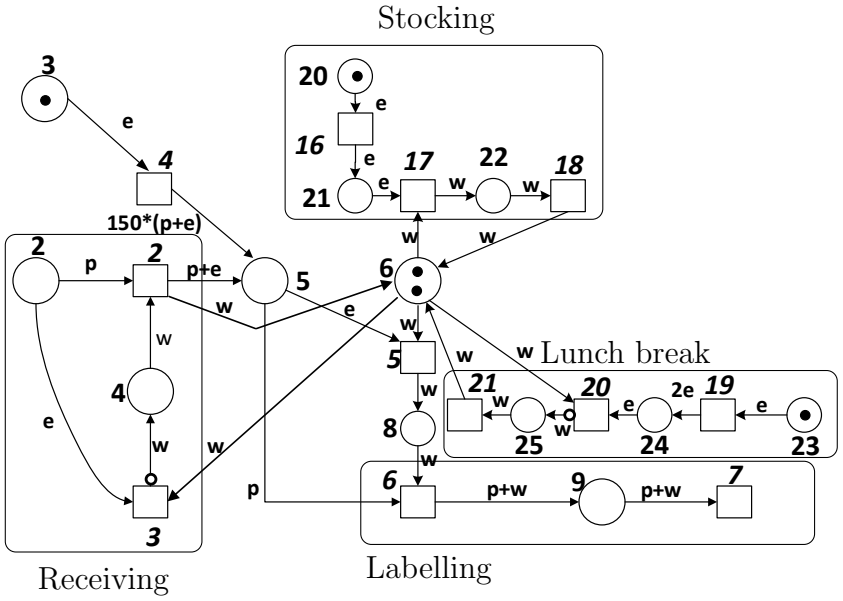
However, there are some cases where the equality between prescription tokens enabling a transition, and the prescription tokens being placed onto output places does not hold. Transitions 1 and 4 do not have this property. No prescription tokens are used to enabled these transitions, yet prescription tokens are placed onto their output places. This is because these transitions represent new prescriptions arriving from outside the pharmacy. The new prescriptions are generated by sampling a Geometric(0.35) random variable to populate a number of items, and an indicator as to whether the prescriptions are walk-ins or deliveries. All other colours, related to the time to dispense, or errors present are empty at the time new prescription tokens are generated.

The other feature that is presented in a non-standard way in Figure 6.10 are the conditional transitions. These are transitions 8, and 11. They both represent accuracy checking stages, which can have multiple outcomes depending on the success or failure of a check. As an example, consider transition 11. Transition 11 has a single input arc with a weight of 1 worker token and 1 prescription token. Additionally, it has 3 output arcs, which when combined, have a weight of 2 prescription tokens and 2 worker tokens. This would appear to break the input-output equality of resource tokens. However, the way the output arcs are constructed means that only 1 prescription and 1 worker token will be placed onto output places. The output arc between transition 11 and place 14, and the pair of output arcs from transition 11 to places 5 and 16 are mutually exclusive. Either tokens will be placed onto place 14, or tokens will be placed onto places 16 and 5. Which set of arcs are used, depends whether the pharmacist finds an error during the accuracy check. If an error is found, the arc from transition 11 to place 14 is used, otherwise the other pair of output arcs are used.

6.5.3 Primary task allocation

Figure 6.11 is an enlarged section of Figure 6.10, showing workers waiting to be allocated to various primary tasks. Note that in the updated version there is only one lane for labelling, and stocking tasks, controlled by places 20, 21 and 22, which now draw resources from the primary worker pool on place 6. The new hierarchy of primary tasks from top priority to lowest is: receiving prescriptions, lunch breaks, stock management and starting to dispense a prescription. Hence, transition 3 is given priority over transition 20, which is given priority over transition 17, which is prioritised over transition 5. Receiving prescriptions is the top priority, since patients waiting at the counter need to be seen quickly. Lunch breaks and stock management are next, above starting to dispense prescriptions, since these are infrequent compared to dispensing, and these tasks may never be completed if dispensing were prioritised, since the constant stream of dispensing would deny staff a lunch break, or the possibility of completing the stocking task.

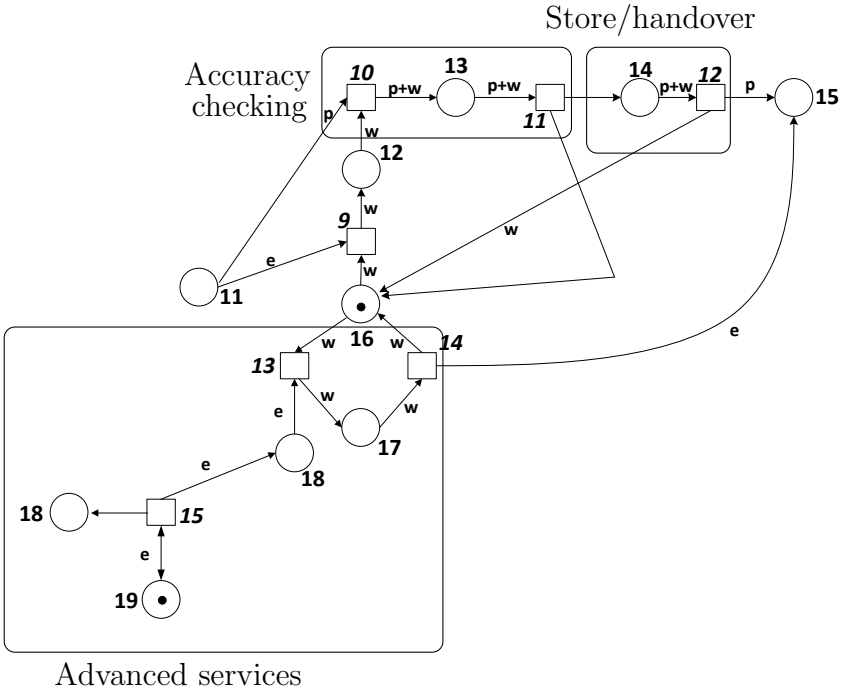
FIGURE 6.11: Primary tasks



6.5.4 Secondary task allocation

Figure 6.12 shows the net section used to allocate pharmacists to secondary tasks. Similarly to the transitions used to allocate staff to primary dispensing tasks, the delays have been changed to instantaneous delays. The hierarchy from highest to lowest is as follows: advanced services, followed by completing secondary dispensing tasks. the justification for this order is the same as that of the primary tasks. Since advanced services are less frequent, they are prioritised.

FIGURE 6.12: Secondary tasks



6.6 Pharmacy Scenarios and analysis

To demonstrate the ability to simulate the effects on performance of different community pharmacy variations, five scenarios were chosen for analysis. Scenarios 1 and 2 investigate the effects of implementing an infallible final accuracy check. Scenarios 3, 4 and 5 have been chosen to demonstrate the effects of changing the make-up of the pharmacy team and the effects of non-dispensing activity. The number of staff in these scenarios is chosen as a representation of small pharmacies in the UK:

1. **Scenario 1**
   Staff - 1 pharmacist, 2 dispensers
   Failures - Probability of Failure in labelling, filling and label application stages.
   No Advanced Services, Lunch hours or Stocking.

2. **Scenario 2**
   Staff - 1 pharmacist, 2 dispensers
   Failures - Probability of Failure in labelling, filling and label application stages, and final accuracy check.
   No Advanced Services, Lunch hours or Stocking.

3. **Scenario 3**
   Staff - 1 pharmacist, 2 dispensers
   Failures - Probability of Failure in labelling, filling, label application and final accuracy check stages.
   Advanced Services - Included.
   Lunch hours - 1 hour off for each dispenser.
   Stocking - Included.

4. **Scenario 4**
   Same as scenario 3, but with 1 pharmacist and 3 dispensers.

5. **Scenario 5**
   Same as scenario 3, but with 2 pharmacists and 2 dispensers.

Different initial conditions of the CPN are used to model each scenario. Figure 6.10 shows the initial conditions of the CPN used in Scenario 3. The initial conditions of other scenarios are created by adding or removing tokens, or altering error rates. Adding additional dispenser tokens to place 6, or pharmacist tokens to place 16, alter the number of staff for Scenarios 4 and 5. The tokens in places 19, 20 and 23 control whether any advanced services, stocking, or lunch breaks are included in the simulation. Additionally, to model an infallible accuracy check in Scenario 1, the probability of 0 is used instead of 0.05 to describe the lack of failures.

6.6.1 Results and their analysis

An object oriented Coloured Petri Net simulation environment was coded in C++, in which the CPN in Figure 6.5 was constructed. A single day of community pharmacy operation was simulated 150,000 times for each scenario. The key results of the performance under the five scenarios are shown in Table 6.8 (note that the sixth and seventh scenarios, 3a, and 3b, are discussed in section 6.6.6). Further results on how members of staff distributed their time between tasks are shown in Tables 6.9 and 6.10.

A convergence test was done on Scenario 3 to test whether 150,000 simulations was sufficient. This was done by running 900,000 simulations of Scenario 3 for comparison with the original results. All results are presented to the level of precision at which the two runs agreed. Scenario 3 was chosen for the convergence test since it has many similarities with all the other scenarios. A further convergence analysis was carried out to investigate how the values of performance indicators changed with the number of simulations. The convergence analysis is presented in Section 6.11.

The values in Table 6.8 are generated by querying place 16, and averaging over the set of simulations. The number of prescriptions completed is counted, and the colours attached to each prescription token are inspected to determine the number of prescriptions that were delayed, the average waiting time, the number of attempts it took to dispense the prescription (near misses), and whether an error was present when the prescription was complete (dispensing error). The number of near misses is recorded by incrementing a near miss colour field on prescription tokens, whenever the pharmacist finds an incorrect prescription. The number of dispensing errors is calculated by the number of prescriptions tokens on place 15. The number of advanced services is equal to the number of basic tokens in place 16.

6.6.2 Scenario variations

Introducing additional non-dispensing demands on the pharmacy increased the waiting time of prescriptions, as can be seen by comparing scenarios 2 and 3.

Table 6.8: Results: Task variations

Scenario	Efficiency					Reliability		
	Deliveries complete %	Total completed %	Advanced services completed %	Delayed %	Waiting time (s)	R	Near Misses	Dispensing errors
1	94.7	96.0	N/a	11.9	554	1	33.6	0
2	95.2	96.4	N/a	11.6	549	0.9923	32.0	1.9
3	78.3	86.3	98.3	17.6	621	0.9923	29.9	1.7
3a	79.6	87.1	98.3	15.9	600	0.9992	3.2	0.2
3b	91.8	94.4	98.2	12.2	552	0.9990	4.6	0.2
4	99.9	99.2	98.4	10.4	518	0.9923	32.2	1.9
5	100.0	99.3	98.4	5.6	443	0.9923	32.2	1.9

Due to the priority given to walk-in prescriptions over deliveries, there is a corresponding decrease in the number of delivery prescriptions completed, as additional non-dispensing demands are considered.

A decrease in the average waiting time was seen when increasing the number of staff. Having three dispensers instead of 2 (scenario 4) reduced the average waiting time for walk-in prescriptions by 103 seconds. A higher still decrease in the average waiting time was gained by adding a second pharmacist to the team (scenario 5). This reduced the average waiting time for walk-in prescriptions by 178 seconds. Notably, in both scenarios where 4 staff were used (scenarios 4 and 5), the pharmacy was finishing a very high percentage of deliveries, over 99% in both scenarios. This suggests that 4 staff working in the pharmacy may have been able to complete a higher number of delivery prescriptions in the allotted time, increasing pharmacy efficiency. This hypothesis is backed up by Tables 6.9 and 6.10, where it can be seen that the staff were idle considerably longer in scenarios 4 and 5.

The difference in the average dispense time between scenarios 1 and 2 is of note. The results show that including the potential for the final accuracy check to fail, reduced the mean waiting time by 5 seconds. This is expected, since a prescription containing an error will always be identified if the final check is infallible (scenario 1), and thus it will be sent to be dispensed again and, therefore, take longer. In scenario 2 a small number of erroneous prescriptions which are not dispensed again (due to them not being spotted at the final accuracy check) make the process faster, but at the cost of dispensing errors reaching patients. This result makes a strong case that, if it was possible to have an infallible final accuracy check, the cost in terms of increased average dispensing times would be low.

The results of the model indicate that the pharmacy dispensing process had a reliability of 0.992, i.e. 99.2% of prescriptions were dispensed correctly. Hence,

the simulations produced a dispensing error rate within the reported range seen in Table 3.1. On the other hand, the frequency of near misses generated during simulations is considerably higher than those reported in previous studies. The discrepancy may relate to the fact that error probabilities in Table 6.3 for dispensing stages may have been set too high, studies may be underestimating the true rate of near misses, or the lack of specifics in the literature make it unsuitable for the validation of the proposed method. Studies on the rate of near misses occurring in community pharmacies are often based on self-report methodologies, which are known to under-report error rates (Flynn et al., 2002a).

6.6.3 Work sampling comparison

Tables 6.9 and 6.10 show the percentage of time that was spent on each task for dispensers and for pharmacists respectively. Work sampling studies have been conducted in community pharmacies, with the objective of describing how pharmacists spend their time (McCann et al., 2009) (Davies et al., 2014) (Emmerton and Jefferson, 1996) (Gregório et al., 2017). These studies are used to collect observations about a subject working in practice, recording their activity at either fixed or randomly timed intervals. If the set of observations is large enough, then the proportion of each recorded activity can be approximated to the overall time spent on that activity. It should be noted that different studies have returned a range of results for how pharmacists spend their time. For example, considering the time pharmacists spend resting, the four studies cited above returned results between 5.8% - 18.1%. Such difference may be attributable to the different work patterns, or variable levels of staff present. Work patterns here, refer to the way tasks are divided up within the pharmacy. In some pharmacies, the pharmacist may not need to contribute to primary tasks, such as stock management or receiving prescriptions, whereas in others, they may have to.

Table 6.9: Dispenser work sample

Scenario	Lunch (%)	Idle (%)	Primary (%)	Receiving (%)	Stocking (%)
1	0	15.6	78.0	6.4	0
2	0	15.9	77.7	6.4	0
3	12.5	13.5	67.1	6.1	0.8
4	12.5	25.3	57.0	4.7	0.6
5	12.5	26.0	55.8	5.1	0.6

For comparison, Davies et al. (Davies et al., 2014) work sample is used as a benchmark for the model, as one of the more recent community pharmacy work sampling studies. The results of the model compared well with the work sample study on a number of fields, the time pharmacists spent on patient counselling and advanced services were within 1% in multiple scenarios. The largest discrepancy was the time spent accuracy checking. In the model pharmacists spend almost 3 times the amount of time checking than reported in the work sample study. This may be because the model omits some activities. Davies et al.'s work sample study has a number of activities which are not included in the model, such as housekeeping, staff training, and sales transactions. Tables 6.9 and 6.10 both show that increasing the number of staff gives staff much more idle time.

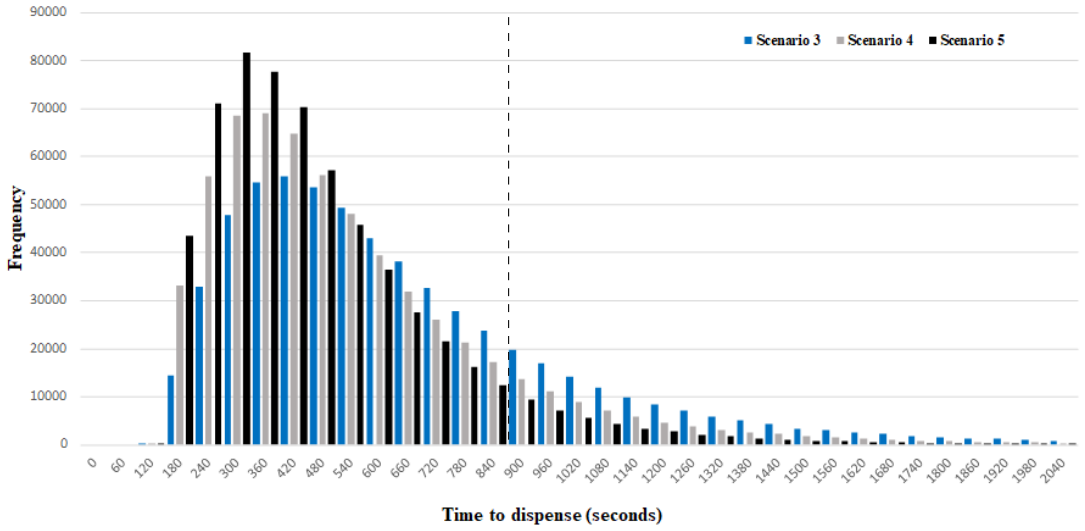
Table 6.10: Pharmacist work sample comparison

Scenario	Idle (%)	Primary (%)	Receiving (%)	Advanced (%)	Checking (%)	Counselling (%)	Storing (%)	Stocking (%)
1	15.6	27.5	2.8	0	32.1	12.9	9.1	0
2	15.8	27.4	2.7	0	31.9	12.9	9.2	0
3	13.9	30.4	3.4	2.8	28.6	12.9	7.6	0.4
4	22.9	17.1	1.6	2.8	32.9	13.0	9.6	0.2
5	29.2	39.3	2.7	1.4	16.4	6.0	4.5	0.4
Work sample	8.6	25.2	N/a	3.2	10.6	12.4	N/a	3.4

6.6.4 Distribution of waiting times

Figure 6.13 shows the distributions of the waiting times for walk-in prescriptions in scenarios 3, 4 and 5. Only these scenarios were considered since they were sufficiently different from scenarios 1 and 2, where no non-dispensing tasks are being completed. The waiting times for each prescription were recorded during a separate run of 6000 simulations, with each scenario totalling just over 600,000 prescriptions dispensed. A prescription was considered to have been delayed if it took more than 15 minutes to be dispensed to a customer, denoted by the dashed line in Figure 6.13.

FIGURE 6.13: A histogram showing the distributions of the time taken to dispense walk-in prescriptions for scenarios 3-5



6.6.5 Causes of delays

There are a number of potential causes for delays. These include: a prescription containing a large number of items being likely to take longer due to the way multiple item prescriptions are handled in the pharmacy, a prescription having to be repeated a number of times due to near misses, or pharmacists being busy with non-dispensing tasks.

Table 6.11 shows the average properties of prescriptions completed in different time periods, for scenarios 3, 4 and 5. It can be seen across scenarios, that prescriptions being completed in increasingly delayed time periods have, on average, a larger number of items and require more iterations of dispensing to complete. As larger teams dispense faster, the average number of items in the same band of duration increases.

TABLE 6.11: A table showing properties of prescriptions completed within fixed time periods t

Scenario		t <15	15 ≤ t<20	20 ≤ t<25	25 ≤ t<30	30 ≤ t<35	35 ≤ t<40	40 ≤ t
3	Avg items	2.40	4.34	4.85	5.13	5.41	5.83	6.59
	Avg itterations	0.086	0.299	0.483	0.686	0.906	1.08	1.45
4	Avg items	2.48	5.30	6.07	6.58	7.05	7.95	9.63
	Avg itterations	0.103	0.434	0.693	0.947	1.22	1.45	1.72
5	Avg items	2.55	6.58	7.64	7.72	8.69	9.72	11.57
	Avg itterations	0.117	0.581	0.918	1.27	1.48	1.67	2.01

6.6.6 Intermediate check

Two additional pharmacy scenarios were included in the analysis to test the effect of implementing an intermediate accuracy check (NPSA, 2007) (Waterfield, 2008) in the dispensing process, in addition to the final check. Two checking methods were included since the mechanics of how staff respond to finding an error during the intermediate check, may also affect performance. In this analysis two responses to finding an error at the intermediate check were proposed.

The first check operated in a similar manner to the final accuracy check, but by checking prescriptions for errors while the labels were being applied, i.e. before the prescription is passed to the pharmacist for the final check, and returning them to be dispensed again if an error was found. The error probability of the check was assumed to be 0.1. The results for this scenario are shown in Table 6.8, the row 3a.

Introducing the first intermediate check at this stage of the process had relatively small effects on the pharmacies efficiency. It reduced the waiting time by 21 seconds, increased the percentage of prescriptions dispensed by 0.8%, and reduced the percentage of delayed prescriptions by 1.7%. This could be due to the fact that when using an additional intermediate check the errors are identified earlier and therefore the prescription can be rectified and completed faster. However, it had comparatively larger effects on process reliability. The additional check increased reliability by 0.069, which in day to day terms meant 29 fewer near misses, and 1.7 fewer dispensing errors per day. Overall, introduction of an intermediate check is a good option for increasing process reliability and efficiency.

The second intermediate accuracy check uses a different protocol. If an erroneous prescription is being checked and the error is spotted, the member of staff rectifies the error there and then, before passing the prescription onto the next stage of the process. However, a chance of failure is included while staff are attempting to fix errors. If they fail to fix an error, the prescription is left with the error as it is passed on to the pharmacist. The same error rates in Table 6.3 are used to control how often staff fail to repair prescriptions correctly during the intermediate accuracy check.

This second accuracy check (Scenario 3b), improved the efficiency of the process more so than the other check. Using this second intermediate check improved

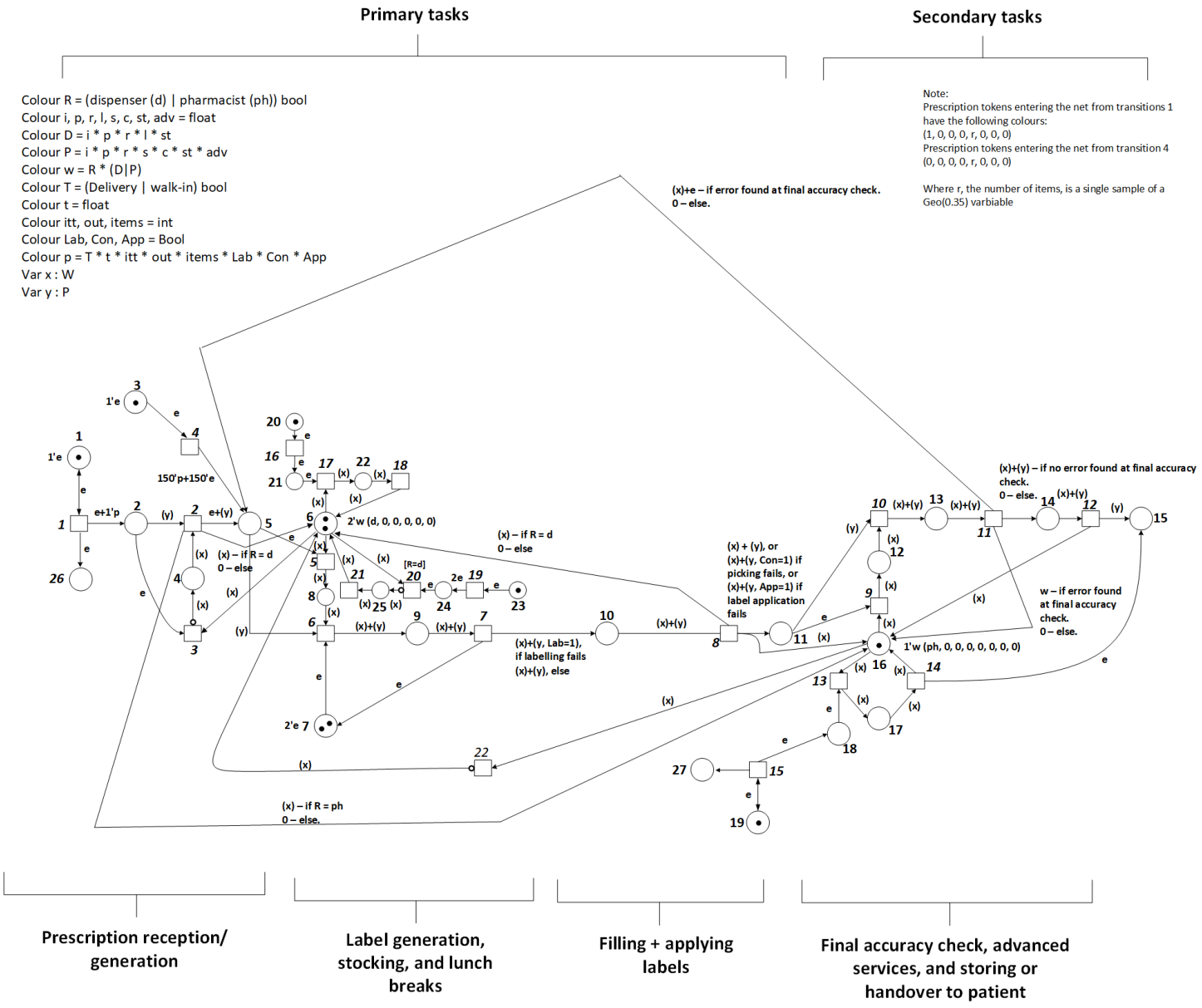
the average waiting time of the pharmacy by 69 seconds, as opposed to using no intermediate check. Furthermore, it increased the number of deliveries completed by 13.5%, and the total completed by 8.1%. As with the previous intermediate check, it had a very positive effect on the reliability of the process, although slightly less than the previous checking strategy. The reliability of the process was improved by 0.067, which translated to 25 fewer near misses, and 1.7 fewer dispensing error per day.

6.7 An alternative representation of the CPN model

Figure 6.5 uses a non-standard formalism for presenting a CPN. The figure omits a number of arc inscriptions, bindings descriptions, and colour set lists, in favour of a more intuitive formalism. In the CPN, token colours are mostly maintained through the firing of transitions. This is in opposition to one of the cornerstones of PN modelling, that tokens are not conserved through the firing of transitions (Schneeweiss, 1999). However, in this CPN model it is mostly the case that the tokens used to enable transitions are the same ones placed into output places. Furthermore, colours are also mostly maintained throughout the firing of transitions.

There are 2 cases where colours are not maintained: when errors are introduced or removed from prescriptions, or when the time between transitions firing is added to colours on tokens. After each transition fires, the delay between the last transition firing and the current one is incremented onto each prescription and worker token. Each increment of time is added onto prescription tokens 'waiting time' colour, and the same increment of time is added to a colour on each worker token corresponding to the task they are currently completing. This property of mostly maintaining properties between transitions firing, is what allows this less strict formalism. However, since it is possible to draw the CPN using the formalism presented in (Jensen, 1996), an alternative CPN diagram can be drawn which follows the formalism. This can be seen in Figure 6.14.

FIGURE 6.14: A coloured Petri net for modelling community pharmacy dispensing



6.7.1 A prescription journey

This section will outline how a single walk-in prescription enters and moves through the net using Figure 6.14, as a means of explaining the net. The chain of transition firings will be explained in consecutive order. The journey begins when a patient arrives at the pharmacy. This event is described by the firing of transition 1. After transition 1 has fired, the patient must wait for a staff resource to become available to receive the prescriptions.

Either there will already be a staff resource available when the prescription arrives, or they will have to wait until a member of staff becomes available in place 6. For this walk-through, assume that a staff member is available on place 6. Then transition 3 will fire, moving the staff resource to the counter. Note how transition 3 requires a basic token from place 2 to be enabled, this stops staff resources being allocated to the counter if there are no customers waiting. The idea of basic tokens controlling staff resource allocation will become a common theme.

Once at the counter, the delay of transition 2 controls how long it takes to receive the prescription from the customer. After the prescription has been received, it is placed into place 5. If there are any prescriptions on place 5 already, it will be placed at the back of the queue of prescriptions with which it shares a type. In this example, the walk-in prescription would be placed behind the other walk-ins, but above any deliveries due to the priority given to walk-ins.

Once the prescription reaches the front of the queue on place 5, and when the next staff resource becomes available on place 6, the memeber of staff will begin to dispense the prescription. Transition 5 allocates a member of staff to complete the primary dispensing for this prescription. Transition 6 models the member of staff choosing the prescription to work on, and after this has fired, both the prescription token, and the staff resource token are then placed onto place 9. The two tokens will then make their way around the net together.

Transition 7 represents the labels being generated, after which, the prescription is placed onto place 10. The time spent here models how long it takes someone to pick the contents, and apply labels for a prescription. After these are completed, the prescription token is separated from the staff resource token. If the staff token was a dispenser, the staff token is placed onto place 6, or if it was a pharmacist,

the staff token is placed onto place 16. The prescription token is placed onto place 11, where it waits to be checked by a pharmacist.

In the queue on place 11, prescriptions are only removed when pharmacists become available to check them for accuracy. Transition 9 allocates an available pharmacist to check a prescription. Transition 10 models a pharmacist choosing which prescription to check. After transition 10 has fired, both tokens are placed into place 13. Transition 11 models the accuracy check. Here the pharmacist accuracy checks the prescription, and if the prescription is correct then both tokens move to place 14.

If, however, there is an error present on the prescription token, picked up from one of the earlier stages of the process, then there is a chance that the pharmacist will spot the error. If the error is spotted, the prescription is returned to place 5 to be dispensed again. If, however, the pharmacist fails to spot the error, both tokens continue onwards to place 14. Similarly, if there is no error present, both tokens continue to place 14 together.

Transition 12 represents the end of the dispensing process. This transition models the prescription being handed out to a patient, or being stored to be picked up later. After this transition has fired, the prescription is placed into place 15. It remains here until the end of the simulation.

6.8 Incorporating in-field data into the CPN: Version 4

Chapter 5 shows the process of data collection and analysis from the study of 4 UK community pharmacies. As presented, the quantitative data was analysed using a distribution fitting analysis, and for 6 key stages of the dispensing process, best fitting probability distributions for the time taken to complete each stage were calculated. To demonstrate the effect of using the real data to control transition timings, a further set of simulations were run using some of the real data. To do this further simulations were run using Version 4 of the CPN, with 5 of the fully developed scenarios used previously in this Chapter (Scenarios 3, 3a, 3b, 4 and 5 from Table 6.8).

Processor transitions in Version 4 of the CPN had to be altered slightly to accommodate the distributions from the on-site studies. Previously, in Versions 2 and 3, multiple item prescriptions had re-sampled the same distribution once for each item in a prescription, taking the sum to be the total time to complete a stage. In Version 4, for each processor stage of the process, the data collection has produced a set of 6 distributions, which may need to be sampled from, depending on the number of items in a prescription. Version 4 of the CPN was altered so that processor transitions could accommodate up to 6 distributions. Which distribution is used to generate a delay for processor transitions, then became dependent on the number of items in a prescription.

Table 6.12 shows results of Versions 3 and 4, where results have been derived using the same pharmacy scenarios, i.e. same number of staff and checking strategies, using different duration distributions. Distribution parameters in Version 3 were guided by expert opinion, and those in Version 4 were derived from fitting real quantitative timings of the process.

TABLE 6.12: Comparing performance of pharmacies using assumed and real data

	Scenario	Efficiency					Reliability		
		Deliveries complete %	Total completed %	Advanced services completed %	Delayed %	Waiting time (s)	R	Near Misses	Dispensing errors
Expert opinion	3	78.3	86.3	98.3	17.6	621	0.9923	29.9	1.7
	3a	79.6	87.1	98.3	15.9	600	0.9992	3.2	0.2
	3b	91.8	94.4	98.2	12.2	552	0.9990	4.6	0.2
	4	99.9	99.2	98.4	10.4	518	0.9923	32.2	1.9
	5	100.0	99.3	98.4	5.6	443	0.9923	32.2	1.9
In-field Data	3	73.2	83.4	98.4	8.0	534.5	0.9926	31.1	1.6
	3a	81.5	88.4	98.0	6.1	499.3	0.9993	3.2	0.2
	3b	84.6	90.3	98.3	5.1	469.9	0.9990	4.4	0.2
	4	100.0	99.5	98.8	1.8	362.4	0.9926	37.0	1.9
	5	81.7	88.6	98.1	4.5	447.6	0.9926	33.0	1.6

The results clearly show that when transitions informed by expert opinion were replaced by those derived from in-field data, the average waiting times produced by the model were reduced in all scenarios. Furthermore, far fewer prescriptions were delayed when the real data was being used. A notable difference between the two sets of results is the variable performance of different set-ups. When expert opinion informed distributions were being used, Scenario 5 appears to outperform Scenario 4. This order changes when in-field data is used to control the firing of transitions. It may be that the expert opinions were overestimating the time taken to complete secondary (pharmacist only) tasks.

6.9 Code runtime comparison

Before developing Version 3 of the CPN, it was proposed that the changes made may improve the efficiency of the computer simulations, and enable for faster simulation times. While the code was being updated with the changes made to reduce the number of objects required to build the CPN, some functions were updated to what were thought to be more efficient versions. To test whether these changes had improved the code's efficiency, Scenario 3 was chosen to use in the investigation. Three Versions (2, 3, and 4) of the CPN simulated Scenario 3 a number of times. Table 6.13 below shows the results of how long each set of simulations took using each version of the CPN model.

Table 6.13: Time taken to complete simulations

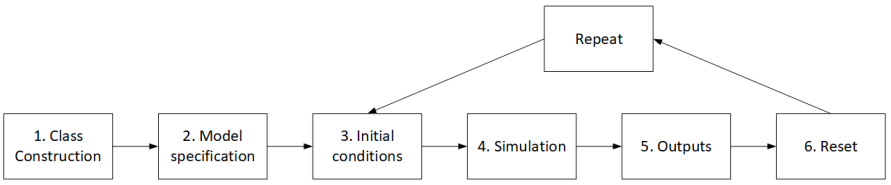
Simulations	Version 2 (s)	Version 3 (s)	Version 4 (s)
1	0.872	0.52	0.59
10	5.570	3.71	3.43
100	54.36	33.30	30.77
1000	652.09	329.50	308.72
100 000	56597.49	33602.48	30971.15
150 000	83414.67	49767.21	47071.97

It can be seen that the reduction of complexity between Versions 2 and 3 has indeed reduced the run time of the code over a range of values for the number of simulations. Similarly Version 4 had a slightly reduced runtime when compared with Version 3. This is not thought to be because of the inclusion of real timing data into the model, but due to other efficiency improvements within the code. One major change that may have contributed was swapping out a 'switch' (a C++ coding object) with 83 cases for a set of 3 functions, which were used to perform the same actions within the code.

6.10 Code implementation

This section will outline how the CPN models developed throughout this chapter were implemented in C++. The code was developed by the author of this thesis. Figure 6.15 shows a flow diagram of how the CPN simulation code was built and was used to simulate different community pharmacy set-ups.

FIGURE 6.15: A flow diagram of how the CPN simulation model was coded



Classes were built for each of the basic elements of Petri Nets: places, transitions, arcs, and tokens (see Appendix D for examples of code). In Version 1, tokens were denoted by integer values attached to places. However, when colours were introduced to the net in Version 2, there became a need for a tokens class. The place class contains a vector of tokens which represent which tokens are on each place.

Then using Figure 6.5, the specifics of how objects in the CPN model are linked together were coded into the model, which corresponds to the stage of model specification in Figure 6.15. This includes: indicating how arcs connect places and transitions, defining behaviours for accuracy checks, and defining which colours are on tokens and how the colour values should be changed throughout the simulation.

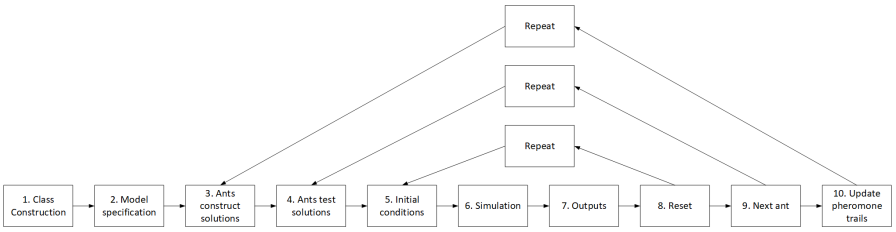
Before starting to simulate the CPN for a given set-up, the initial conditions must correspond to the set-up. This involves setting the number of dispensers and pharmacists, and making sure there are no other tokens on the net beyond those seen in Figure 6.5.

The simulation is run by firing transitions consecutively as they become enabled. An internal simulation clock is maintained, and the simulation ends when the simulation clock indicates that the time is past 5pm. Once 5pm has been reached, no more transitions are fired. The ending state of the Petri net is then written as a line of an Excel spreadsheet. An example of how this is done is that, the number of tokens on place 36 in Figure 6.5 at the end of a day simulation indicates how many prescriptions were completed during the day.

The net is then reset back to the initial condition. This involves removing all the prescription tokens that have entered the net, and moving all staff tokens back to their starting positions. All the values of colours attached to staff tokens are also reset to zero at this point. If a different pharmacy set-up needs to simulated, then the initial conditions can be changed at this point.

Figure 6.16 below shows a flow diagram of how the CPN simulations were combined with the Ant Colony Optimisation (presented in Chapter 7) to find optimal community pharmacy set-ups.

FIGURE 6.16: A flow diagram of how the CPN simulation model was coded



The code has the basic three stage structure of: ants construct solutions, each ant tests the solution it generated using the CPN to evaluate the set-up, and finally the best ants set-up is used to lay down a pheromone trail onto the search space. The optimisation is terminated after one of two conditions are met, either a maximum number of iterations is reached, or if the same iteration best path is returned on 3 consecutive occasions.

6.11 Convergence analysis

This chapter has demonstrated how a CPN can be used to model a community pharmacy and generate 8 performance indicators for the pharmacy. This section will analyse the convergence of each of the performance indicators using a set of convergence graphs. Each graph plots a running average of the performance indicator value on the y-axis, as the number of simulations increases with the x-axis. Figures 6.17 - 6.23 show how the value of 7 performance indicators generated by the community pharmacy CPN converge to a value as the number of simulations increases.

The analysis of different set-ups in this Chapter used 150,000 simulations per scenario. In the ACO optimisation framework used in Chapter 7, there was a larger number of set-ups to consider. Therefore, it was necessary to reduce the number of simulations used to test each set-up. Ants used 500 simulations to initially explore the viability of a set-up, and then used 1000 simulations to verify the set-ups performance. It should be noted that where ants tested the same set-up more than once, the results were combined and averaged. Thus the minimum number of simulations used to evaluate a set-up would be 1000. Although it

should be noted that many solutions were tested more than once, and therefore 2000 or more simulations were used. It can be seen from the convergence graphs above, that 2000 simulations appears to be the point at which all the performance indicators have converged.

FIGURE 6.17: Convergence analysis: total completed

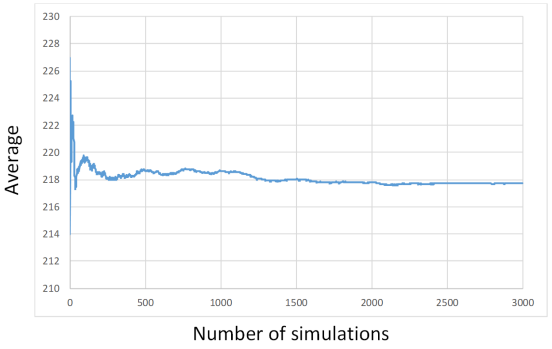


FIGURE 6.18: Convergence analysis: deliveries completed

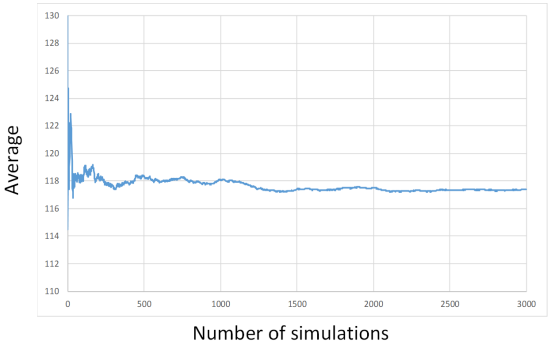


FIGURE 6.19: Convergence analysis: waiting time

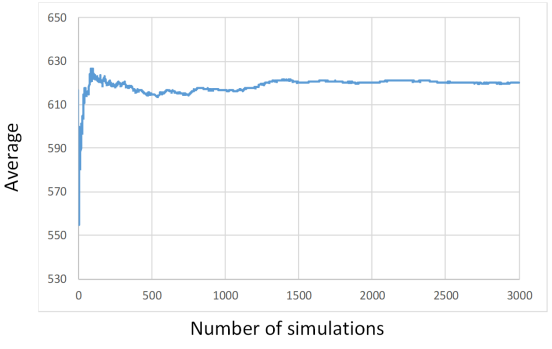


FIGURE 6.20: Convergence analysis: near misses

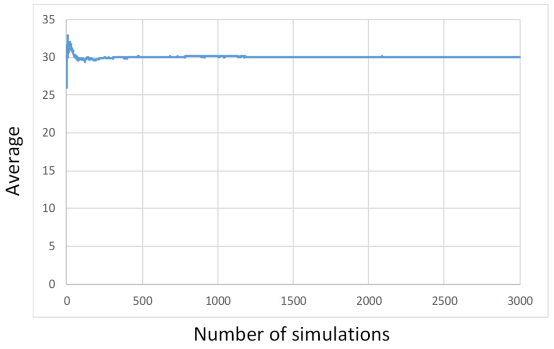


FIGURE 6.21: Convergence analysis: dispensing errors

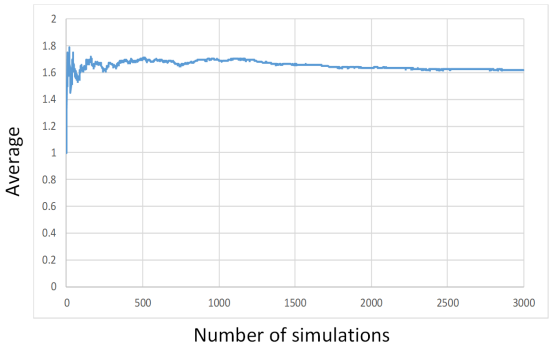


FIGURE 6.22: Convergence analysis: delayed

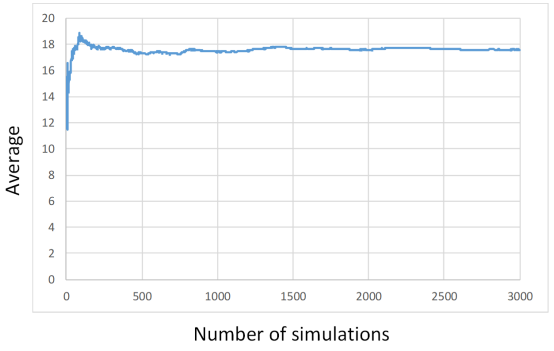
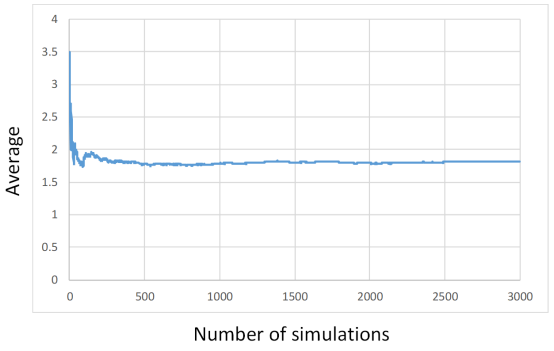


FIGURE 6.23: Convergence analysis: advanced services completed



6.12 Conclusion

This chapter has described the development of a model for simulating the community pharmacy dispensing process. An initial non-coloured net was proposed, which was further developed into a Coloured net because of the need to incorporate failures, and include additional token information to be recorded during simulations. The initial coloured Petri net was introduced, and a further iteration of the CPN, which made use of self-concurrent transitions and instantaneous delays with priority parameters was presented. Detailed results of the final CPN iteration were generated, and a comparison of the run time between the two models was carried out. Finally, an alternative representation of the CPN was shown in 6.14. CPN has been shown to be a suitable model for evaluating multiple aspects of community pharmacies performance within a single modelling framework. Additionally, an outline of how the CPN code was implemented, and an analysis of the convergence of performance indicators were given. Given the model produced it this Chapter, a key question presents itself: what are the best community pharmacy set-ups?

Chapter 7 will present the findings of applying an optimisation framework to the community pharmacy set-up model. An Ant Colony Optimisation technique is used to find the best set-ups.

Chapter 7

Application of Ant Colony Optimisation to the community pharmacy dispensing process modelled using the CPN

7.1 Introduction

This chapter outlines the approach taken to optimise the Coloured Petri net model developed in Chapter 6. A description of the basic methodology and the biological basis of the technique is given first. This is followed by an investigation of a small example application to the community pharmacy CPN seen in Chapter 6, which is developed further throughout the Chapter as increasingly more variables are taken into account. Optimisation of the CPN model is needed since decision makers are faced with conflicting objectives, such as reducing staff costs, providing a fast and efficient service, and reducing errors.

7.2 Optimisation techniques: a review

This section represents a review of the set of optimisation techniques which could have been used to optimise community pharmacy set-ups. It will outline how

it was decided that ACO (Ant Colony Optimisation) was the best optimisation technique for the problem at the time of developing this work.

The pharmacy model built in Chapter 6 presents an initial set-up problem related to the best possible initial conditions of the model. In Chapter 6, the model may be set up to have a number of dispensers, a number of pharmacists, and one of three checking strategies. If each set-up is assigned a performance score based on the pharmacies performance, then choosing the best values for these three decision variables represents an optimisation problem. The problem could be written as a mathematical optimisation, as shown in Equation 7.1:

$$\text{Choose } x \text{ to minimise } U, \text{ where } U = f(x) \tag{7.1}$$

where x is a 3-tuple of positive integers, U is a utility value assigned to pharmacy set-ups using performance indicators generated through CPN simulation of a pharmacy with initial condition x, and f is the transformation that is used to transform the set of intputs x into a set of performance indicators using the CPN model. The first question to answer when deciding which optimisation technique to use is, what type of optimisation problem does the community pharmacy set-up problem represent?

The problem involves choosing, in this instance three variables, which may only take discrete values (a set-up may not use 2.5 dispensers, or checking strategy 1.5). Hence, the optimisation is a discrete multivariate problem where the function (in this case, the CPN) does not have any analytical form. This rules out any techniques which require an analytical form of the function f. This includes ruling out all linear and non-linear programming optimisation methods, and all gradient based methods such as the Newton-Raphson method.

Since there is no analytical form of the function f, the choice of optimisation techniques is restricted to metaheuristic methods that do not rely on the existence of derivatives or gradient functions. There are a number of metaheuristic techniques which could be used to tackle the problem. Candidate metaheuristic algorithms included: the tabu search, particle swarm optimisation, ant colony optimisation, simulated annealing, and genetic algorithms.

Some studies have attempted to compare the performance of metaheuristic algorithms on a fixed set of optimisation problems. For example, Ali et al. (Ali et al.,

2005) attempted to compare the performance of 5 algorithms on a set of 50 test problems. The 5 algorithms were: improving hit and run, hide and seek, coded random search, and genetic algorithms. All 5 algorithms were tested on a set of 50 problems, where the number of objective evaluations was varied. To compare performance, 4 measures were used: general applicability, efficiency, trustworthiness and ease of use. Their work showed that the best choice of algorithm can be dependent on the maximum number of iterations the user is prepared to computer, with genetic algorithms performing well when the number is large, and improving hit and run and hide seek algorithms performing well with fewer iterations.

Ezugwu et al. carried out a similar study on a wider range of algorithms (Ezugwu et al., 2019). They tested 12 popular algorithms on a set of 50 optimisation problems, using a similar set of evaluation metrics to Ali et al. Their results showed that 6 of the twelve algorithms tested performed much better than the other 6. The relatively higher performing algorithms were: DE (differential evolution), SOS (symbiotic organism search), PSO ()particle swarm optimisation), GA (genetic algorithm), (CS) cuckoo search, and (ACO) ant colony optimisation. In addition, it was also observed that the quality of solutions produced by each of DE, SOS, CS, and ACO algorithms reliably improved as the number of objective evaluations allowed per problem increased.

Example applications of metaheuristic techniques within the healthcare sector have included using simulated annealing (Baesler et al., 2015), the tabu search (Niu et al., 2013), or genetic algorithms (Yeh and Lin, 2007) to optimise hospital units, and ACO algorithms have been used to optimise diabetes screening policies (Brailsford et al., 2007) and emergency department efficiency (Fruggiero et al., 2008). Meta-heuristic techniques also see application in a wide range of non-medical fields. Examples have included using ACO to predict appropriate credit ratings for businesses (Martens et al., 2010), and the examination timetabling problem (Dowsland and Thompson, 2005).

The literature suggests that a number of optimisation techniques could be feasible solutions for solving the pharmacy set-up optimisation problem, and further work is required to definitively asses which algorithms are most applicable for any given situation. However, ACO has shown a near-optimal performance on travelling salesman problems (a classic discrete optimisation problem) (Selvi and Umarani, 2010), and Xin-She recommends ACO as the best metaheuristic method for dealing with discrete optimisation problems (Xin-She, 2007). It was on this basis that

ACO was selected as the optimisation method to be used in this thesis, since the pharmacy optimisation problem is clearly a discrete optimisation problem.

7.3 Ant Colony Optimisation

Ant Colony Optimisation uses artificial ants, which mimic the behaviour of real ants leaving their nest to search for food, in an attempt to find the shortest route to a food source. In the metaheuristic, artificial ants search the state space of an optimisation problem, where ants that find good solutions to the problem are able to influence the future choices of other artificial ants.

7.3.1 Biological Ant Behaviour

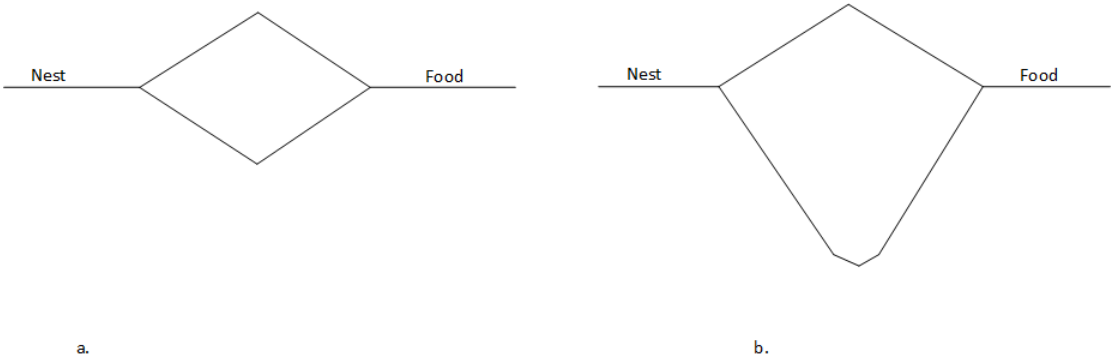
Ants are social insects by nature, and they live together in large colonies which can be home to up to 25 million ants. While searching for food, each ant can communicate with other members of the colony by secreting a pheromone trail of chemicals. As well as secreting this pheromone, every ant can also detect the trails of pheromone left by other foraging ants (Xin-She, 2007). By following the trails left by other members of the colony, ants can find their way to food sources which have already been scouted by other ants. The methodology the ants use, a collective path marking/path following behaviour, is the basis of Ant Colony Optimisation (Dorigo and Stutzle, 2004).

When ants locate a source of food, they will secrete pheromone onto the food source, and then leave a trail of pheromone marking the path to and from the food, back to the nest. As more ants journey to visit the food source, a particular path (often the most efficient, or shortest) will come to be used by the majority of the ants due to a higher pheromone concentration. This behaviour encourages individual ants to follow paths with higher concentration of pheromones, rather than a top down instruction from the queen of the nest.

7.3.2 The Double Bridge Experiment

Laboratory experiments have been conducted by Goss et al. (Goss et al., 1989) (Deneubourg et al., 1990) demonstrating how colonies of Argentinian ants (*Iridomyrmex humilis*) are capable of reliably finding the shortest path between their nest and a food source. Experiments were ran with an ant colony separated from a food source by two bridges. Initially, the bridges are of equal length, and they are later changed so that the two branches have different lengths. The two configurations are shown in Figure 7.1.

FIGURE 7.1: Branch configuration for the double bridge ant colony experiment



In Figure 7.1 a., the branches are the same length, and in Figure 7.1 b., one branch is longer than the other. The experiment was run as follows. Ants were freely allowed to move between the nest and the food while the proportion of ants choosing each branch was recorded. The results showed that when the branches were equal, initially the ants would start using one of the branches more than the other, and later on after the ants had been left making the journeys back and forth for some time, the ants converge to all using the same branch.

This is explained by the fact that initially there are no pheromone trails on either branch, so the ants choose freely between the two. However, due to the randomness of the process, one branch will be chosen by slightly more ants than the other. This small difference causes the next set of ants to approach the bridge to be more likely to choose the already slightly more travelled branch. Over time, this *positive feedback* leads to the majority of the ants choosing the same path to the food. However, a small percentage of ants do still continue to choose the less popular branch.

In the unbalanced experiment the bridges are of differing length. Their branch configuration used a ratio of 2, such that the longer branch was twice the length

of the shorter one. With this ratio between the two branches, the ants chose to use the shorter path in the majority of cases. This experiment begins in a similar manner to the first, with both ants choosing either branch equally. However, the shorter branch makes an impact on ants returning from food to the nest. Ants which initially chose to follow the short branch arrive at the food first, leaving a marking of pheromone on the shorter branch. When they set off on the return trip, the shorter path they took out to the food is already marked with some pheromone, whereas the longer branch is not. Thus they are more likely to choose the shorter branch to return back to the nest; again increasing the amount of pheromone on the shorter branch.

FIGURE 7.2: Configuration for the short cut ant colony experiment

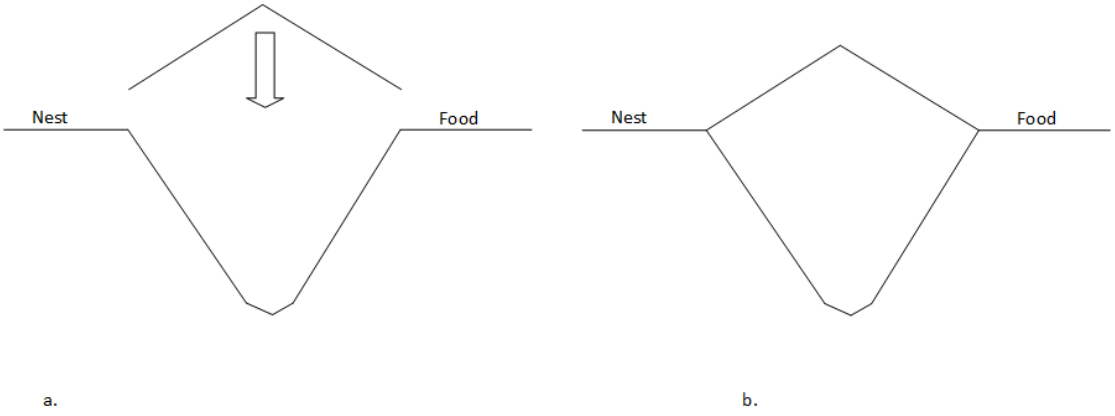


Figure 7.2 shows the configuration used to test the ant colony's response to being offered a shorter route after having already converged to a path. After 30 minutes, the shorter bridge to the food is introduced. The new shorter route is only chosen intermittently, and in most cases the majority of ants still continue to use the sub-optimal route. This is due to the already high concentration of pheromones on the sub-optimal path, combined with a slow evaporation rate. If the pheromone evaporated faster, this would bias the ants to be more likely to search newer paths and potentially forget suboptimal solutions.

7.3.3 Summary

This subsection has introduced the biological basis for Ant Colony Optimisation algorithms. The double bridge experiment was introduced to show how ants can use pheromone trails to find the shortest path to food sources, and an experiment

which introduces a new path to the ant's environment showed how using only the biological mechanism to find shortest paths is not optimal.

7.4 Simple-Ant Colony Optimisation

The set of bridge experiments shows that ants do have a tendency to optimise for the shortest route, however the short cut bridge experiment shows that using artificial ants which exactly match the behaviour of biological ants would be unsuitable for complex problems. Fortunately, the behaviour of biological ants can be improved upon in a number of ways when using artificial ants (Dorigo and Stutzle, 2004). They can be given:

1. An ability to remember the path they have taken from their nest to the food, (rather than relying on leaving a pheromone trail while on the outward journey).
2. Probabilistic path generation, which is entirely controlled by pheromone concentration on paths.
3. Delayed pheromone marking, such that pheromone is only placed on paths when ants return back to the nest.
4. Paths (or solutions) are evaluated when an ant reaches the food, such that the amount of pheromone placed on the ants path on the return journey is proportional to the quality of the solution.
5. The evaporation rate of pheromone can be set at any level desired, ideally to increase ants propensity for looking for new solutions while not forgetting high quality ones.
6. Their behaviour can be coordinated, rather than be identical for each ant. An example could be letting a number of ants find their way from nest to food. Once at the food they evaluate each of their paths, but only let the ant with the best solution place pheromone onto the path it took.

Dorigo et al. (Dorigo and Stutzle, 2004) introduce the concept of simple Ant Colony Optimisation (S-ACO), an algorithm used to solve minimum path problems on graphs. S-ACO is a starting point to understand how ACO functions, on top

of which other more advanced and efficient implementations of the ant colony optimisation algorithm can be understood. The concepts used in S-ACO will be introduced next, and later used as a basis to explain the more advanced min-max ACO.

7.4.1 Solution construction in S-ACO

Ants in the S-ACO algorithm have two distinct modes of operation, one for when they are leaving the nest to travel to the food source (forward), and another when they are returning to the nest (backward). While in forward mode, the ant travels towards the food source, and upon reaching the food, it will switch to backward mode and head back towards the nest. One restriction that the algorithm has is that the state space is discrete, and can be represented as a graph of nodes, with edges indicating valid paths available to the ants.

While in forward mode, ants construct solutions on the graph by repeatedly applying a decision policy. The policy is probabilistic, and dependent on the amount of pheromone on edges connecting to neighbouring nodes. The algorithm starts with an equal amount of pheromone on all edges. When an ant is on node i, it uses the concentration of pheromone on all edges to neighbouring nodes to probabilistically decide which path to take. If N_i is the set of nodes neighbouring node i, then an ant chooses the next node to visit, j, with probability p_{ij}. If the concentration of pheromone on paths from node i to all neighbouring nodes j are denoted by τ_{ij}, then we have:

$$p_{ij} = \begin{cases} \frac{\tau_{ij}}{\sum_{l \in N_i} \tau_{il}} & \text{if } j \in N_i \\ 0, & \text{if } j \notin N_i \end{cases} \tag{7.2}$$

where the neighbourhood of node i, N_i, is defined as the set of nodes which can be reached from node i by travelling along one edge, except for the node that was previously visited by the ant. Ants move from node to node, using this decision policy to decide where to move next until they reach the food source. Since each ant will take a different route due to the probabilistic nature of the decision rule, the routes will differ in objective value. In the most basic case of finding the shortest path, the objective value will be the length of the path. In the case of optimising a community pharmacy, valid objective values could be, the number of

prescriptions dispensed, the average waiting time, the number of errors made, or a combination of these.

7.4.2 Pheromone deposit

While on its way back to the nest from the food, the kth ant deposits an amount of pheromone on each edge that was in its path, equal to $\triangle\tau_k$. Hence if the kth ant used the edge (i, j) in is its path to the food source, after the ant has placed its pheromone, the pheromone on the edge will increase following Equation 7.3.

$$\tau_{ij} \rightarrow \tau_{ij} + \triangle\tau_k \tag{7.3}$$

Hence if an ant chooses to use an arc (i, j) in its path to the food source, it is more likely that future ants will also choose to use this arc again. It should be noted that in some problems, it may be possible for ants to construct loops in their paths to the food source. It is at this point, before placing pheromone onto the path that all loops should be eliminated from the path. However, for the applications used in this Chapter, the possibility of ants creating loops is eliminated by the way nodes and edges are constructed.

A key question to the pheromone deposit stage of the algorithm is, how much pheromone should be secreted by each ant? A simple solution is to have each ant secrete the some constant amount. Alternatively, ants can deposit an amount of pheromone based on the quality of solution they produced, such that ants which find higher quality solutions are allowed to deposit more pheromone.

7.4.3 Pheromone evaporation

Pheromone evaporation can be thought of as the method by which ants can forgo using the best path they have found so far, and explore new paths. Its main purpose is to avoid situations in which the ants quickly converge to a local optimum. Evaporation of pheromone occurs at a constant rate at each iteration of the algorithm. One way to implement the evaporation is to update the pheromone on all arcs according to Equation 7.4, after all ants have constructed their solutions, but before the pheromone update is placed onto trails.

$$\tau_{ij} \rightarrow \tau_{ij}(1-\rho), \quad \forall(i,j) \in A \tag{7.4}$$

where ρ is a constant parameter such that $\rho \in (0, 1]$, and A is the set of edges. ρ can be thought of as the evaporation rate of the pheromone. A single iteration of S-ACO involves one cycle of the three stages outlined above: ant solution construction, pheromone evaporation, and pheromone secretion. Applying the three stages repeatedly constitutes an ant colony optimisation algorithm. The algorithm is repeated until one of the stopping conditions is met. Stopping conditions can be a maximum number of iterations, the same iteration best path being returned multiple times in a row, or when a number of different solutions to the problem have been tried.

Initially, three Ant System algorithms were developed in the literature, following the basic structure of S-ACO. Two versions, ant density and ant quantity, used ants which laid down pheromones onto the paths between nodes immediately after crossing the arc, arc by arc. The third version, ant cycle, only let ants lay pheromone onto trails after all the ants had constructed and evaluated a solution. The amount of pheromone then laid down by each ant was a function of the solution quality. Ant cycle had superior performance, and this is the common method ACO algorithms use since the two other inferior techniques were abandoned.

7.4.4 Summary

The method by which biological ants find the shortest routes to food sources near their nest has been introduced. This has been followed by a description of Simple-Ant Colony Optimisation, a useful starting point for understanding and developing a more complex ACO algorithm. The main principles and techniques used in S-ACO will be the basis for further development of another more advanced ACO technique in the following section.

7.5 Choosing a framework

Before starting to implement an Ant Colony Optimisation algorithm onto the CPN model, a decision had to be made about the type of algorithm that will be used. Since the first ant colony optimisation algorithm, ant system, was released

a number of ACO algorithms have been seen in the literature. Table 7.1 shows a summary of the main versions of the optimisation algorithm.

TABLE 7.1: ACO algorithms in Chronological order of appearance from Dorigo at al. (Dorigo and Stutzle, 2004)

ACO algorithm variant	Year of appearance
Ant System (AS)	1991 (Dorigo, 1992)
Elitist AS	1991 (Dorigo, 1992)
Ant-Q	1995 (Gambardella and Dorigo, 1995)
Ant Colony System	1997 (Dorigo and Gambardella, 1997)
Max-Min Ant System	1997 (Stültze and Hoos, 1997)
Rank-based Ant System	1997 (Bullnheimer et al., 1997)
Hyper-cube AS	2001 (Blum et al., 2001)

Dorigo et al. (Dorigo and Stutzle, 2004) suggest that out of all the ACO algorithms seen in the literature, max-min ant system, and ant colony system often have the best performance. In a controlled test environment using the travelling salesman problem, Dorigo et al. showed that max-min ant system had the best performance over long run times, compared to the other ACO algorithms. Furthermore, max-min ant system has been shown to be an effective algorithm even when heuristic data is not available. Note that heuristic data, is information known about the problem before the optimisation starts, which influences the ant's decisions about which nodes to visit. Since it was unclear whether there would be any heuristic data when optimising the dispensing process, on this basis it was decided that max-min ant system would be the ACO algorithm used in the proposed approach.

7.5.1 Max-min Ant System (MMAS)

Max-min ant system has 4 main innovations compared to S-ACO. The first, is that rather than letting all ants place pheromone onto the solutions they construct, MMAS allows only ants which found high quality solutions to place pheromone onto the trail. This is implemented by either, only letting the iteration best ant, or the global best ant to lay down pheromone. Using this mechanism alone might lead to ants becoming quickly attached to a suboptimal solution due to the reinforcing nature of ACO algorithms. However, the second changes used in MMAS places limits on possible range of pheromone values. These constrict the value of pheromone trails on paths, such that they are limited to an interval $[\tau_{min}, \tau_{max}]$. Thus, even if pheromone is only laid on the same trail for a number

of iterations, there is a limit to the probability that ants will keep choosing that marked path. The ants will then still have some tendency to explore other routes.

The third change is implemented by initialising all trails with the same amount of pheromone. This amount of pheromones is set to an estimate of the upper limit, τ_{max}. A small pheromone evaporation rate is then used, which again increases the ants tendency to explore many routes. If all routes are marked with the maximum amount of pheromone, the ants don't tend to favour any others until the evaporation has reduced the concentration on paths that are not re-marked by the ants. The final change in MMAS is that pheromone trails can be reinitialised when the system reaches stagnation (i.e. no improved route is found for a number of iterations).

7.5.2 Solution construction in MMAS

The solution construction in MMAS algorithms is similar to that of S-ACO seen previously (See Equation 7.2), although in MMAS the ants are able to take heuristic information about the optimisation problem into account. While in the forward mode, a number of ants simultaneously construct solutions on the graph by repeatedly applying a decision policy. The decision policy is probabilistic, and dependent on two factors: the amount of pheromone on edges connecting to neighbouring nodes, and the inherent desirability of each edge. Thus when an ant is on node i, it uses the concentration of pheromone and some prior heuristic information about the edges, to probabilistically decide which node to move to next. If N_i is the set of nodes neighbouring node i, then an ant chooses the next node to visit, j, with probability p_{ij}. If the concentration of pheromone on paths from node i to all neighbouring nodes j are denoted by τ_{ij}, and the heuristic information about node j in relation to node i is ν_{ij}, then Equation 7.5 describes the decision probability function:

$$p_{ij}(t) = \begin{cases} \frac{[\tau_{ij}]^{\alpha}[\nu_{ij}]^{\beta}}{\sum_{l \in N_i}[\tau_{il}]^{\alpha}[\nu_{il}]^{\beta}} & \text{if } j \in N_i \\ 0, & \text{if } j \notin N_i \end{cases} \tag{7.5}$$

where $p_{ij}(t)$ is the probability that an ant on node i at time t moves to node j as its next node. N_i, the neighbourhood of node i, is defined as the set of nodes which can be reached from node i by travelling along one edge, except for the node

that was previously visited by the ant. The two parameters α and β control the relative importance given to pheromone or heuristic information in the decision policy. Ants move from node to node, using this decision policy to choose where to move next, until they have constructed a solution to the optimisation problem.

7.5.3 Pheromone update and evaporation rate

When each of the ants has built a solution, the pheromone update is done by completing three actions. First each ant evaluates the solution it has constructed by calculating an objective value. Secondly, pheromone is evaporated from all paths according to the evaporation rate of the algorithm, and finally new pheromone is deposited onto *one of* the best paths found by the ants so far.

Pheromone evaporation and deposit in MMAS are carried out using the same equations as in S-ACO (See Equations 7.4 and 7.3), although only one ant is allowed to deposit pheromone. Equation 7.6 describes how pheromone trails are updated by combing the pheromone evaporation and deposit (Stültze and Hoos, 2000):

$$\tau_{ij}(t+1) \rightarrow \tau_{ij}(t)\rho + \triangle\tau_{ij}^{best}, \quad \forall(i,j) \in P^{best} \tag{7.6}$$

where $\triangle\tau_{ij}^{best} = 1/O^{best}$, and O^{best} is best objective value found by an ant, and P^{best} is the path used by the best ant. This O^{best} can either be the 'global best', i.e. the best solution found in all completed iterations, or it can be the 'iteration best', the best objective value found in *only* the current iteration of the algorithm. It is common for MMAS algorithms to alternate between reinforcing the 'global best' and the 'iteration best' paths. Stültze and Hoos (Stültze and Hoos, 2000) indicate that when applying MMAS, mixed strategies that used some global best updates with mostly iteration best updates perform strong optimisations, although the best choice can depend on the size of the problem. A strategy they highlight for large problems involves using global best updates with increasing frequency over time throughout the optimisation.

7.5.4 T-max-T-min

In MMAS, pheromone trails are limited to an interval of values [τ_{min},τ_{max}]. These limits are imposed to stop the ants converging to a single path too quickly, by limiting the possible maximum difference between pheromone concentration on different paths. Using these limits has the following effect on the probabilities of ants path choices. The probabilities of an ant choosing a given path will lie in a interval [p_{min},p_{max}], where, $0 < p_{min} \leq p_{ij} \leq p_{max} < 1$. If an ant has only one choice of node for its next decision, then $p_{min} = p_{max} = 1$.

It can be shown that the maximum concentration of pheromone on a path is bounded by $\frac{1}{1-\rho}\frac{1}{O^{best}}$, where $\frac{1}{O^{best}}$ is the objective value being used to update the pheromone trails. This can be shown be imagining that pheromone is added to the same edge $i \rightarrow j$ in every iteration of the algorithm. For this edge, the pheromone concentration τ_{ij} is shown in equation 7.7:

$$\tau_{ij}^{max} = \sum_{i=1}^{t} \rho^{t-i} \frac{1}{O^{best}} + \rho^t \tau_{ij}(0) \tag{7.7}$$

which converges to $\frac{1}{1-\rho}\frac{1}{O^{best}}$ as t $\rightarrow \infty$, since $\rho < 1$. Equation 7.8 shows this more clearly, note that $\tau_{ij}(0)$, is initialised to an estimate of O^{best}, and each of the O^{best} terms are not identical, but the objective value used to lay pheromone at each iteration of the algorithm (Stültze and Hoos, 2000).

$$\tau_{ij}^{max} = \frac{1}{O^{best}} + \rho\frac{1}{O^{best}} + \rho^2\frac{1}{O^{best}} + ... + \rho^t \tau_{ij}(0) \tag{7.8}$$

Initially τ_{max} is set to be $\frac{1}{1-\rho}\frac{1}{O^{best}}$, where ρ is the evaporation rate of the pheromone, and O^{best} is an estimate of the optimal objective value. Meanwhile, the lower limit for pheromone trails, τ_{min}, is shown in Equation 7.9:

$$\tau_{ij}^{min} = \frac{\tau^{max}(1 - p_{dec})}{(avg - 1)p_{dec}} \tag{7.9}$$

where $p_{dec} = \sqrt[n]{p_{best}}$, and p_{best} is a given probability of an ant choosing the best path once an MMAS algorithm has converged. Convergence here does not mean that all the ants use the same path as in the biological example, but instead a path has become marked with τ_{max} concentration of pheromone, and all other paths are

marked with τ_{min} concentration of pheromone. This value p_{best} can be chosen, and it will then imply a corresponding value of τ_{min}. Therefore, the choice of p_{best} characterises how much the ants will stick to the best path they have found so far, or search other paths, as the algorithm converges.

7.5.5 Summary

This section has justified the decision to use the specific ACO extension MMAS. Additionally it has also been shown how to implement an MMAS version of ACO using an alternative set of pheromone update and evaporation equations, and a set of moving upper and lower bounds on the value pheromones can take.

7.6 Application of MMAS to the pharmacy optimisation problem

This section will explain how the MMAS optimisation algorithm has been applied to the CPN model of community pharmacy developed in Chapter 6. Considering the CPN developed in Chapter 6, 3 decision variables for the optimisation could be immediately considered:

1. The number of dispensers,
2. The number of pharmacists,
3. The checking strategy to use while dispensing.

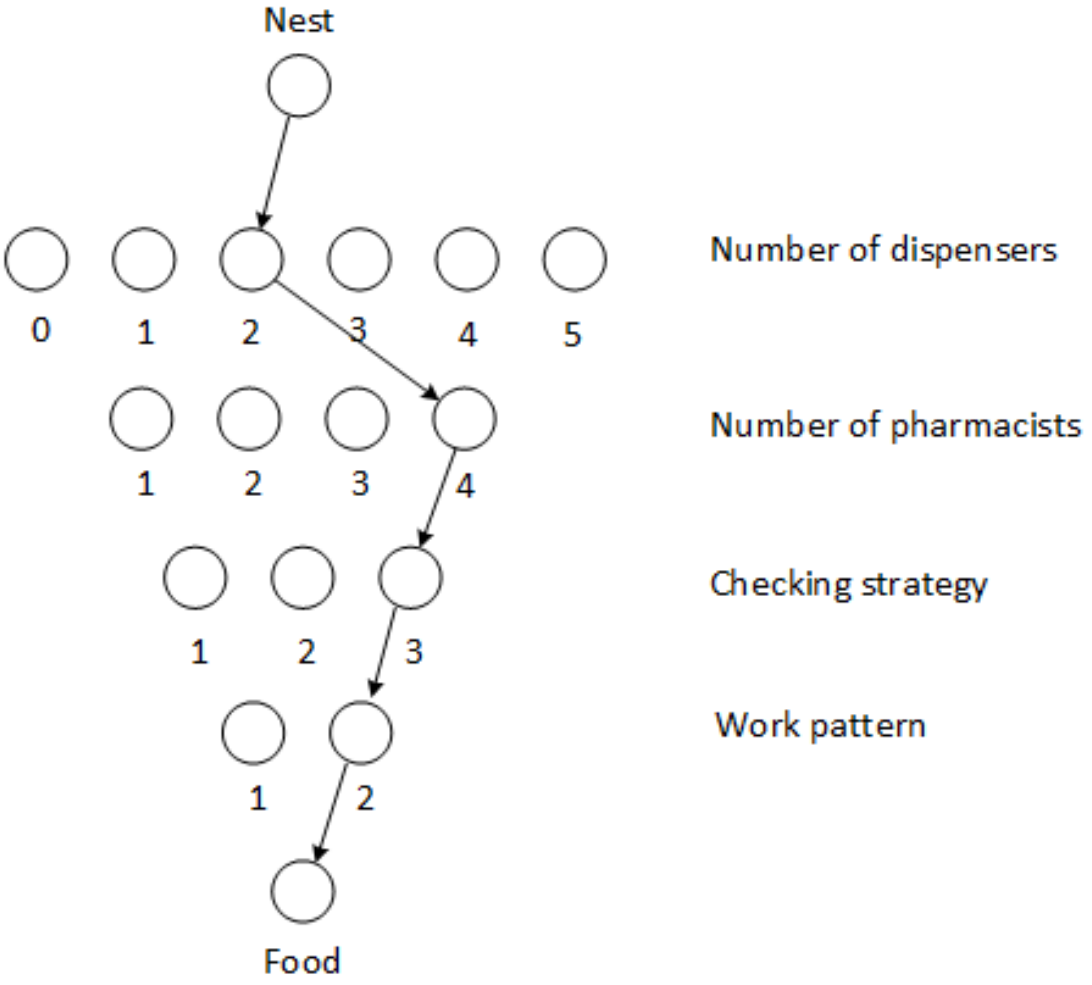
A fourth decision variable is added in this Chapter, which is used to allow or disallow pharmacists to contribute to the primary tasks of dispensing and further investigate the efficacy of different work patterns. Therefore within this framework a pharmacy set-up, P, takes the form of a 4 tuple of integers, $P = (d, p, c_{strat}, w_{pattern})$, where d is the number of dispensers, p is the number of pharmacists, c_{strat} the checking strategy used, and $w_{pattern}$ the work pattern.

The optimisation of a community pharmacy can be defined as choosing the best inputs in terms of staff, work patterns, and safety precautions, which maximise the safety of the process and patient satisfaction, while minimising the costs.

7.6.1 The state space

By assigning each of the decision variables a range of valid values, a state space for the optimisation problem can be constructed. Figure 7.3 shows a set of nodes representing the state space of the problem within the ACO framework. The state space is made up of layers of nodes, where arcs exist between the layers, but not within. Ants must choose a single node from each layer to create a route from their nest to the food source. For example, the path in Figure 7.3 corresponds to a pharmacy set up with 2 dispensers and 4 pharmacists, using checking strategy 3, and work pattern 2.

FIGURE 7.3: Example ACO configuration



Note that there is a legal requirement for a pharmacist to be present if a pharmacy is going to dispense prescriptions. There is no such requirement for dispensers, i.e. a pharmacy can operate without dispensers. The maximum values of d and

p have been chosen to be representative of a small to medium sized pharmacy in the UK.

The three checking strategies are as in Chapter 6:

1. A single check is performed by the pharmacist.
2. Two checks, the first done by a dispenser, and the second by the pharmacist. Any errors which are found are sent back to stage 2 to be dispensed again.
3. Two checks are performed as in 2. Any errors which are found by dispensers are attempted to be fixed straight away, without going back to stage 2. Pharmacists send errors to be dispensed again.

Each strategy is feasible, although the prevalence of each strategy in day to day practice is unknown. The two work patterns are defined:

1. Pharmacists are able to contribute to primary dispensing tasks, such as receiving prescriptions from patients or dispensing prescriptions.
2. Pharmacists are unable to contribute to primary tasks.

There are 132 unique pharmacy set-ups in this problem ($(6 \times 4 \times 3 \times 2)$ - 12 = 132), since 12 set-ups are inappropriate because they have 0 dispensers and use work pattern 2. In such a situation, no primary tasks are completed since the pharmacy team consists only of pharmacists, who are restricted to non-primary dispensing by work pattern 2.

This small state space may not initially appear to justify the use of a meta-heuristic search, since an exhaustive search is also possible (see section 7.7.4). However, if the staff decision variables were diversified by breaking up the working day into multiple shifts, the state space would increase significantly. As an example, by partitioning the working day into 3 shifts of equal length, the state space would include 82,944 scenarios ($(6 \times 4)^3 \times 3 \times 2$). Therefore, even small increases in the decision variables can result in a large number of scenarios, unsuitable for the exhaustive search method.

Further expansions of the optimisation problem could be introduced by allowing for additional decision variables, or allowing the existing decision variables to range over more values. Another examples of a decision variable that could be included in the optimisation is the prioritisation of tasks that staff members use. Alternatively, the state space could be increased by expanding existing decision

variables, e.g. allowing for more dispensers (or pharmacists), or introducing more potential checking strategies.

7.6.2 Cost of wages

The cost of employing the required staff for each set up is taken into consideration. According to the Office of National statistics, in 2015 the median earnings of UK pharmacists was £41,500, and £21,134 for a dispensing technician (ONS, 2015). Therefore, the cost of wages is calculated using Equation 7.10:

$$\textbf{Cost} = 41,500 \times p + 21,134 \times d \tag{7.10}$$

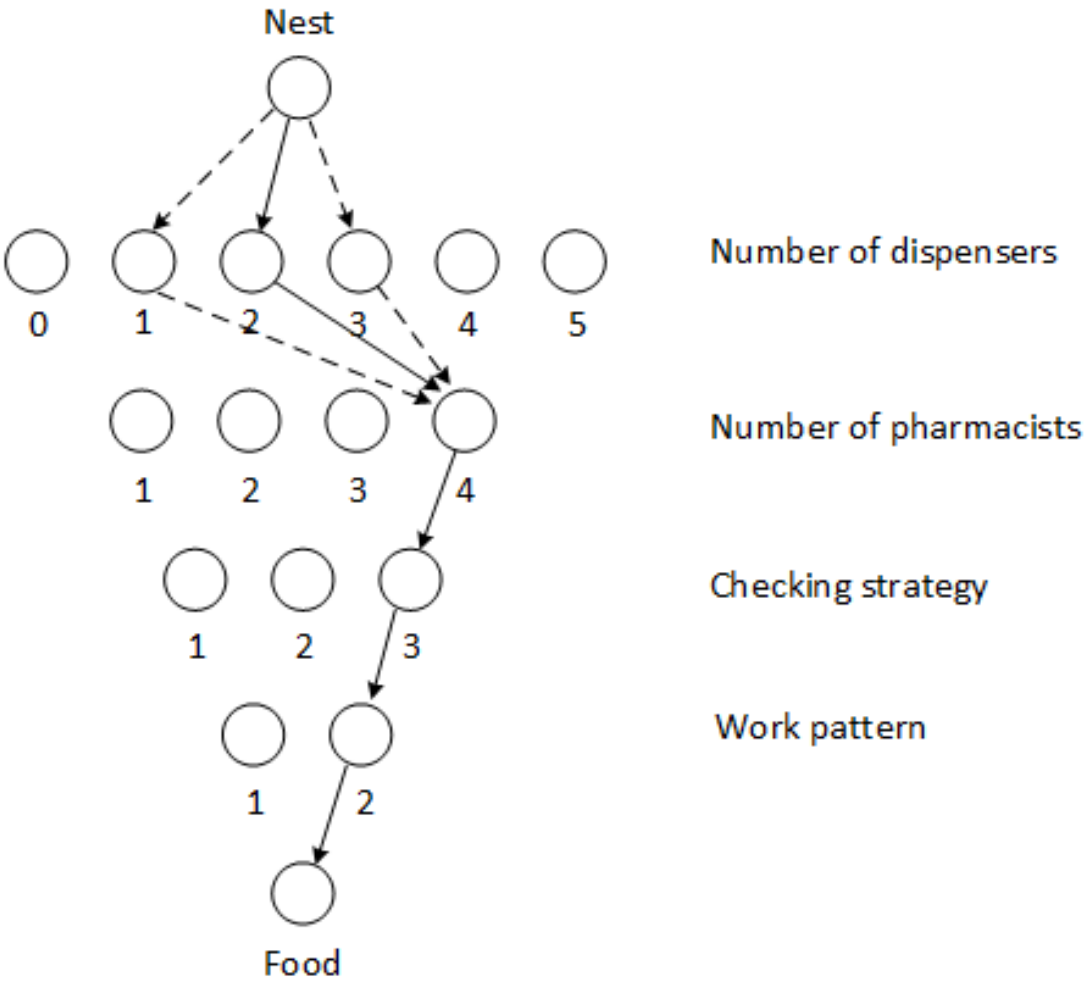
The overall cost is used to compare the price of different set ups. Additionally, the costs for dispensers and pharmacists can be used to derive heuristic information for the MMAS algorithm (see Section 7.6.4).

7.6.3 Local search

ACO algorithms can also make use of localised search patterns. This is a situation where the path of an ant which has returned a good solution is modified slightly, based on the belief that other good solutions may be in the surrounding neighbourhood. In this implementation of MMAS, a local search step is applied after the initial search phase. In the initial search a number of ants construct and evaluate solutions, of which the best ant is sent to a local search. The local search works as follows: one of the layers of nodes is randomly chosen for a local search with uniform probability. The path is kept constant except in the layer being locally searched, where the nodes 1 above and 1 below are tested. In the case of the work pattern, only one other option is tested.

Figure 7.4 shows an example of how the local search is applied. The best ant path from the initial search was 2 dispensers and 4 pharmacists, using checking strategy 3, with work pattern 2. Then in the local search phase, the same paths but with 1 dispenser, and 3 dispensers is tested. The best of the 3 tested at the local search stage is the iteration best path.

FIGURE 7.4: Local search implementation



7.6.4 Heuristic information

One of the requirements of the optimisation is to find low cost solutions to the problem, since lower cost options may be more desirable to decision makers. To have the option of incorporating this preference, where a cost is associated with choosing an option, the heuristic value of an option is calculated as the inverse of the cost using Equation 7.11:

$$\nu_{ij} = \begin{cases} 1/(jC_d) & \text{if } i = nest, \text{ j} \in L_1 \\ 1/(jD_p), & \text{if } i \in L_1, \text{ j} \in L_2 \\ 1 & \text{else} \end{cases} \tag{7.11}$$

where ν_{ij} is the heuristic value for node j if an ant is on node i. L_1 and L_2 are the first and second layer of nodes in the search space, C_d is the cost of a dispenser, and

C_p is the cost of a pharmacist in $£1 \times 10^4$. For example, the heuristic value from the nest to a node indicating a solution with one dispenser, has a heuristic value of $\frac{1}{2.1134} \approx 0.48$. This way of composing the heuristic values promotes the exploration of cheap set ups to the community pharmacy problem. By inverting the heuristics, more expensive solutions to the problem can be encouraged too. Equation 7.12 below shows how heuristics can be inverted to promote more expensive solutions:

$$\nu_{ij} = \begin{cases} 1/(C_{max} - C_j) & \text{if } i = nest, \text{ j} \in L_1 \\ 1/(C_{max} - C_j), & \text{if } i \in L_1, \text{ j} \in L_2 \\ 1 & \text{else} \end{cases} \tag{7.12}$$

where C_{max} is the maximum cost for the layer of nodes which node j is in, and C_j is the cost associated with node j.

7.6.5 Utility function

When evaluating the performance of each set-up, there are multiple variables used to consider whether it was a successful set-up. In this paper, the average waiting time, the number of errors, and the number of prescriptions completed are used. This represents a multi-objective optimisation problem. To transform this multi-objective optimisation problem into a single objective optimisation, a three termed utility function is proposed, shown in Equation 7.13:

$$U = \lambda_1 X_{Waiting} + \lambda_2 X_{Errors} - \lambda_3 X_{Completed} + \gamma \tag{7.13}$$

where $\lambda_1, \lambda_2, \lambda_3 > 0$. Better solutions to the problem will minimise the value of the utility function, U. Such solutions will, therefore, minimise the waiting time and the number of errors, while maximising the number of prescriptions being completed. A constant of $\gamma = 30$ is added to ensure positive value of all evaluations. Each of the variables were considered to be of relatively equal importance, so the three weighting variables were chosen such that the value of each of the variables would be of the same order. λ_1 was chosen to be 0.1, λ_2 was chosen to be 60, and λ_3 was chosen to be 0.1. These weightings imply that an improved average waiting time of 5 minutes, has the equivalent value of reducing the average number of errors by 0.3 per day, or completing 300 more prescriptions.

7.6.6 Pheromone update

In this Chapter the pheromone updates were controlled using a cycle of 3. Every 3rd pheromone update was done by the best ant found so far, and every other update was done using the iteration best ant.

7.6.7 Summary

This subsection has shown how the MMAS optimisation algorithm can be applied to the community pharmacy set-up problem. This has included: setting up a state space to represent the problem, assigning costs to each pharmacy set up, implementing a local search framework, including some heuristic information based on costs, setting up a utility function to evaluate pharmacy performance, and specifying the pheromone update procedure.

7.7 Results

The proposed optimisation algorithm used 2 phases for each iteration. The first phase used 4 'search' ants, where each ant constructed solutions on the state space, and returned objective values after simulating the CPN with a set-up corresponding to their path choice 500 times. The number of ants affects how many objective evaluations are needed at each iteration of the algorithm, and the number of fresh solutions that are tested. The path of the 'search' ant which returned the lowest objective value, was then passed to the local search (the second phase). The local search tested the original path a further 500 times, and then checked each of the neighbouring variations another 1000 times each. Pheromone trails were updated at the end of each iteration. The estimate for the minimum objective, $\tau_0 = 48.2$ was obtained by running the algorithm for 1 iteration.

The MMAS algorithm was run in three stages, presented in Table 7.2. Stage 1 did not use any heuristic information, stage 2 included heuristic information which encouraged the exploration of cheaper solutions, and stage 3 used heuristic information which encouraged the exploration of more expensive solutions. Each optimisation was run for a maximum of 50 iterations, with a stopping condition such that if the same iteration best path was returned 3 times consecutively the

Table 7.2: MMAS parameter settings

Parameters	1	2	3
α	1.0	1.0	1.0
β	0.0	1.0	1.0
ρ	0.98	0.98	0.98
ν_{ij}	n/a	$\begin{cases} 1/C & \text{if C>0} \\ 1, & \text{if C=0} \end{cases}$	$\begin{cases} \frac{1}{C_{max}-C} & \text{if C}\neq C_{max} \\ 1, & \text{if } C = C_{max} \end{cases}$
m	4	4	4
τ_0	48.2	48.2	48.2
Max iterations	50	50	50
p_best	0.2	0.2	0.2

algorithm would terminate. The Pareto optimal solutions were found for the results of each instance of the algorithm before being combined into a single set of solutions.

A characteristic issue with using a heuristics derived from nature (HDN), of which ACO is one, is that their performance is controlled by a set of user determined parameters, and finding good values for these parameters can represent a whole new optimisation problem in itself (Engin et al., 2007). There are several approaches one can take to set the parameters. These have previously included: ad-hoc selection (Aytug et al., 2003), utilising recommendations from previous research (Zecchin et al., 2005), or using experiments to identify optimum parameter settings (Kucukkoc and Zhang, 2015) (Li et al., 2010). One issue with using parameters that have performed well in previous studies is that the parameters can be very sensitive to the specific problem domain.

Rather than set up a secondary optimisation problem of finding the best parameters to use for the ant colony optimisation problem, in this approach there is a preference for using previously used parameters, and ad-hoc selection based on preliminary testing of results. Once a feasible set up has been determined a variable heuristic method is proposed to maximise the ants exploration of the solution space. The variable heuristic method involves running the algorithm under 3 heuristic preferences. The three heuristics, used once in each stage, are as follows:

1. No heuristic preference
2. Heuristic preference for cheaper solutions
3. Heuristic preference for expensive solutions

After the optimisation has been run using one of the heuristic preferences, the pheromone values are reset, and the optimisation is run again under the alternative heuristic conditions.

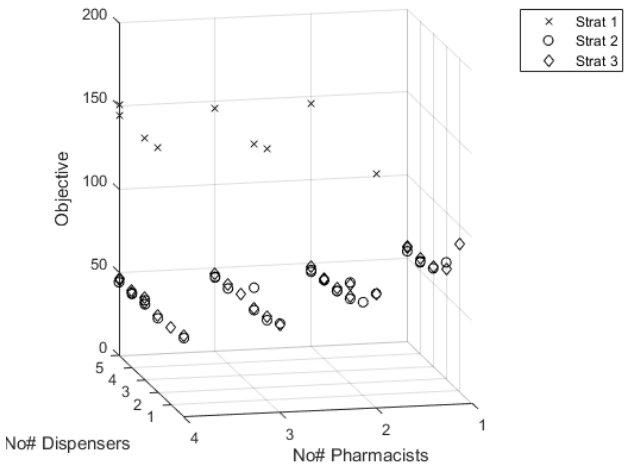
The parameters used during each stage of the optimisation are shown in Table 7.2. The evaporation rate, $\rho = 0.98$, has been used previously in (Stültze and Hoos, 2000) and (Zecchin et al., 2005). The algorithm is very sensitive to the choice of ρ. It must be chosen such that, $0 < \rho < 1$. A value close to 1 indicates very little evaporation, and a colony using a value near 0, will loose almost all information at each iteration. The size of the ant colony was set to be 4, this is at the lower end of potential colony sizes (Shmygelska and Hoos, 2005), since the optimisation state space is relatively small. A β value of 0 is used during the first stage since no heuristic data is used, and the following stages use a β value of 1 so that the different heuristics are taken into account. Stultze et al. Stültze and Hoos (2000) use $p_{best} = 0.005$ in a travelling salesman problem application, however this value seems to be much too small for this size of problem, since the probability of choosing a given path with no pheromone effect is $1/72 = 0.0138$. Using this in Equation 7.9 leads to a τ_{min} value larger than the initial τ_{max}. To compensate for this a larger value of p_{best} was chosen, $p_best = 0.2$.

7.7.1 Stage 1

The objective values returned by all ants at the local search during stage 1 are shown in Figure 7.5. Each point in the x-y plane corresponds to a choice of staff team: a number of pharmacists and a number of dispensers. Each ant is marked using a different symbol depending on the chosen checking strategy, for each point in the x-y plane, there is variation in objective values. The vertical variation is caused by different checking strategies and work patterns. Note that local ants, which found the same set-up, have been aggregated into a single point, so none of the vertical variation is due to random variations in the CPN simulations.

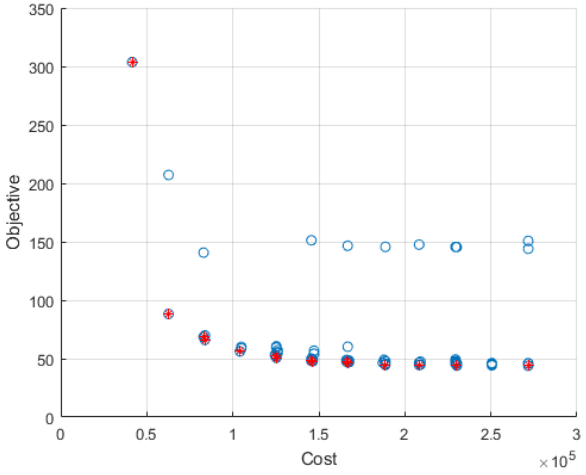
It can be seen that the results form two distinct layers. The lower layer is made up of variations using checking strategies 2 and 3. These set-ups perform well, and the clusters of results indicate that different checking strategies and work patterns can be viable for each choice of staff. However, for many points in the x-y plane there is a worse performing combination of work pattern and checking strategy that produces greater objective values, which form the upper layer. The

FIGURE 7.5: State space of solutions for stage 1



upper layer of poor performing set-ups uses checking strategy 1, as the upper layer is almost entirely made up of cross markers. This strategy has no intermediate check, and is producing worse objective values when the same number of staff are employed.

FIGURE 7.6: Objective function plotted against cost for stage 1



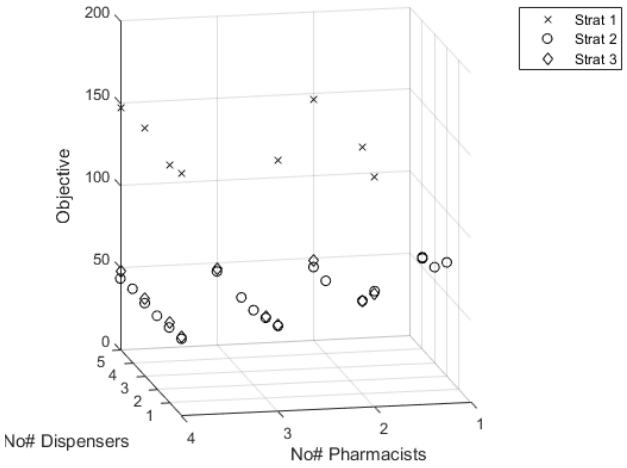
Each set-up is assigned a cost using the annual salaries of dispensers and pharmacists in the UK. Figure 7.6 shows a two variable scatter plot of cost against the utility function for all aggregated local ants returned during this stage of the optimisation. Pareto optimal set-ups are indicated with red markers. For many of the solutions costing more than £1.2×10^5 per year, the optimisation generated a group of results with low utility function values (between 40 and 60). The Pareto

front contained 15 solutions (Note: 4 pairs of Pareto solutions were found close together in cost and utility at: £1.66×10^5, £1.45×10^5, £1.24×10^5, and £0.83×10^5).

7.7.2 Stage 2

The second stage of the optimisation was performed using a heuristic preference for cheaper pharmacy set-ups. Pheromones were reinitialized to τ_0, and the algorithm was run again while using heuristic information. The heuristics used were equal to the inverse of the cost of each set-up in thousands of pounds. For example, the heuristic probability of an ant moving from the nest to a node representing 2 dispensers would be $\frac{1}{21*2}$. Note, if a set up chose 0 dispensers, the heuristic value is 1, indicating no negative preference for the cheapest solution. Figure 7.7 shows the aggregated utility values obtained by all local ants in the algorithm when using the parameter set 2.

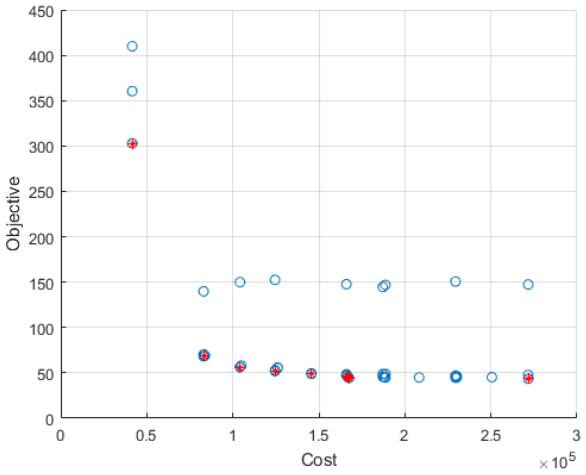
FIGURE 7.7: Objective state space for stage 2



In this stage of the optimisation, the heuristic data directed the ants to search more set-ups with a lower cost and fewer expensive solutions were tried. The search was also much less exploratory than in stage 1. This is indicated in Figure 7.7 by decreased clustering of results around set-ups which contained more staff, and some clustering of results around the cheaper solutions. There is a similar pattern of how set-ups are affected by the checking strategy, as was seen in stage 1. Set ups using checking strategy 1 performed worse than those using 2 or 3.

Figure 7.8 confirms that more solutions of a lower cost and less of a higher cost were tested. For example, three solutions with an annual cost of less than £0.5×10^5 were tested in stage 2, compared with just one in stage 1. There were far fewer alternative solutions tested for set ups costing more than £1.5×10^5 during stage 2 of the optimisation. Red markers indicate the 9 Pareto optimal solutions found during stage 2 (Note: 3 Pareto solutions were found with very similar cost and utility at £1.66×10^5). Five of these solutions were also part of the Pareto front of solutions during stage 1.

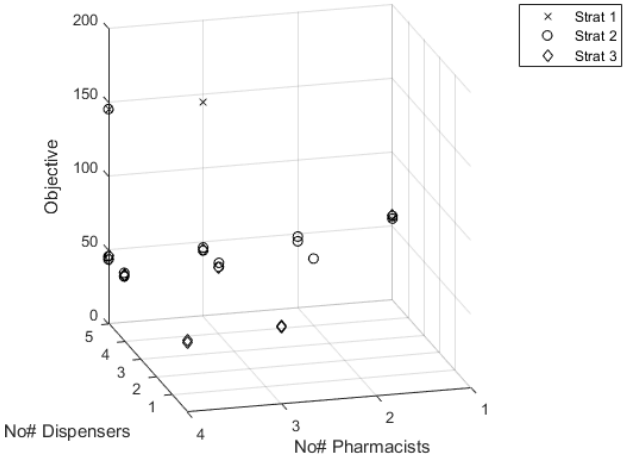
FIGURE 7.8: Objective plotted against cost for stage 2



7.7.3 Stage 3

Stage 3 used inverted heuristic information to encourage the algorithm to search more expensive solutions to the problem. Previous heuristic values were inverted, so that the most expensive set-ups were preferentially chosen, in the same way that the least expensive set-ups were preferentially chosen in stage 2. Expensive solutions are not normally targeted in this way by heuristic information. This is due to an assumption that more expensive solutions will perform better. The assumption may not always hold, and by using this heuristic, any very strong solutions expensive solutions to the problem are more likely to be found. As an example of how they have been implemented, the heuristic value used for a path indicating a choice of 0 dispensers was set to $\frac{1}{5*21}$, and the heuristic value for a path containing 5 dispensers was 1.

FIGURE 7.9: Objective state space for stage 3



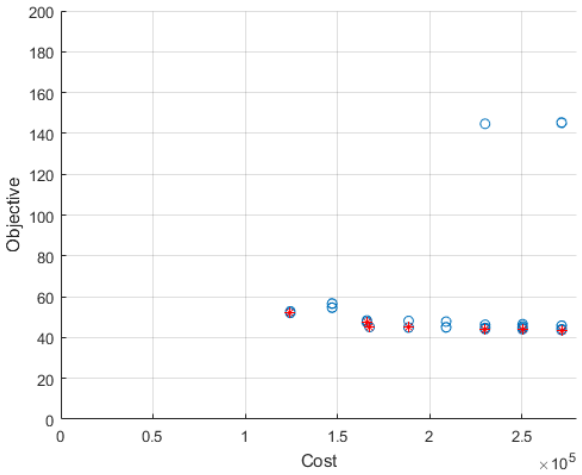
It can be seen in Figure 7.9 that the ants tested solutions containing larger numbers of staff more frequently than in the previous two stages. Most solutions tried contained either 5 or 4 dispensers, and few solutions containing small numbers of staff were tested. There is less of a discernible pattern in the checking strategies during this stage, this may be due to solutions using checking strategy 1 being tested less than in the previous stages.

Figure 7.10 shows that the algorithm explored expensive solutions to the problem, and no solutions costing less than £1×10^5 were evaluated during this stage. This may be due to the strong preference of the heuristic information to select only the most expensive set ups, leading to slightly less expensive set-ups being left out. Furthermore, since the expensive set-ups may be likely to be strong solutions to the problem, there is less motivation for the ants to explore other solutions. In stage 3, only 7 Pareto optimal solutions were generated, of which, 5 had not been Pareto optimal in either stages 1 or 2.

7.7.4 Combining and comparing results with the exhaustive search

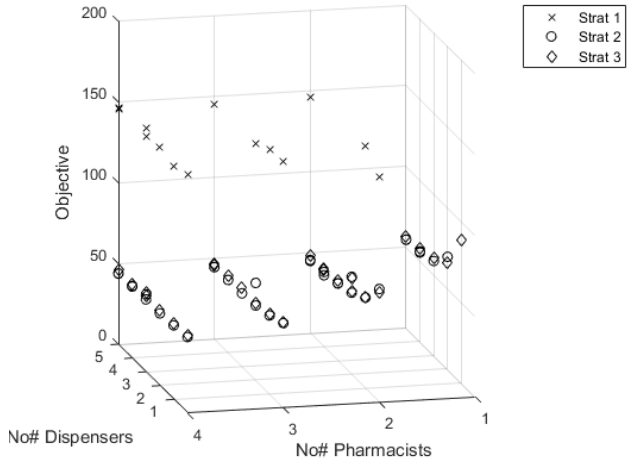
Overall, the optimisation evaluated 84 unique solutions at the local search stage. The combined results from all three stages are plotted in Figure 7.11. Of the 84 solutions, 14 set-ups were non dominated, i.e. each solution in this set, cannot be improved without increasing the price, shown in Table 7.3. The optimal set

FIGURE 7.10: Objective values against cost for stage 3



up, was a team of 5 dispensers and 4 pharmacists, using a combination of checking strategy 2, and a non-flexible work pattern. Each stage of the optimisation contributed at least 1 Pareto optimal solution that was not found by the other stages. Stage 1 contributed 4 unique Pareto solutions, stage 2 contributed 2 unique Pareto solutions, and stage 3 found only a single unique Pareto solution. All other solutions were found by multiple stages.

FIGURE 7.11: Objective state space for combined stages 1-3



The solutions in the Pareto front were made up of teams containing either a majority of dispensers (5), a majority of pharmacists (6), or an equal number of each (3). Solutions 1-4 were high quality solutions, performing almost as well as the optimal set up.

The solutions 5-14 were of increasingly worse quality, with solution 14 being significantly worse than the rest. A general trend is: if money is not limited, better teams will be those with a high number of staff, with either an equal number of each staff type, or more dispensers. However, if the budget for hiring staff is limited, the best set-up may contain fewer staff, and its likely that the best set-up will contain a majority of pharmacists. This can be seen by comparing the expensive set-ups in the Parto front shown in Table 7.3, against the cheaper set-ups. For example, of the top 6 solutions all except 1 contains a majority of dispensers. Where as, looking at the bottom 6 (solutions 9-14), all but 1 solution had a majority of pharmacists or an equal number of each.

TABLE 7.3: Pareto Optimal Solutions: MMAS

Solution	Dispensers	Pharmacists	Checking Strat	Work Pattern	Waiting time (s)	Errors	Completed	Cost($£year^{-1}$)	Objective
1	5	4	2	0	282.4	0.18	251.6	271670	43.7
2	4	4	2	0	290.5	0.17	251.5	250536	44.3
3	5	3	2	1	286.8	0.18	251.5	230170	44.3
4	4	2	2	1	306.1	0.15	251.7	167536	44.4
5	0	4	2	1	314.3	0.18	251.6	166000	47.2
6	3	2	2	0	323.7	0.18	250.9	146402	47.8
7	1	3	2	1	325.5	0.19	251.5	145634	48.6
8	2	2	2	1	346.2	0.19	251.6	125268	50.7
9	0	3	2	1	364.1	0.18	248.6	124500	52.4
10	1	2	2	1	397.9	0.19	246.1	104134	56.3
11	2	1	3	1	466.1	0.20	228.7	83768	66.0
12	0	2	3	1	456.2	0.19	186.1	83000	68.4
13	1	1	3	1	656.6	0.17	178.2	62634	88.3
14	0	1	3	1	2766.1	0.09	88.9	41500	303.0

The purpose of the optimisation was to avoid checking the entire state space exhaustively, and generate potentially good solutions using fewer simulations than would be required in an exhaustive search. The three optimisation stages combined used a total of 750,000 simulations. To test the Pareto optimal solutions generated by the ACO, an exhaustive search was also conducted on the state space. During the exhaustive search each of the 132 possible pharmacy set ups was simulated 15,000 times, which required 1,980,000 simulations. The ACO optimisation ran for a total of 19.8 hours, and the exhaustive search took 44.7 hours. In terms of time saved, the ACO optimisation routine took 24.9 hours less than the exhaustive search.

Table 7.4 shows the Pareto set of solutions from the exhaustive search. Solutions 7-14 returned by the ACO algorithm appeared in almost exactly the same order and position as in the set of Pareto solutions generated by the exhaustive search. However, solutions 1-6 from the ACO algorithm did not appear in the exhaustive search Pareto front, although 4 solutions were very close, i.e. solutions 1, 3, 4 and 6 all appeared in the exhaustive Pareto front, but using the alternative work pattern.

The results were different due to two factors, one reason was that sometimes very similar set-ups which performed almost identically, were being differentiated by random variations in the CPN simulation results. The other reason the Pareto front differ is that the ACO did not try every possible solution, and some of the solutions that were not tested were optimal.

TABLE 7.4: Pareto Optimal Solutions: Exhaustive

Solution	Dispensers	Pharmacists	Checking Strat	Work Pattern	Waiting time (s)	Errors	Completed	Cost($£year^{-1}$)	Objective
1	5	4	2	1	283.5	0.18	251.7	271670	43.7
3	5	3	2	0	284.6	0.18	251.6	230170	44.2
4	5	2	2	0	295.32	0.17	251.5	188670	44.7
5	4	2	2	0	301.8	0.18	251.5	167536	45.9
6	2	3	2	1	307.7	0.18	251.6	166768	46.5
7	0	4	2	1	314.2	0.18	251.7	166000	46.9
8	3	2	2	1	322.3	0.18	251.5	146402	48.0
9	1	3	2	1	325.4	0.18	251.6	145634	48.2
10	2	2	2	1	346.1	0.18	251.5	125268	50.5
11	0	3	2	1	364.2	0.18	248.6	124500	52.1
12	1	2	2	1	398.2	0.18	246.0	104134	55.8
13	2	1	3	1	470.0	0.23	228.0	83768	67.3
14	0	2	3	1	456.4	0.18	186.2	83000	67.7
15	1	1	3	0	646.3	0.16	163.0	62634	87.7
16	0	1	3	1	2769.2	0.09	88.8	41500	303.5

It was noted when looking at the results of the exhaustive search that many solutions using the same staff set up and checking strategy would return similar objective values. Only solutions 2 and 5 from Table 7.3 made no appearance (including alternative work patterns) in the exhaustive Pareto front. These may have been considered optimal by the ants due to a lack of familiarity. For example, after a close look at the results it is clear that solution 5 was tested only once (1,000 simulations) by the ants throughout all 3 stages of the optimisation, and the strong evaluation of this solution may be due to a small variation because of the randomness inherent in CPN simulations. On the other hand, in some cases, the preference of the ants for alternative work patterns with the same set ups may be justified. The ants evaluated solution 1 in Table 4 to be preferable to the same set-up with a flexible work pattern (the best solution from the exhaustive search), and during the MMAS optimisation both variations were tested 27 times (27,000 simulations). Hence the ants indicating that this set-up is preferable in this instance should be considered more valid than the result of the exhaustive search since they have simulated both solutions more times.

In summary, the optimisation results compare well with the results generated by the exhaustive search, although results that the ants consider optimal that have only been tested a small number of times should be verified by further simulations before being put into practice. The performance of the algorithm on larger state

spaces remains untested, and further investigation is needed to evaluate whether ACO algorithms are a good choice in such situations.

The Pareto front is only a small subsection of the results generated during the optimisation. If a decision maker was to specify a price, it would be possible to provide a large variety of non-Pareto optimal solutions at that price point. The set ups would not be optimal, however having a set of options to choose from would allow for alternative configurations when there are staff or resource shortages.

7.7.5 Summary

This subsection has presented the results of the ACO optimisation algorithm developed and applied to the CPN model from Chapter 6. The ACO optimisation was run in three stages using a variable heuristic which initially gave no preference to any solutions, then in subsequent stages cheaper or more expensive solutions were favoured. The algorithm's results were compared with results derived from an exhaustive search of the state space.

7.8 Conclusion

This Chapter has shown the development of the MMAS optimisation algorithm used for optimising community pharmacy set-ups. The optimisation is run over 4 decision variables: number of dispensers, number of pharmacists, work pattern, and checking strategy. The optimisation was run 3 times under different heuristic information patterns to increase the exploration of the ants by promoting the search of cheap and expensive solutions. Results of the MMAS optimisation were compared to an exhaustive search of the solutions. MMAS was shown to be a suitable optimisation framework for finding community pharmacy set-ups.

Chapter 8

Conclusion

The work completed in this thesis has been undertaken to use modern reliability modelling and optimisation techniques to improve the safety and efficiency of dispensing prescriptions to patients. The modelling and optimisation tools developed throughout this thesis may be able to provide evidence for decision makers when deciding how best to set up community pharmacies.

8.0.1 CPN model

The thesis has demonstrated that Coloured Petri Nets (CPN) are an effective tool for evaluating the performance of community pharmacies. The CPN developed throughout the thesis can assess different pharmacy configurations, which include variable numbers of dispensers and pharmacists, use one of 3 different checking strategies to check prescriptions, and follow one of the 2 work patterns. The three checking strategies are: no intermediate check, an intermediate check where erroneous prescriptions are sent to be dispensed again, or an intermediate check where erroneous prescriptions are fixed as they are found. The two work patterns are: pharmacists are able to contribute to the completion of primary dispensing tasks such as dispensing and receiving prescriptions, or pharmacists are not able to contribute to such tasks. Checking strategy 2 produced consistently stronger results, whereas it was unclear if either work pattern was strongest. The non-flexible work pattern was preferable for larger budgets, but the flexible work pattern was more effective with lower budgets.

Pharmacy set-ups are evaluated using a set of performance indicators. These include, patient waiting time, the number of near misses and dispensing errors handed out to patients, the number of prescriptions dispensed, and the number of delayed prescriptions. Further performance indicators, such as how long staff spend their time inactive, can also be derived from the model.

The model indicated that if it were possible to enhance the ability of accuracy checking to the point where the check became infallible, the cost in terms of increased averaged waiting times for patients would be low.

The model is capable of producing a set of performance indicators, and different checking strategies and work patterns can be tested and evaluated. Similar modelling work has been completed in the field of healthcare, where only a single aspect of a process, either the reliability or efficiency is considered. Set-ups using an intermediate check where erroneous prescriptions are sent to be dispensed again produced the highest reliability, and efficient set-ups were generally composed of a large number of staff. The main novelty of the CPN model proposed in this thesis is the ability to model the reliability and efficiency of a healthcare process, within a single modelling framework. The final version of the CPN can complete a simulation of a 9-5 working day of pharmacy operations in 0.59s. The convergence analysis showed that for all of the performance indicators, the values will have converged within 2000 simulations.

8.0.2 In-field data

Data was collected from 4 UK community pharmacy sites to complement and inform the modelling work carried out in this thesis. Two types of data were collected: quantitative timings of the stages of the dispensing process, and qualitative interviews with community practitioners about their views of the dispensing process.

The quantitative data collected was a set of recordings of how long individual stages of the dispensing process took. Where appropriate, the number of items in the prescription being processed through a stage was also recorded. Once collected, the data was subjected to a distribution fitting analysis. In this analysis, the data was separated into segments, and a best fitting distribution for modelling how long that stage of the process should take for each prescription size was selected.

These distributions were then implemented in the CPN to control the time taken for tasks to be completed. This makes the CPN a model representative of a typical UK community pharmacy.

As well as collecting quantitative timings of the process, qualitative interviews were carried out with practitioners on each of the 4 sites. These were recorded, transcribed, and subjected to a qualitative analysis, which highlighted a number of ways in which the CPN model could be improved. Potential improvements were related to including phenomena seen in community pharmacy practice which were not included in the modelling framework. Examples of these potential changes included: modelling split pack prescriptions, considering the time taken to record near misses, and modelling how patients can visit pharmacies to buy non-prescription goods.

The data collected and analysed as a part of this thesis represents a novel research contribution, as a comparable set of data has not been collected in the field previously. Where data has been collected and published in the literature, the data sets were either small in comparison, or did not consider timing individual stages of the dispensing process.

8.0.3 Optimisation

It has been shown that the community pharmacy dispensing process can be optimised using an Ant Colony Optimisation (ACO) algorithm, a modern optimisation technique based on the way ants find food sources located close to their nest. Choosing values for the 4 configuration variables represents an optimisation problem, where the best set-up must be found. Initial results indicate that significant improvements in reliability can be made by including an intermediate accuracy check in the dispensing process. Furthermore, by assigning each pharmacy set-up a cost based on median yearly wages, and a utility value, consisting of a sum of 3 key performance indicators (waiting time, total number of prescriptions dispensed, and number of dispensing errors were used) a Pareto front of optimal solutions has been constructed. Results in the Pareto front indicated that if a large budget was available the best set-ups would contain a large number of staff, made up of a majority of dispensers, using checking strategy 2. However, if the budget for a team was smaller, the optimal set-up would contain a small number of staff with

an equal number of dispensers and pharmacists, or a majority of pharmacists using a flexible work pattern.

The optimisation work carried out in this thesis represents two novel contributions to the field. The combination of CPN and ACO has not previously been used as an optimisation technology for the problem of community pharmacy set-up, and in addition the three stage heuristic optimisation used to find optimal solutions is a novel method: the conventional technique is to run an optimisation with a fixed heuristic which directs ants to be more likely to choose cheaper solutions to the problem. The conventional method is comparable to just running stage 2 of the optimisation alone. Since each stage produced unique Pareto optimal solutions, the method have been shown to be useful for optimising pharmacy set-ups.

8.1 Future work

The research carried out as a part of this thesis could be extended in multiple directions. The following sections highlight 4 potential areas for future research.

8.1.1 A more detailed CPN

The CPN developed as a part of this thesis did not exhaustively model all aspects of the community pharmacy dispensing process. A number of tasks identified during work sampling studies of community pharmacies which are not currently modelled, could be included into the model. These additional activities identified from work sampling studies included: staff training, housekeeping, sales transactions, money and managerial tasks, and non professional interactions. Further additions to the model could consist of modelling more types of customer requests, and considering a more detailed set of prescriptions, examples of which could include split packs, and controlled drug prescriptions. Further detail could also be included by introducing additional staff types to the model, such as counter assistants or ACTs. Finally, a comparison could be made between different dispensing paradigms using the same modelling technique. This would involve comparing the performance of automated dispensing against the more standard manual dispensing method. Additional modelling implications identified through the on-site visits could also be included in the CPN. The major aspect of the model which needs to

be improved before it can be used to guide decision makers is the inclusion of a wider set of phenomena, such as patients coming in to buy non-prescription goods, split packs, or other similar commonly occurring significant phenomena within UK pharmacies.

8.1.2 Extended error analysis

The probabilities used to control the occurrence of errors could be a topic of further research. This could involve conducting more detailed fault tree analyses of all the fallible events, or carrying out further on site research to investigate error rates for each stage of the dispensing process.

8.1.3 Increased optimisation search space

Future work could include extending the state space of the optimisation problem. This could be achieved by either including more variables in the set-up configurations, or by increasing the current range of values. Two candidate variables to include in an extended optimisation are: the number of labelling stations available, and which tasks should be given priority when staff are working.

8.1.4 Consider expanding the scope of the modelling

The modelling work carried out in this thesis has focused on the behaviour of a single pharmacy made up of a team of practitioners. Future work could consider large multiples, made up of a set of pharmacies, and how to choose the best way to set up a company owning multiple stores. Additional costs of running a pharmacy, other than staff wages could be considered, or the scope of the modelling could be expanded to include the supply of medicines to pharmacies.

8.1.5 Different level of staff skill

It could be useful to model how different levels of staff experience or skill can impact upon the dispensing process. It was noted during interviews with practitioners that staff can perform to different levels or standards.

8.1.6 Utility function analysis

Further work could include conducting a sensitivity analysis of the weights used in the ACO optimisation algorithm's utility function. Furthermore, it is known that the objective value assigned to different solutions is dependant on the weighting given to each variable. Additional research could aim to quantify pharmacy preferences.

Appendix A

Observational profiles

A.1 Observational profile - Pharmacy 1

A.1.1 Introduction

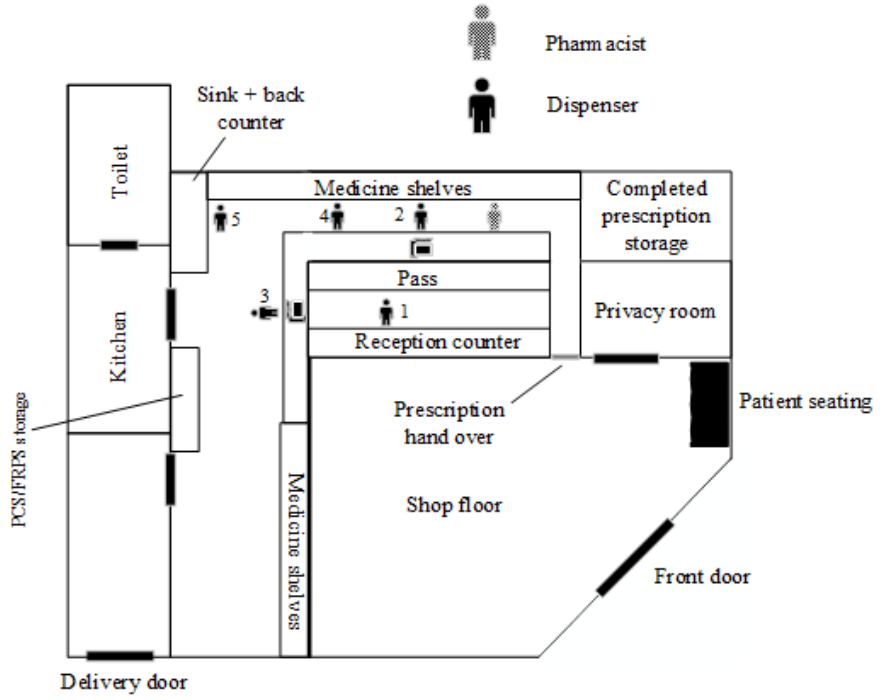
The observation took place over 3 days, Monday 22nd - Wednesday 24th of January 2018, where each day consisted of approximately 9 hours of observation between the hours of 9am and 6pm. The researcher recorded the time taken to complete the individual stages while observing the process, and conducted and recorded interviews with members of staff.

A.1.2 Overview

Figure A.1 shows the spatial layout of Pharmacy 1, and the typical working positions for the members of staff. Each member of staff had a designated task, or set of tasks to complete whilst working. There were 4 dispensers, 1 ACT, and 1 pharmacist present.

Throughout the day the pharmacist was typically carrying out final accuracy *and* clinical checks of dispensed prescriptions. Their position in the pharmacy was close to both the completed prescription storage area, and the area used for handing out prescriptions.

FIGURE A.1: Spatial layout and typical working positions in Pharmacy 1



Dispenser 1 was a counter assistant. They dealt with patients initially arriving in the pharmacy, checking whether an EPS (electronic prescription service) prescription was ready for collection, receiving walk-in prescriptions, selling over-the-counter goods, and informing walk-in patients of the approximate waiting time. They were positioned between the pass and the reception counter, where they are easily accessible to both customers and other members of staff, to help solve any issues.

Dispenser 2 was typically responsible for dealing with walk-in prescriptions, while also generating labels for EPS prescriptions. Their position was close to the main medicine shelves where the majority of the medicines are stored. Dispenser 3 was the branch manager (ACT), and they would be processing EPS prescriptions, or accuracy checking prescriptions which had already undergone a clinical check by a pharmacist. Both dispensers 2 and 3 were positioned in front of a computer.

Dispensers 4 and 5 were typically working on a stage of processing EPS prescriptions. Although once the new stock delivery came, these two dispensers began putting the new stock onto shelves.

A.1.3 Background

The pharmacy is co-located next door to a GP practice, and belongs to one of the large pharmacy chains. The manager mentioned that a lot of the pharmacies foot-fall was related to the nearby medical centre. All staff were aware, and frequently mentioned that 4pm was a very busy time in the pharmacy due to lots of patients visiting the medical centre after school finishes.

The pharmacies opening hours are between 8.15am to 6.45pm. The pharmacy had won Branch of the Month award before, and they usually dispensed between 2700 and 3000 items each week. The pharmacy has two computers (used by dispensers 2and 3), which are used to print labels for prescriptions, and check the whereabouts of prescriptions which cannot be easily located. Most items for prescriptions are located on sliding shelves behind where the majority of the team is working, although sometimes staff have to travel further to the alternative medicine shelves to pick items. These items located further away seemed to be rarer in the frequency with which they were dispensed. Examples of the items located further away included gluten free products, large tubs of milkshake or moisturisers.

A.1.4 Staff

On each day of observation, the team of people working in the pharmacy was fluctuating between 3 and 6 members of staff. Upon arriving in the morning at 9am, 5 members would be present, with another arriving at around 10am. This short increase in staff numbers coincided with the delivery of new medicines, which arrived between 10.45 and 11.15 in the morning each day. A member of staff would finish at 1 pm, and another would leave at 5pm. The manager would leave just before the researcher at 6pm, which left the pharmacist and 2 dispensers to finish off the shift, from 6 to 6.45pm.

2 pharmacists and 6 different dispensers worked during the observation period. The second pharmacist worked on the Wednesday, and 1 Dispenser who had worked on Monday and Tuesday was replaced by a new dispenser on the Wednesday. The researcher was told by the manager that since Christmas, their staff hours had been reduced by 1 member. On the Wednesday, the area manager came

into the store to have a lengthy (around an hour long) meeting with the store manager in the privacy room.

A.1.5 Work flow and prioritisation

The work flow in the pharmacy varied by prescription type. For the majority of prescriptions, a dispenser would process a large number of prescriptions through the same stage one after the other. One example of this was observed on Wednesday. The store manager was printing labels for a large number of prescriptions all day, from between 10.30am - 4 pm. Approximately 200 prescriptions. These prescription forms were placed into baskets with the printed labels, and placed onto the EPS storage shelf. These would then be picked the next day, labelled the day after, and on the following day they would be checked and placed into bags. Other examples were frequently observed of dispensers picking items for a large number of EPS prescriptions, one after the other, and the same with labelling.

The majority of EPS prescriptions would move through the dispensing process at the rate of around one stage of the process per day. After printing the labels for the EPS prescriptions, a member of staff would have to put in an order for all the medicine included in those prescriptions, which would arrive in the delivery the next day. The recommended time for a patient to return to collect their EPS prescription was a week. One of the problems the pharmacy were working to overcome, was communicating this message to patients so that patients didn't turn up too early. If a patient arrived before their EPS prescription was finished they would be told it was not ready, and given the choice whether to wait for it there and then, or come back at an appropriate time. If they chose to wait for it, the prescription would be stored at some intermediate stage amongst many others, which could occasionally cause a bit of confusion to find. Once found, if the patient was waiting in the pharmacy, the prescription would then be treated as a walk-in.

The advantage of this method of dispensing could be seen when patients using the EPS service came at an appropriate time. The process would be as follows. Patient arrives at the counter, asks "Is the prescription for Nastasya Fillipovna ready?" and the counter assistant would go to the completed prescriptions storage area, find their prescription and give it to them. This would often take less than a minute to complete, and cause no disruption to the rest of the team. Priority

was given to walk-in customers. For walk-in prescriptions, the process would be different. A walk-in prescription would be received at the counter and placed into a red basket on the pass. Dispenser 2 would then print the labels, pick the medicine, apply the labels and pass it onto the pharmacist to be checked, before being handed out to the patient. This would take between 10 and 20 mins.

A.1.6 Quantitative data collection

Data collection was intended to follow individual prescriptions passing through the pharmacy, timing how long each stage took. However, most prescriptions were EPS, thus taking multiple days to pass through the pharmacy. This made them unsuitable for the intended data collection scheme. On the other hand, walk-ins did pass through the pharmacy reasonably quickly, but were of a much lower volume, and presented other data collection issues besides.

Besides the lower volume of walk-ins, the other problem was related to how dispensers were dispensing walk-in prescriptions. Observing them, the process was not broken up into the distinct stages nearly as much when dispensing walk-ins. The process would follow this pattern; the labels would be printed and placed into a basket, then almost straight away the dispenser would begin picking items 1 at a time, sticking the label on as they were picked and placing them into the basket. This presented a number of challenges to the data collection. First, some of the stages were intermingled, making it difficult to distinguish between stages, and secondly, there would be no break between the stages (which were relatively short in duration), making it hard to make a record of the data and begin timing the next action accurately.

Faced with these problems, data collection focused on timing the EPS prescriptions on a single stage basis. A single dispenser or pharmacist, repeatedly completing the same task would be observed while working. Due to the number of timings needing to be completed (960), the researcher rarely asked staff to clarify the number of items in prescriptions directly, but attempted to capture the number of items by looking at the prescription form being dispensed, or (at the picking and labelling stages) counting how many items were picked/labelled.

Recording repetitive EPS prescription data was not without its own set of difficulties. Due to the less pressing nature of this kind of dispensing, if staff were

completing a stage for a particularly large prescription, they would fairly frequently get interrupted in their task. Interruptions could come in a number of forms, for example; answering the phone, talking to another member of staff, making a cup of tea, going to greet a customer or any general customer enquiry at the receiving counter, getting a biscuit, helping the pharmacist correct a prescription, opening the door for the delivery driver or talking to the researcher. Larger prescriptions were much more prone to this interruption, and it was often ambiguous as to whether timing should continue on the original task if the member of staff has gone to do another productive task.

A.1.7 Split packs

At the label application stage, dispensers were sometimes required to add or remove a number of tablets from boxes of medicine. This involved opening multiple medicine boxes and removing the blister packs, then using a pair of scissors to add or remove a number of tablets. Finally they would fold up a plain package box into existence, which could contain all the tablets, place all the tablets inside the new box, and finally adhere the label. This had to be done, since most medicine boxes contain 28 tablets as standard, and the GP practice across the road frequently prescribed multiples of 30 (for a month).

It was noticeable how much longer this made the labelling stage of the dispensing process take, as compared to prescriptions where no splitting of packs was required. It seemed to occur more frequently in the larger prescriptions, again further increasing the amount of time they took.

A.1.8 Wastage

Some EPS prescriptions are designated as PCS (patient collection service), these are collected by patients coming into the pharmacy. On the Monday, there was a large box full of completed prescriptions which were wastage. They had been on the completed prescription shelf waiting to be collected for 5 weeks, and no one had come to collect them. The prescriptions had to be unpacked from bags, labels removed, and returned to shelf. The manager estimated that the box of wastage prescription from the previous week contained between 100-150 prescriptions.

Their policy on the PCS prescriptions was thus; if after 4 weeks the patient had not collected the prescription they sent out a letter saying to the patient to come and collect it. If they didn't collect it in the following week, the prescriptions would be unpacked and medicines returned to shelves (provided they were still in date).

Notably, if someone who receives one of the letters turns up during the 5th week, a lengthy search of a big box of unordered prescriptions has to be carried out to find it.

A.1.9 Mood and atmosphere

The pharmacy has a friendly yet professional atmosphere. Initially staff were joking between themselves light heartedly about how I was recording everything, but the novelty wore off by the second day. As the study continued the researched was sometimes involved in the general workplace chat. Most members of staff were interested in why the study was taking place, there was a fairly common misconception among staff at the start of the study that I may have been from one of the company's internal process monitoring teams. Day profile On Tuesday a brief day profile of the store was recorded. Morning 9-12 The store is quite busy with walk-ins, restocking started around 11. Midday 12-3 Less busy in terms of customers. Stock was finished being put away during this period. 1 technician went out for lunch. Pharmacist also took 20 minutes off for lunch. A customers came in needing a prescription while the pharmacist was eating lunch, and the dispenser took the prescription into the lunch room to be checked. 1 customer came in and the pharmacy didn't have the item in stock. Afternoon 3-6 Around 4 there was the busiest period of the day in terms of walk-ins. Many customers came into the store needing walk-in prescriptions during a relatively short period of time.

A.1.10 Interviews

All but one member of staff agreed to take part in research interviews, although out of the 8 members of staff who were working over the three days, interviews were only completed with 5. This was due to some staff not being on the rota for some days of the study period.

A.1.11 Customer acknowledgement

Over the observation, the manager explained one of the methods the company uses to evaluate their store. In the evaluation, one of their pieces of feedback was that they needed to be faster in acknowledging customers when they walked into the store. It was explained that any member of staff should go to greet a customer at the counter as soon as possible. In practice, this was played out. Although it was mostly the counter assistant or the store manager who would go to greet customers voluntarily. Other members of staff would generally have to be directed to go see to a waiting customer by the manager. Staff would more frequently verbally engage customers by saying "Just be a moment", rather than going to attend to the customer immediately.

A.1.12 Low blood sugar levels

One of the members of staff was a diabetic. They were experiencing symptoms of shaking and nausea on the third day of observation for about twenty minutes. Discrepancies between current assumptions and observations The pharmacist rarely, if ever, engaged in any primary dispensing activities. Stocking was completed by between 2 and 3 dispensers, taking approximately between half an hour and an hour and a half. The batch mode of production used for the majority of prescriptions. Pharmacist did leave for lunch (for about 15 minutes) on one day.

A.1.13 Dossette trays

These trays are used to encourage patient adherence when complex medicine regimes are being used. Dispensing them involves a dispensers placing tablets into a compartmented tray, where each compartment corresponds to a time period of the day. These would be done by a different dispenser each day, and would take up 1-2 hours of their time.

A.1.14 Near Miss

One near miss was observed during observations. The pharmacist asked one of the dispensers to print out a fresh label for a prescription. The label stated take

when needed, instead of 4 times a day. There was not a large disruption to the pharmacy, the dispenser quickly printed out a new label and passed it to the pharmacist. The researcher was unsure whether this had been recorded as a near miss or not.

A.2 Observational profile - Pharmacy 2

A.2.1 Introduction

The observation of this pharmacy took place over 4 days, Monday 5th of Feb - Thursday 8th of Feb 2018. Where each of the first 3 days consisted of approximately 9 hours of observation, between the hours of 9am and 6pm, the 4th day of observation lasted a shorter amount of time, from 11.30 am- 4 pm. The morning of the first day was spent observing the environment, and becoming familiar with the processes used in the pharmacy. The rest of the study period was spent collecting quantitative data of how long process stages take, and conducting research interviews with staff. One interview took place on the first day of observation, since a locum pharmacist was working, and they would not be returning later in the week. Following interviews took place on the remaining three days of observation. In addition, direct data was collected during this observation on how many items were included in prescriptions. This was done by inspection of completed walk-in prescriptions, and counting sorting the set of printed EPS prescriptions into alphabetical order.

A.2.2 Overview

The pharmacy is part of one of the larger pharmacy chains. The pharmacies opening hours were between 8.30 am to 6.30pm, and it dispensed between 1500-2000 items per week. There was a poster in the pharmacy titled: Power hour. When asked what this poster was about, it was explained that before Christmas it had been common practice that during the busy period of the day (3pm-4pm), 2 people would be assigned to the reception counter to help deal with customers. It was further explained that this practice no longer took place, due to a reduction in the number of staff.

Medicines were located all around this pharmacy. This meant that picking items could involve a lot of walking around, especially if more than 1 trip had to be made if a prescription contained multiple items. This was most noticeable when a dispenser was picking EPS prescriptions sequentially from the position of dispenser 5 in Figure 1.

The new medicine stock delivery arrived between 10.30 and 11. When it arrived, between 2 and 3 staff would be assigned to putting it away. On the third day of observation, this process was timed, and on that day it took three dispensers 20 minutes to put away the stock.

A.2.3 Spatial layout and typical working conditions

FIGURE A.2: Spatial layout and typical working positions in Pharmacy 2

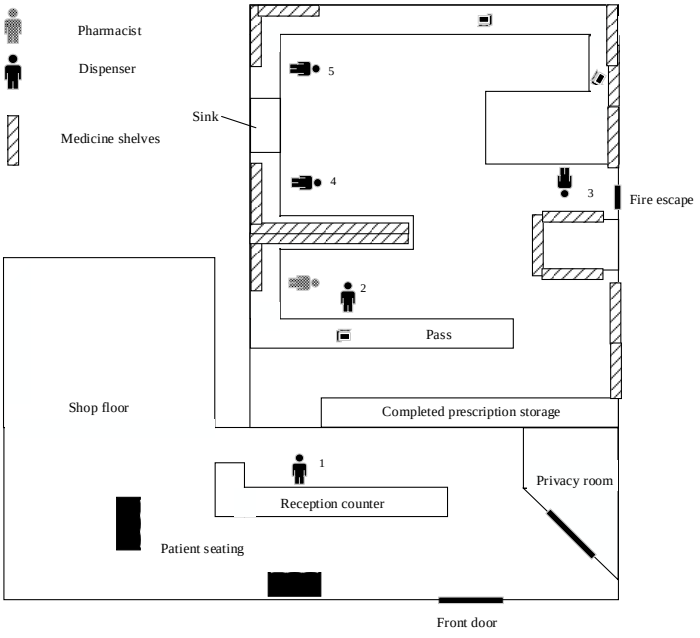


Figure one shows the spatial layout of the pharmacy and the typical working positions of members of staff. During the observation of this pharmacy, there was little time spent observing the counter. This was due to the dividing wall which separated the back of the pharmacy where dispensing took place, from the reception counter. Due to this geographical quirk of the pharmacy, there was little observation of the shop floor, handing out, and patient reception stages of the process. One member of staff would be working on the reception counter.

This is shown in Figure 1 by dispenser 1, although during observations this role was sometimes occupied by a pre-registration pharmacist. Working on the reception counter involved, from the perspective of an observer in the back pharmacy, bringing the prescriptions forms (walk-in or EPS) through in red baskets. If the

prescription was on the system as completed, the person on the counter would try and find the prescription themselves in the completed prescription storage.

The dispenser at positon 2 is located close to the pharmacist, and they would typically be working on processing walk-in prescriptions, and passing them over to the pharmacist. Dispensers 3 and 4 were putting together dossette trays. Dispenser 5 would be an ACT, who would typically be processing EPS prescriptions, or accuracy checking clinically checked prescriptions.

A.2.4 Staff

There was a relatively high turnover of staff in this pharmacy. In the 4 day observation period there were 3 different pharmacists, a pre reg pharmacist, 2 ACTs, 2 relief dispensers and 4 regular dispensers working, a total of 12 different staff members. The team size would change between 3 and 6 people. Upon arrival there would be a whole complement of 6 people present. The pre-reg pharmacist would finish at 4, with two more dispensers finishing between the hours of 4 and 6. Two dispensers and the pharmacist would end the shift between 6 and 6.30. There was a mixture of experience in the staff working at this pharmacy. The pre-registration pharmacist was on their 1st year of experience working in practice, others had been working for much longer.

A.2.5 Work flow and prioritisation of dispensing

This pharmacy seemed to have a focus on dossette trays. During the majority of the observation period there would be at least one, more commonly two, and sometimes three dispensers working on dispensing dossette trays. The only period where no-one was observed dispensing trays would be during the quiet periods of the day (12-2pm), when many staff had gone out for lunch. Otherwise dispensers dispensing trays was one of the constant background activities of this pharmacy.

The work flow of dispensing a dossette tray was as follows: pick the medicine boxes into a basket, assemble the dossette tray using the foam, plastic pods and plastic cover, fill each pod with the required tablets from the packs, stick a paper form inside the dossette tray and place the whole tray inside a plastic bag. These trays came in two types, weekly or monthly, where a monthly tray consisted of 4 weekly

trays. Each dispenser assigned to dispensing trays had their own patients who they had become familiar dispensing for. The dispensers mentioned that during staff training, the pharmacist had highlighted how dossette trays presented a much higher potential for near misses and errors, since tablet in each pod on the dossette tray could be incorrect. It was mentioned that in the month of January, there were approximately 100 near misses from dossette trays. Due to the bank holiday on January first, and the pharmacy being closed on Sundays. This equates to more than 4 tray near misses per day on average.

The workflow of dispensing walk-ins, followed the process seen in pharmacy 1. A dispenser generates labels, picks, applies labels, and hands over to a pharmacist who accuracy checks and hands over to patients. EPS prescriptions also followed a similar process.

A.2.6 Quantitative data collection

Timing the processes of dispensing stages focused mainly on the batch EPS processing, and also walk-in dispensing if there was no EPS dispensing going on. While timing a member of staff doing EPS prescriptions, they quickly realised the researcher was peering at the prescription to count the number of items. To help the data collection process, this dispenser began calling out how many items were in each prescription. This practice spread to the rest of the pharmacy as I observed other people working. The dispenser who began the practice, would encourage other members of staff to call out in a similar way, how many items were in each prescription. This eased the process of data collection related to how many items were in prescriptions passing through each stage.

Although the data collection was easier in one sense, due to staff calling out how many items were in each prescription. It was made harder in other ways. A principle way, was that the dividing wall in the pharmacy meant that only one area of the pharmacy could be observed at once. The researcher took the decision that most activity was occurring in the back of the pharmacy, and therefore data collection should focus on activities occurring there. This meant that no data was collected on the patient reception or handing over/storing stages of the process.

Another difficulty arose from the fact that this pharmacy was so heavily focused on dispensing dossette trays. The trays take a very long time to dispense, it was

mentioned a monthly tray could take 20 - 30 minutes, so they were not deemed to be suitable to time in the same data set as stages of dispensing conventional prescriptions. However, there were considerable periods of time in the pharmacy where no time-able activities were occurring.

A.2.7 Items

The number of items in a prescription during this observation was defined as the number of distinct items on a prescription form. Staff were made aware that this was the case when they were asked to call out the number of items.

A.2.8 Splitting packs

The lengthy process of splitting packs was again observed at this pharmacy. It was less noticeable due to the lower number of walk-ins and EPS prescriptions. Packs sometimes had to be split during dossette dispensing, however since the researcher was not directly timing this activity splitting packs seemed much less prevalent.

A.2.9 Wastage

When asked about wastage in the pharmacy, the pre-registration pharmacist looked into how many prescriptions had been unclaimed the previous week. It turned out that 18 notifications had been sent out to patients the previous week.

A.2.10 Mood and atmosphere

This pharmacy was quitter than pharmacy 1. This could have been due to the higher turnover of staff, and potential unfamiliarity with each other. Although the mood lightened in the third and fourth day. There were no concerns that the research was an internal company exercise at this pharmacy, and the staff all seemed mostly comfortable with the presence of a researcher. Notably, another pharmacy student on work experience also came in to the pharmacy on the fourth day of observation.

A.2.11 Interviews

Not all members of staff working in this pharmacy were interviewed. 5 members of staff were interviewed in this pharmacy, 3 pharmacists, 1 ACT and a dispenser.

A.2.12 Relief staff

During the observation there were two relief staff working. They both worked morning shifts, being there already when the research arrived on days 3 and 4 and finishing at 1pm. They were assigned to work on processing EPS prescriptions, due to difficulty posed at dispensing trays. However, processing EPS prescriptions appeared difficult for staff unused to the layout of this pharmacy. They would walk around, attempting to locate items, and often have to ask other member of staff where things were. Furthermore, a number of near misses were observed in the items picked and labelled by the relief worker. On day three in particular, two or three near misses related to wrong dose, or a split pack were spotted by the pharmacist.

A.2.13 NHS England contract visit

On the third day of observation that was an NHS contract visit. The visit was designed to ensure that processes in the pharmacy were being carried out safely. It took up twenty minutes of the pharmacists' time, as they had to show the visitor various things in the pharmacy.

A.3 Observational profile - Pharmacy 3

A.3.1 Introduction

The observation took place over 5 days, Tuesday 15th - Wednesday 16th May, Tuesday 22nd - Wednesday 23rd, and Thursday 21st, where each day consisted of approximately 7 hours of observation between the hours of 9am and 4pm. The researcher paid a preliminary visit to the pharmacy on Tuesday the 8th of May, however the study was postponed at that time due to a key member of staff being away.

The researcher's time in the pharmacy was spent recording the time taken to complete process stages, and conducting recorded interviews with members of staff. The interviews were done on the first four days of observation, and they were either completed in the outside area between the main pharmacy store, or in the shed out the back. The objective of the observation was to gain an understanding of the process of dispensing in community pharmacies.

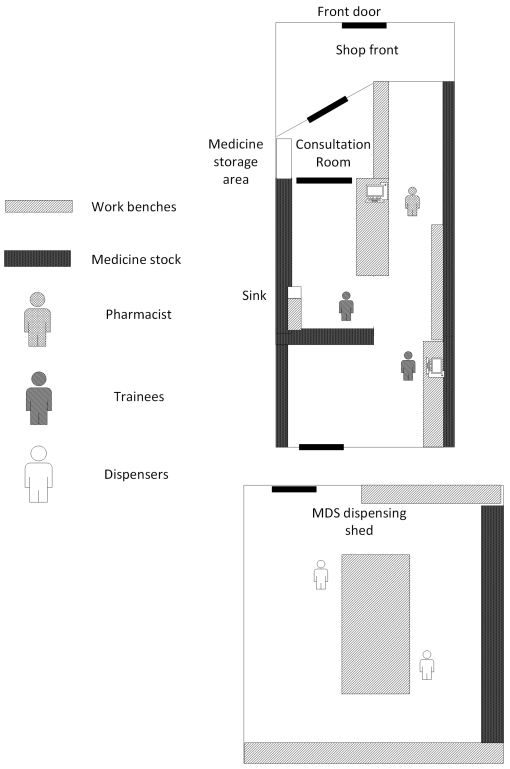
A.3.2 Spatial layout and typical working conditions

Figure one shows the spatial layout of the pharmacy and some typical working positions of members of staff. Positions were a lot loss structured than in other pharmacies. If present in the team on a given day, there were two dispensers in the shed working on MDS trays. 1 or 2 pre-reg students would be in the dispensary, dealing with patients, picking prescriptions, or filling out the order form. The pharmacist would split their time between the shed and the main pharmacy.

A.3.3 Overview

This pharmacy was split into two sections. The main area is shown at the top of Figure 1. This main section of the pharmacy is where all non-MDS prescriptions were dispensed. The secondary section of the pharmacy was the shed, which was used for dispensing two types of patient adherence medicine trays. This section of the pharmacy was in use much less than the main part. Of the days observed in the pharmacy, the MDS shed was only observed being used for dispensing on

FIGURE A.3: Spatial layout and typical working positions in Pharmacy 3



2 of them, and then only during the mornings. The shed also served as a lunch room which was used by staff for lunch breaks. There was a microwave, kettle and computer in there which members of staff would use during their hours off. At other times it was observed in use as a training room, where the pharmacist would pose formulation questions to the trainees.

The pharmacy is an independently owned store, run by the two pharmacists who both were observed working. They also own another store, in a different part of the city. They had owned the stores for a number of years, and members of staff had all been working in the branch for a long time. There was no emergency fill in staff as had been seen at other pharmacies, there were a fixed number of staff on the payroll, and if people had holidays or were sick, then there was no one to call up to fill empty places.

The pharmacy seemed to have stronger interpersonal relationships with the patients that visited than had been previously seen. Notably, almost all patients who were on addiction treatments were greeted in a friendly manner by name every time they came in, and would engage the staff in friendly conversation rather than waiting in silence.

The pharmacies opening hours were between 9am to 6pm, on all days apart from Thursdays, when the pharmacy closed at 1pm. The pharmacy had two computers in the main section, which were used to print labels for prescriptions, and two more in the back which seemed to be for pastoral activities.

A.3.4 Staff

On each day of observation, the team of people working in the pharmacy would fluctuate between 3 and 6 members of staff. Upon arriving in the morning at 9am, 3 - 6, members of staff would be present. Other members would all arrive within the next half an hour, so the full team for the day would be present by 9.30. One or two members of staff (the ones working in the MDS shed) would then finish at 1 pm, depending on whether they had been in for the day.

A.3.5 Quantitative data collection

This process took much longer than it had done previously. This was due to the generally low levels of dispensary activity during the day. For example, there were long periods during days where no dispensing was being done. Not even MDS dispensing. Staff also were commenting (in a humorous manner) how if I kept on coming I would put the pharmacy out of business, since many of the days when I went to observe were very quiet.

A.3.6 Split packs

Having seen how split packs can cause delays in the dispensing process, it was decided that the number of split packs would be counted during these observations. Over the course of the 6 days, 32 prescriptions containing split packs were observed to be dispensed.

It should be stated that the researcher was not present in the main pharmacy section of the pharmacy for all of the observation period, and staff were not asked to indicate to the researcher when there was a split pack. So it is possible that the number of prescriptions containing split packs exceeded 32.

A.3.7 Mood and atmosphere

The pharmacy had a relaxed atmosphere, although there was sometimes frustration with the trainees making mistakes or not understanding things. It was unclear how much of the displays of frustration were designed to exaggerate a teaching point, or in fact genuine displays of frustration. The dispensers were generally quiet. Whereas in other pharmacies the dispensers would chat to the researcher quite a bit, and ask me about what I'm doing, in this pharmacy it was only really the owners, and sometimes the pre-reg who would interact with me on a regular basis. There was however less of a feeling that the dispensers felt like cogs in a large machine than had been seen elsewhere. They seemed generally happier and more upbeat, despite not engaging with me in particular.

A.3.8 Day profile

On most days of observation, activity in the pharmacy followed a very similar pattern of activity. In the morning, there would be a lot of dispensing as around 20-40 scripts (estimate) were dispensed. This would take around an hour to complete, during or sometime after which, the delivery of new stock would arrive. Everyone except the pharmacist would help to put the stock away, which took around 15 minutes each day. After the morning spike of activity, there would generally be quite long quiet periods in terms of prescription completion. Another burst of activity would occur later in the day around 3pm, as more scripts were printed off. The activity profile was different on the Thursday observation since all the activity had to be condensed into a shorter day, and dispensing activity was almost constant throughout the morning.

A.3.9 Interviews

All but two members of staff agreed to take part in research interviews, so out of the 8 members of staff who were working over the days of observation, interviews were completed with 6.

The attitude to interviews was very different in this pharmacy. One member of staff completed an interview while dispensing an MDS tray. The pharmacist and another dispenser were in the room at the time. Other interviews took place

outside, and it seemed that some staff were reluctant or too busy to give me their undivided attention. For example, during one interview the interviewee was putting up a glasses stand.

A.3.10 Customer acknowledgement

The customer acknowledgement was very fast in this store. As soon as a customer walked in the door someone would be checking to make sure they were being seen to. There was also a loud beeper attached to the door, such that every time the door opened a loud noise was played in the pharmacy to alert staff to the fact that a customer had arrived.

There were no period where customers were seen waiting at the counter to be seen for extended periods of time. Notably, this didn't seem like a top down driven policy. All members of staff took part in the duty of ensuring patients were being seen. This might involve pointing out to someone not occupied that someone was waiting, or calling into the adjacent room that the counter needed to be attended.

A.3.11 Discrepancies between current assumptions and observations

The largest discrepancy in this store was the waiting time for patients, and the stores approach to dispensing EPS prescriptions. Their policy was to print EPS prescriptions throughout the day and work on them as they're printed out, rather than the batch production day by day methods seen before. Waiting times were in general much lower in this store than previously observed. This may have been due to the lower footfall, or more regular and experienced staff working. Analysing the quantitative timings taken during observations should give a good indication as to the cause of the lower waiting times. If task timings are similar, this may indicate that a lower volume of work was allowing the lower waiting times rather than more efficient individual staff.

One of the pharmacists working during observations did not bag up prescriptions after completing the final accuracy check, which had in all other cases been observed to be the case. This may have been due to an injury during the observation period.

A further discrepancy was observed during an especially busy period when a large number of scripts were being completed. A dispenser was printing off EPS scripts, printing the labels and placing them into baskets, then a pre-reg would pick items for the prescriptions into the baskets, and place the baskets onto the big desk in the middle of the main pharmacy area, where a second pre-reg was applying labels and sorting out any split packs. This process had multiple individuals completing the stages of dispensing on the same scripts. However it was only observed once during an extremely busy period.

The pharmacist was often outside the main dispensing room during the day. This meant that after prescriptions had been fully dispensed, the dispenser or pre-reg would have to take the prescription out back to the shed, to get it checked by the pharmacist. This must have slowed the process, since the travel time between the two areas was not insignificant.

Another notable observation was that the size of the delivery that needs to put away each day is directly related to how many prescriptions were dispensed the day before. If the model were to be extended to a weekly run time, this could factor into simulations.

A.3.12 Training

A lot of the pharmacist's time was dedicated to teaching and conversing with the trainee pre-regs. Especially on the quieter days, much time would be dedicated to this. Where previously pre-regs were not noticeably different to other members of staff, here there was a large amount of time dedicated to questioning and stretching their knowledge throughout the day.

A.3.13 Smoke breaks

Most of the dispensers working in the pharmacy were smokers, and they were allowed to take smoke breaks throughout the day. This was seen twice. The group of smokers, if working on the same day would take the break all together for around 5 minutes. They would use the area outside between the two dispensing areas to smoke.

A.4 Observational profile - Pharmacy 4

The observation took place over 5 days, Monday 11th to Friday the 15th of June 2018, where each day consisted of approximately 7 hours of observation between the hours of 9am and 4pm each day.

The observation period was spent timing the time taken to complete process stages, and conducting recorded interviews with members of staff. Patient privacy rooms were used to conduct recorded interviews with all members of staff.

A.4.1 Spatial layout and typical working conditions

FIGURE A.4: Spatial layout and typical working positions in Pharmacy 4

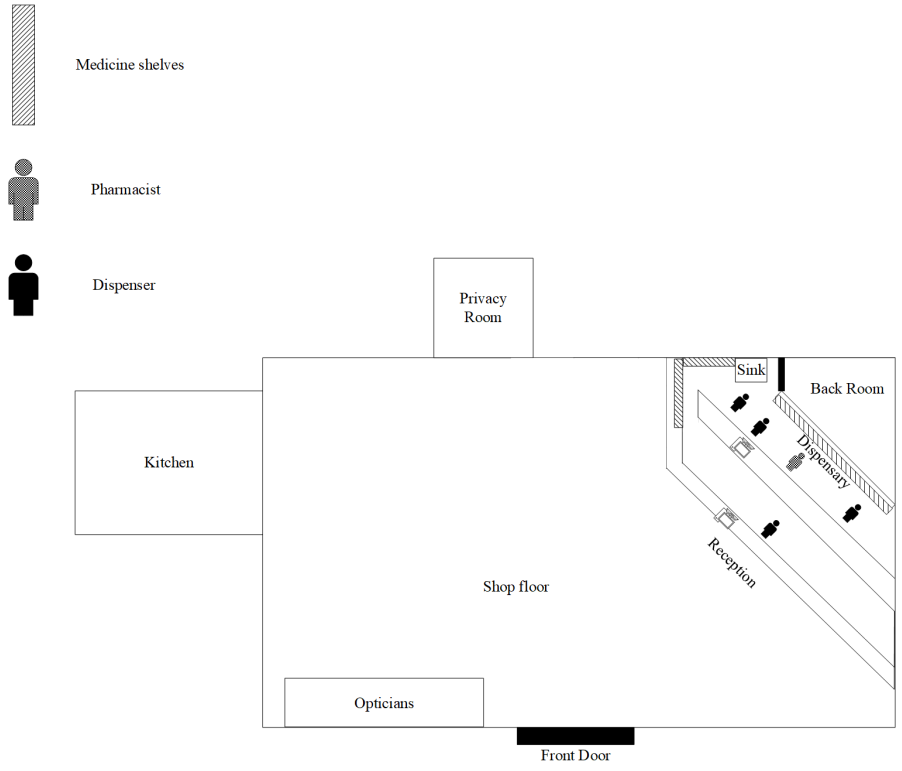


Figure A.4 shows the spatial layout of Pharmacy D. The dispensary is located in one corner of the shop with a large floor space. This pharmacy was by far the largest of the 4 sites. Staff would typically be working in the main dispensary area, dispensing prescriptions.

A.4.2 Overview

The shelving arrangements in Pharmacy D were very compact. There was a large row of multi level shelving located at the back of the dispensary where the majority of the medicines were stored, with a small amount of stock stored in a corned next to the sink. The shelving seemed to be useful, in that the dispensers didn't have to walk too far to collect medication in the majority of cases. However, on occasion there was an issue with having to reach into the top level shelves. The shelves were tall, and the top layer could not be reached without the aid of a short step ladder. The top shelves were designed to be used for storage so that the step ladder would not need to be used regularly, although it was observed that the step ladder was needed more regularly than expected if this were the case.

The pharmacist is an independent store owned by a single pharmacist. Unlike Pharmacy C, very few MDS prescriptions were dispensed in this store. The owner mentioned similar issues with staff shortages, since they did not have a large pool of people to draw on if a member of staff was absent. If the pharmacist couldn't work then a locum pharmacist would be hired, but otherwise there was little staff cover.

The pharmacy was located in a shopping complex so it was common for patients to come into the pharmacy and hand in their prescription, do some shopping in the surrounding area, and come pick up their prescription after finishing their shop. This meant it was uncommon for large groups of patients to be waiting in the pharmacy for their prescriptions.

A.4.3 Staff experience

There were many relatively inexperienced or trainee staff working in Pharmacy D. The two dispensers were young dispensing apprentices, who had begun working in the pharmacy after finishing school. One had recently finished the full training, while the other was still in a training period. Additionally, one of the pre-registration pharmacists was coming to the end of their six month placement working at the pharmacy. Their last day working at the pharmacy took place of the second day of the observation. On the following day, a new trainee came in to the pharmacy , to begin their 6 month placement by having their first ever

day working in a community pharmacy. The difference in experience and understanding between the two pre registration pharmacists was noticeable. It impacted how the dispensing process was performed, i.e. slower with a need to frequently ask questions, and how they interacted with customers. The experienced trainee would have an idea of what patients were after whilst engaging with them, while the newer trainee sometimes misheard or didn't understand patient's requests.

A.4.4 Day profile

The pharmacy was busiest in the morning between 9-10am and in the afternoon 3-5pm. EPS prescriptions were printed and completed as soon as they were sent to the pharmacy.

A.4.5 Interviews

Interviews were carried out with all the members of staff at Pharmacy D. The spacious privacy room or the smaller back room behind the dispensary were used to conduct the interviews.

A.4.6 Pharmacist restocking shelves

During the observation period, there was one occasion where the pharmacist of this store was seen restocking the shelves. This was the first and only time that a pharmacist was observed contributing towards the restocking of shelves in the entire observation period of all 4 pharmacies. They did no restock a whole box of stock, but they did contribute to putting some of it away.

A.4.7 Mood and atmosphere

The atmosphere of Pharmacy D had a quiet and clinical character. The staff wouldn't regularly engage in conversation on topics unrelated to their work. They were all friendly and willing to engage in conversation when prompted.

Appendix B

Example Raw Data

B.1 Example Raw Data

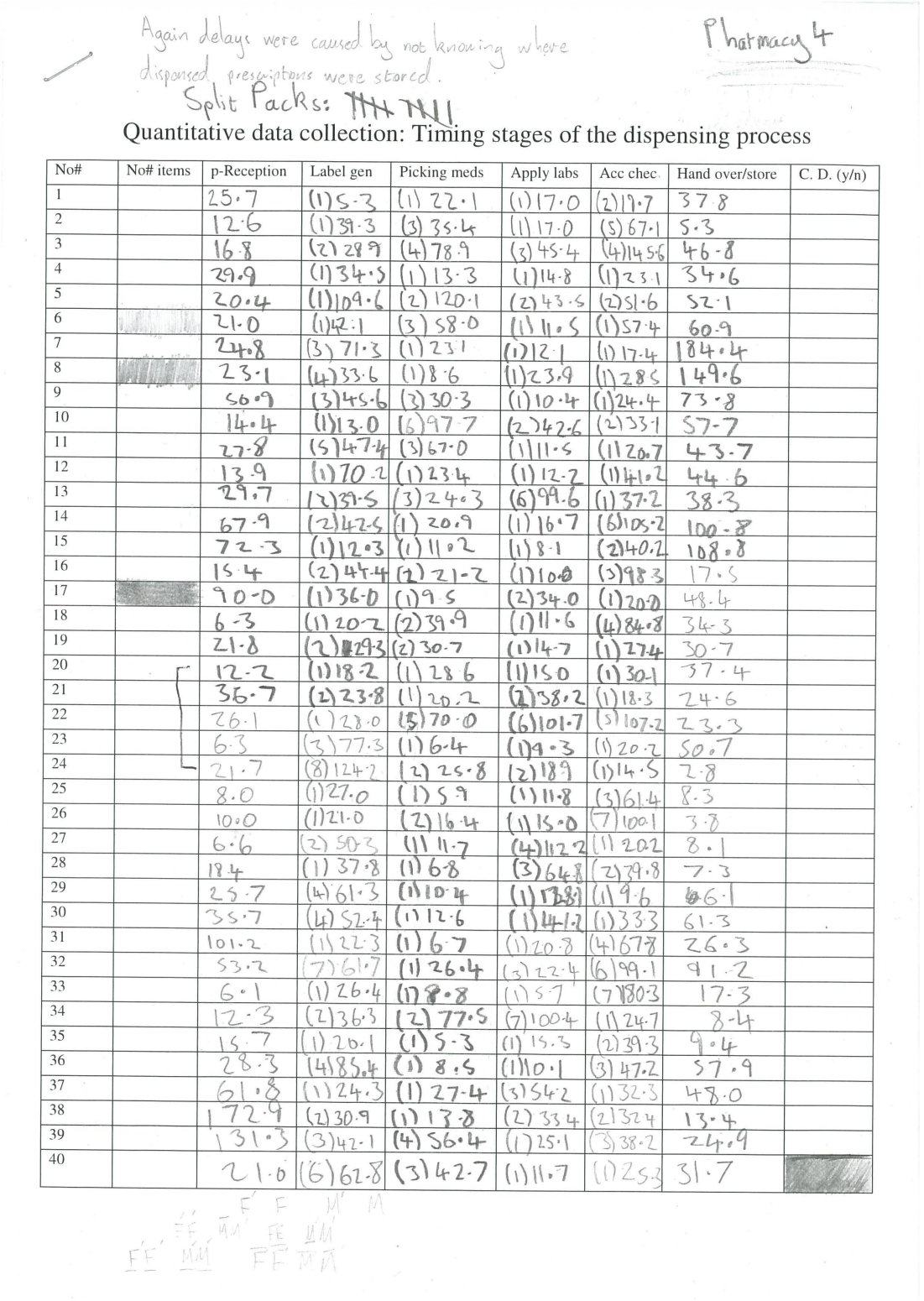
This appendix will display an example of a set of raw data collected from one of the 4 on-site studies. The data shown in this appendix was collected from Pharmacy 4.

The data is recorded in 6 columns titled: p-Reception, Label gen, Picking meds, Apply labs, Acc chec, and Handover/Store. These correspond to each of the main stages of dispensing.

The two stages p-Reception and Handover/Store are recorded as single variable timing data, whereas the 4 other categories are recorded as two variable data. The second variable is written in brackets, and it indicates the number of items in the prescription that was passing through the stage.

Some notes related to what was going on inside the pharmacy during the observation period are written in the borders of the recording sheets.

FIGURE B.1: Raw timing data from Pharmacy 4: Page 1



Again delays were caused by not knowing where dispensed presciptions were stored.

Split Packs: ||||-|||

Pharmacy 4

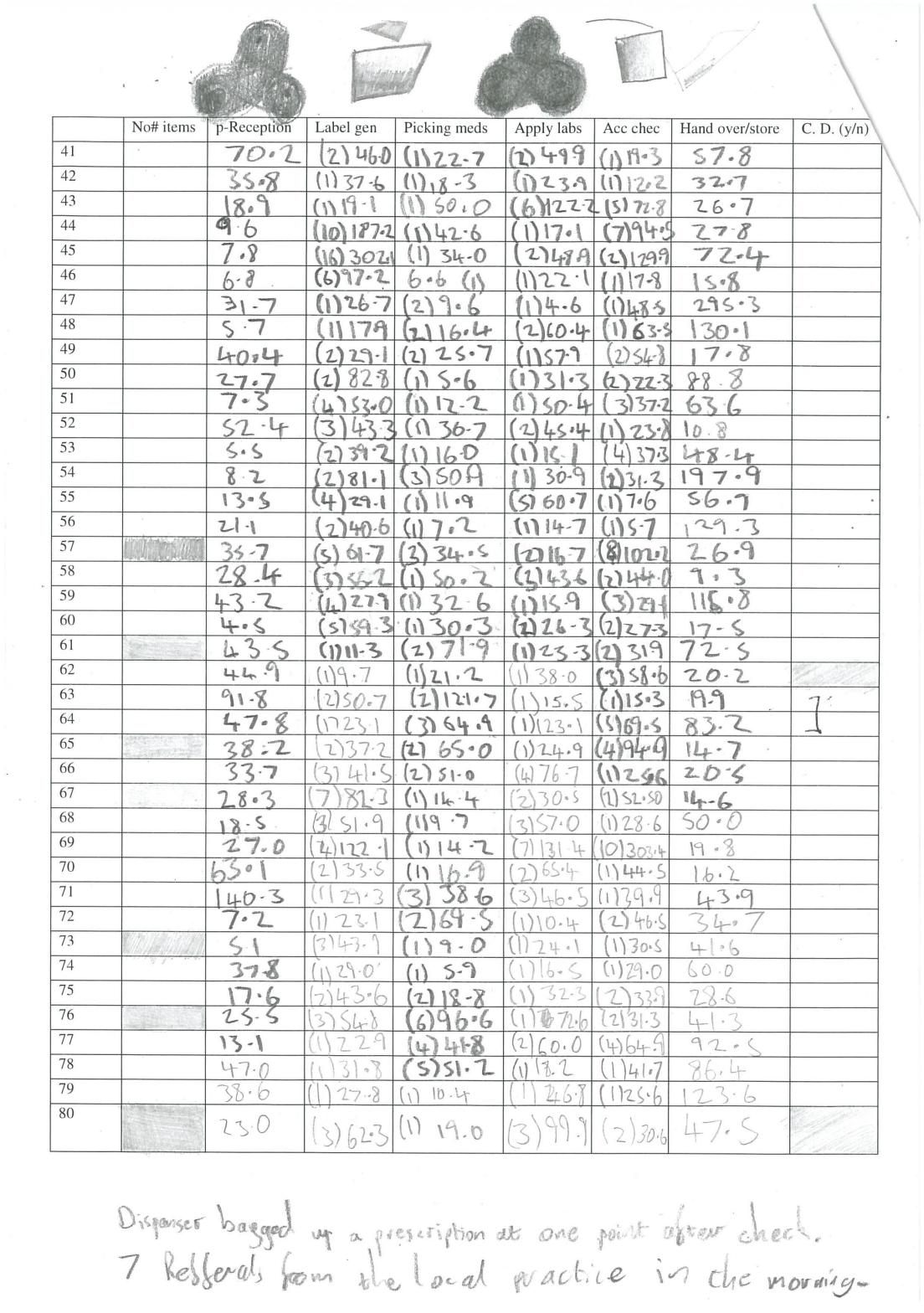
Quantitative data collection: Timing stages of the dispensing process

No#	No# items	p-Reception	Label gen	Picking meds	Apply labs	Acc chec	Hand over/store	C. D. (y/n)
1		25·7	(1) 5·3	(1) 22·1	(1) 17·0	(2) 19·7	37·8	
2		12·6	(1) 39·3	(3) 35·4	(1) 17·0	(5) 67·1	5·3	
3		16·8	(2) 28·9	(4) 78·9	(3) 45·4	(4) 114·56	46·8	
4		29·9	(1) 34·5	(1) 13·3	(1) 14·8	(1) 23·1	34·6	
5		20·4	(1) 109·6	(2) 120·1	(2) 43·5	(2) 51·6	52·1	
6		21·0	(1) 42·1	(3) 58·0	(1) 11·5	(1) 57·4	60·9	
7		24·8	(3) 71·3	(1) 23·1	(1) 12·1	(1) 17·4	184·4	
8		23·1	(4) 33·6	(1) 8·6	(1) 23·9	(1) 28·5	149·6	
9		56·9	(3) 45·6	(3) 30·3	(1) 10·4	(1) 24·4	73·8	
10		14·4	(1) 13·0	(6) 97·7	(2) 42·6	(2) 33·1	57·7	
11		27·8	(5) 47·4	(3) 67·0	(1) 11·5	(1) 20·7	43·7	
12		13·9	(1) 70·2	(1) 23·4	(1) 12·2	(1) 41·2	44·6	
13		29·7	(2) 39·5	(3) 24·3	(6) 99·6	(1) 37·2	38·3	
14		67·9	(2) 42·5	(1) 20·9	(1) 16·7	(6) 105·2	100·8	
15		72·3	(1) 12·3	(1) 11·2	(1) 8·1	(2) 40·2	108·8	
16		15·4	(2) 44·4	(2) 21·2	(1) 10·0	(3) 98·3	17·5	
17		90·0	(1) 36·0	(1) 9·5	(2) 34·0	(1) 20·0	48·4	
18		6·3	(1) 20·2	(2) 39·9	(1) 11·6	(4) 84·8	34·3	
19		21·8	(2) 129·3	(2) 30·7	(1) 14·7	(1) 27·4	30·7	
20		12·2	(1) 18·2	(1) 28 6	(1) 15 0	(1) 30·1	37·4	
21		36·7	(2) 23·8	(1) 20·2	(2) 38·2	(1) 18·3	24·6	
22		26·1	(1) 28·0	(5) 70·0	(6) 101·7	(5) 107·2	23·3	
23		6·3	(3) 77·3	(1) 6·4	(1) 9·3	(1) 20·2	50·7	
24		21·7	(8) 124·2	(2) 25·8	(2) 18 9	(1) 14·5	2·8	
25		8·0	(1) 27·0	(1) 5 9	(1) 11·8	(3) 61·4	8·3	
26		10·0	(1) 21·0	(2) 16·4	(1) 15·0	(7) 100·1	3·8	
27		6·6	(2) 50·3	(1) 11·7	(4) 112·2	(1) 20·2	8·1	
28		18·4	(1) 37·8	(1) 6·8	(3) 64·8	(2) 39·8	7·3	
29		25·7	(4) 61·3	(1) 10·4	(1) 12·8	(1) 9·6	66·1	
30		35·7	(4) 52·4	(1) 12·6	(1) 41·2	(1) 33·3	61·3	
31		101·2	(1) 22·3	(1) 6·7	(1) 20·8	(4) 67·8	26·3	
32		53·2	(7) 61·7	(1) 26·4	(3) 22·4	(6) 99·1	91·2	
33		6·1	(1) 26·4	(1) 8·8	(1) 5·7	(7) 80·3	17·3	
34		12·3	(2) 36·3	(2) 77·5	(7) 100·4	(1) 24·7	8·4	
35		15·7	(1) 20·1	(1) 5·3	(1) 15·3	(2) 39·3	9·4	
36		28·3	(4) 85·4	(1) 8·5	(1) 10·1	(3) 47·2	57·9	
37		61·8	(1) 24·3	(1) 27·4	(3) 54·2	(1) 32·3	48·0	
38		172·9	(2) 30·9	(1) 13·8	(2) 33·4	(2) 32·4	13·4	
39		131·3	(3) 42·1	(4) 56·4	(1) 25·1	(3) 38·2	24·9	
40		21·6	(6) 62·8	(3) 42·7	(1) 11·7	(1) 25·3	31·7	

F F M M

FF MM FF MM FF MM

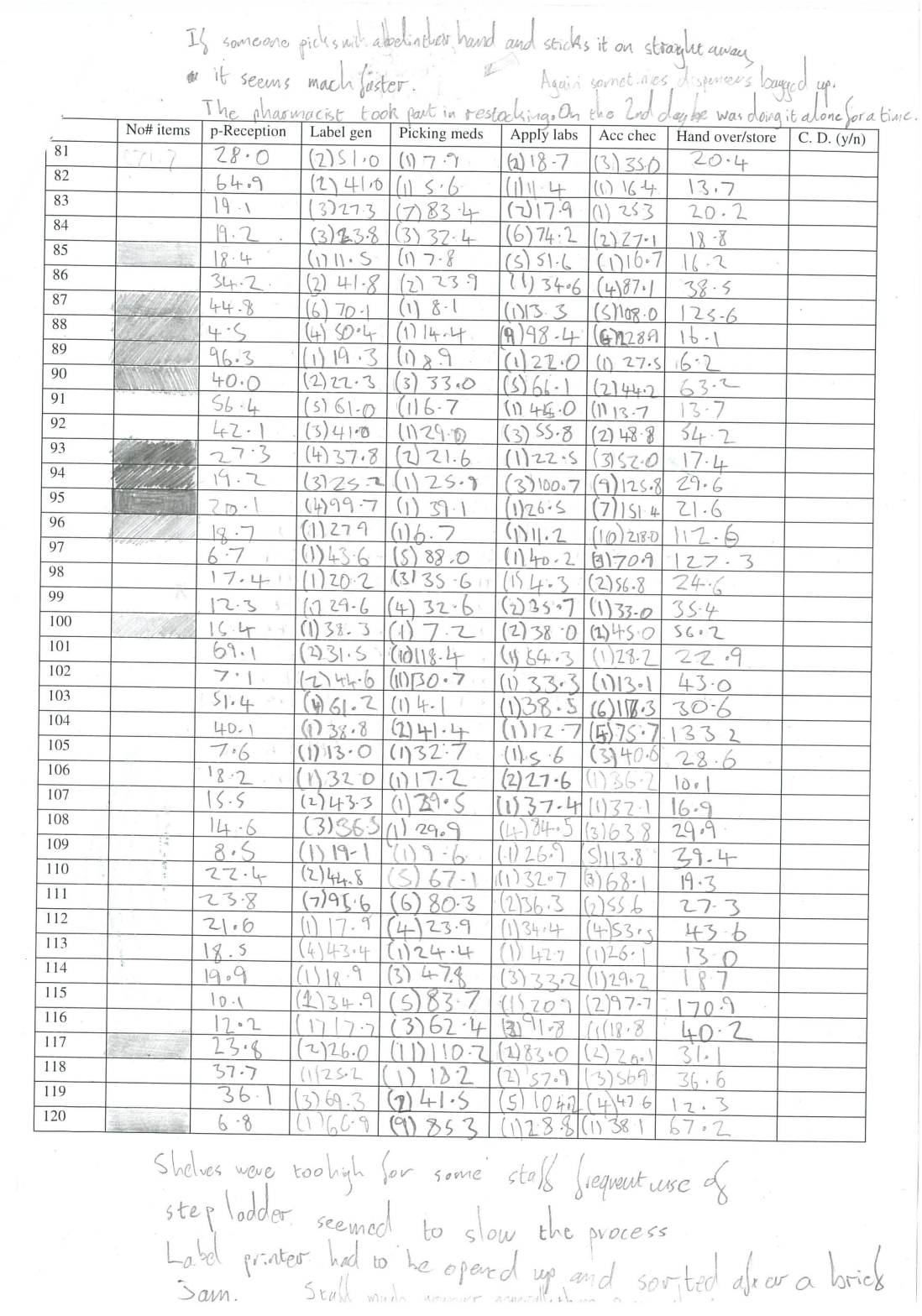
FIGURE B.2: Raw timing data from Pharmacy 4: Page 2



	No# items	p-Reception	Label gen	Picking meds	Apply labs	Acc chec	Hand over/store	C. D. (y/n)
41		70·2	(2) 46·0	(1) 22·7	(2) 49·9	(1) 19·3	57·8	
42		35·8	(1) 37·6	(1) 18·3	(1) 23·9	(1) 12·2	32·7	
43		18·9	(1) 19·1	(1) 50·0	(6) 122·2	(5) 72·8	26·7	
44		9·6	(10) 187·2	(1) 42·6	(1) 17·1	(7) 94·5	27·8	
45		7·8	(16) 302·1	(1) 34·0	(2) 48·9	(2) 129·9	72·4	
46		6·8	(6) 97·2	6·6 (1)	(1) 22·1	(1) 17·8	15·8	
47		31·7	(1) 26·7	(2) 9·6	(1) 4·6	(1) 48·5	295·3	
48		5·7	(1) 17·9	(2) 16·4	(2) 60·4	(1) 63·5	130·1	
49		40·4	(2) 29·1	(2) 25·7	(1) 57·9	(2) 54·8	17·8	
50		27·7	(2) 82·8	(1) 5·6	(1) 31·3	(2) 22·3	88·8	
51		7·3	(4) 53·0	(1) 12·2	(1) 50·4	(3) 37·2	63·6	
52		52·4	(3) 43·3	(1) 36·7	(2) 45·4	(1) 23·8	10·8	
53		5·5	(2) 39·2	(1) 16·0	(1) 15·1	(4) 37·3	48·4	
54		8·2	(2) 81·1	(3) 50·9	(1) 30·9	(2) 31·3	197·9	
55		13·5	(4) 29·1	(1) 11·9	(5) 60·7	(1) 7·6	56·7	
56		21·1	(2) 40·6	(1) 7·2	(1) 14·7	(1) 5·7	129·3	
57		35·7	(5) 61·7	(3) 34·5	(2) 16·7	(8) 102·2	26·9	
58		28·4	(3) 56·2	(1) 50·2	(3) 43·6	(2) 44·0	9·3	
59		43·2	(4) 27·9	(1) 32·6	(1) 15·9	(3) 29·1	116·8	
60		4·5	(5) 59·3	(1) 30·3	(2) 26·3	(2) 27·3	17·5	
61		43·5	(1) 11·3	(2) 71·9	(1) 25·3	(2) 31·9	72·5	
62		44·9	(1) 9·7	(1) 21·2	(1) 38·0	(3) 58·6	20·2	
63		91·8	(2) 50·7	(2) 121·7	(1) 15·5	(1) 15·3	19·9	1
64		47·8	(1) 23·1	(3) 64·9	(1) 23·1	(5) 69·5	83·2	
65		38·2	(2) 37·2	(2) 65·0	(1) 24·9	(4) 94·9	14·7	
66		33·7	(3) 41·5	(2) 51·0	(4) 76·7	(1) 29·6	20·5	
67		28·3	(7) 82·3	(1) 14·4	(2) 30·5	(1) 52·50	14·6	
68		18·5	(3) 51·9	(1) 19·7	(3) 57·0	(1) 28·6	50·0	
69		27·0	(4) 122·1	(1) 14·2	(7) 131·4	(10) 303·4	19·8	
70		63·1	(2) 33·5	(1) 16·9	(2) 65·4	(1) 44·5	16·2	
71		140·3	(1) 21·3	(3) 38·6	(3) 46·5	(1) 39·9	43·9	
72		7·2	(1) 23·1	(2) 69·5	(1) 10·4	(2) 46·5	34·7	
73		5·1	(3) 43·9	(1) 9·0	(1) 24·1	(1) 30·5	41·6	
74		37·8	(1) 29·0	(1) 5·9	(1) 16·5	(1) 29·0	60·0	
75		17·6	(2) 43·6	(2) 18·8	(1) 32·3	(2) 33·9	28·6	
76		25·5	(3) 54·8	(6) 96·6	(1) 72·6	(2) 31·3	41·3	
77		13·1	(1) 22·9	(4) 44·8	(2) 60·0	(4) 64·9	92·5	
78		47·0	(1) 31·8	(5) 51·2	(1) 18·2	(1) 41·7	86·4	
79		38·6	(1) 27·8	(1) 10·4	(1) 24·8	(1) 25·6	123·6	
80		23·0	(3) 62·3	(1) 19·0	(3) 99·9	(2) 30·6	47·5	

Dispenser bagged up a prescription at one point after check.
7 Referrals from the local practice in the morning.

FIGURE B.3: Raw timing data from Pharmacy 4: Page 3

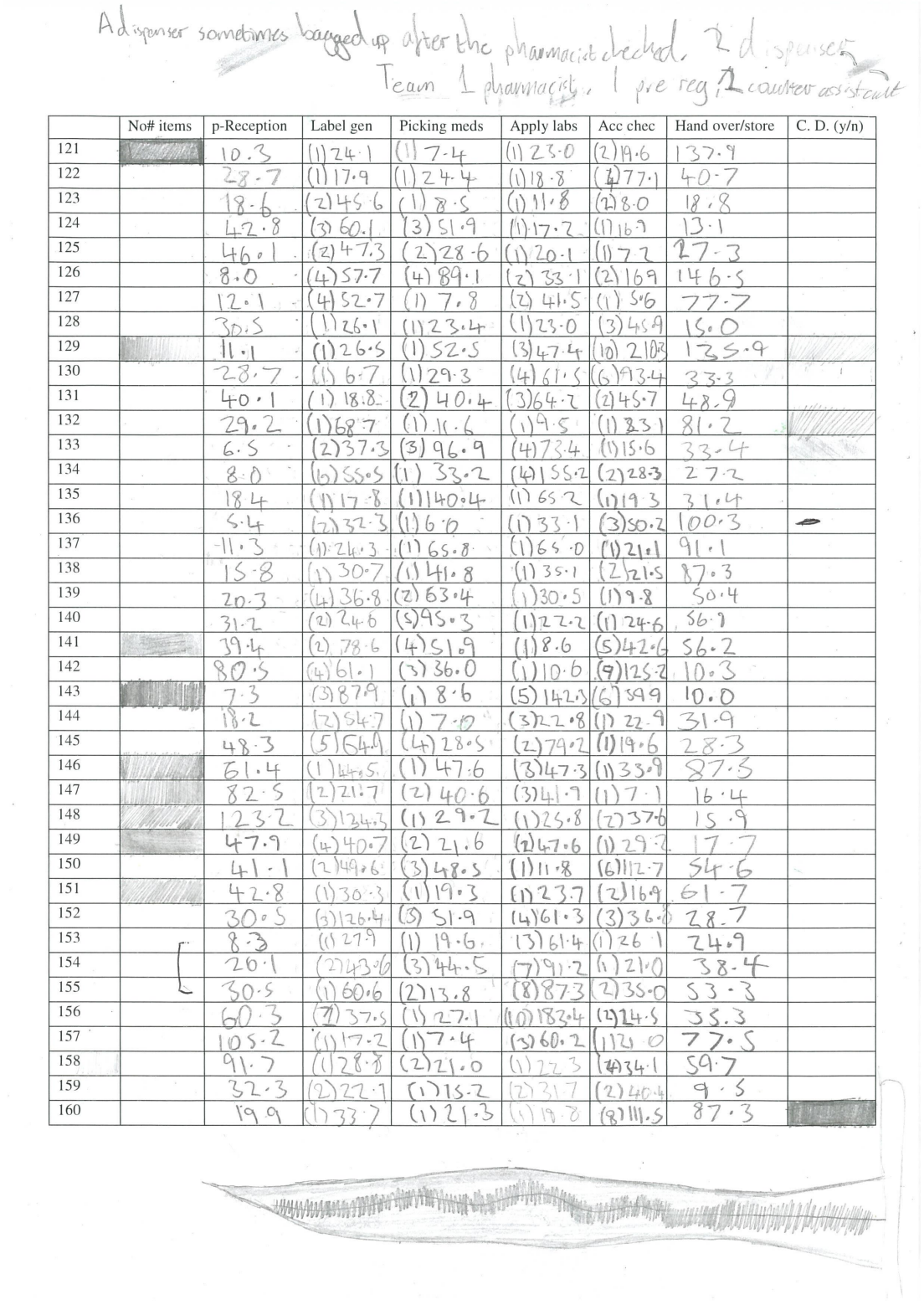


If someone picks with abdominal hand and sticks it on straight away it seems much faster. Again sometimes dispensers bagged up.
The pharmacist took part in restocking. On the 2nd day he was doing it alone for a time.

	No# items	p-Reception	Label gen	Picking meds	Apply labs	Acc chec	Hand over/store	C. D. (y/n)
81		28·0	(2) 51·0	(1) 7·9	(2) 18·7	(3) 35·0	20·4	
82		64·9	(2) 41·0	(1) 5·6	(1) 11·4	(1) 16·4	13·7	
83		19·1	(3) 27·3	(7) 83·4	(7) 17·9	(1) 25·3	20·2	
84		19·2	(3) 23·8	(3) 32·4	(6) 74·2	(2) 27·1	18·8	
85		18·4	(1) 11·5	(1) 7·8	(5) 51·6	(1) 16·7	16·2	
86		34·2	(2) 41·8	(2) 23·9	(1) 34·6	(4) 87·1	38·5	
87		44·8	(6) 70·1	(1) 8·1	(1) 13·3	(5) 108·0	125·6	
88		4·5	(4) 50·4	(1) 14·4	(9) 98·4	(6) 128·9	16·1	
89		96·3	(1) 19·3	(1) 8·9	(1) 22·0	(1) 27·5	16·2	
90		40·0	(2) 22·3	(3) 33·0	(5) 66·1	(2) 44·2	63·2	
91		56·4	(5) 61·0	(1) 16·7	(1) 45·0	(1) 13·7	13·7	
92		42·1	(3) 41·0	(1) 29·0	(3) 55·8	(2) 48·8	34·2	
93		27·3	(4) 37·8	(2) 21·6	(1) 22·5	(3) 52·0	17·4	
94		19·2	(3) 25·2	(1) 25·9	(3) 100·7	(9) 125·8	29·6	
95		20·1	(4) 99·7	(1) 39·1	(1) 26·5	(7) 151·4	21·6	
96		18·7	(1) 27·9	(1) 6·7	(1) 11·2	(10) 218·0	112·6	
97		6·7	(1) 43·6	(5) 88·0	(1) 40·2	(3) 70·9	127·3	
98		17·4	(1) 20·2	(3) 35·6	(1) 54·3	(2) 56·8	24·6	
99		12·3	(1) 29·6	(4) 32·6	(2) 35·7	(1) 33·0	35·4	
100		15·4	(1) 38·3	(1) 7·2	(2) 38·0	(2) 45·0	56·2	
101		69·1	(2) 31·5	(10) 118·4	(1) 84·3	(1) 28·2	22·9	
102		7·1	(2) 44·6	(11) 130·7	(1) 33·3	(1) 13·1	43·0	
103		51·4	(4) 61·2	(1) 4·1	(1) 38·5	(6) 118·3	30·6	
104		40·1	(1) 38·8	(2) 41·4	(1) 12·7	(4) 75·7	133·2	
105		7·6	(1) 13·0	(1) 32·7	(1) 5·6	(3) 40·6	28·6	
106		18·2	(1) 32·0	(1) 17·2	(2) 27·6	(1) 36·2	10·1	
107		15·5	(2) 43·3	(1) 29·5	(1) 37·4	(1) 32·1	16·9	
108		14·6	(3) 36·3	(1) 29·9	(4) 84·5	(3) 63·8	29·9	
109		8·5	(1) 19·1	(1) 9·6	(1) 26·9	(5) 113·8	39·4	
110		22·4	(2) 44·8	(5) 67·1	(1) 32·7	(3) 68·1	19·3	
111		23·8	(7) 95·6	(6) 80·3	(2) 36·3	(2) 55·6	27·3	
112		21·6	(1) 17·9	(4) 23·9	(1) 34·4	(4) 53·5	43·6	
113		18·5	(4) 43·4	(1) 24·4	(1) 42·7	(1) 26·1	13·0	
114		19·9	(1) 18·9	(3) 47·8	(3) 33·2	(1) 29·2	18·7	
115		10·1	(2) 34·9	(5) 83·7	(1) 20·9	(2) 97·7	170·9	
116		12·2	(1) 17·7	(3) 62·4	(3) 91·8	(1) 18·8	40·2	
117		23·8	(2) 26·0	(11) 110·2	(2) 83·0	(2) 20·1	31·1	
118		37·7	(1) 25·2	(1) 18·2	(2) 57·9	(3) 56·9	36·6	
119		36·1	(3) 69·3	(7) 41·5	(5) 104·7	(4) 47·6	12·3	
120		6·8	(1) 66·9	(9) 85·3	(1) 28·8	(1) 38·1	67·2	

Shelves were too high for some staff frequent use of step ladder seemed to slow the process
Label printer had to be opened up and sorted after a brick Sam. Staff made manual corrections

FIGURE B.4: Raw timing data from Pharmacy 4: Page 4



	No# items	p-Reception	Label gen	Picking meds	Apply labs	Acc chec	Hand over/store	C. D. (y/n)
121		10.3	(1) 24.1	(1) 7.4	(1) 23.0	(2) 19.6	137.9	
122		28.7	(1) 17.9	(1) 24.4	(1) 18.8	(4) 77.1	40.7	
123		18.6	(2) 45.6	(1) 8.5	(1) 11.8	(2) 8.0	18.8	
124		42.8	(3) 60.1	(3) 51.9	(1) 17.2	(1) 16.9	13.1	
125		46.1	(2) 47.3	(2) 28.6	(1) 20.1	(1) 7.2	27.3	
126		8.0	(4) 57.7	(4) 89.1	(2) 33.1	(2) 16.9	146.5	
127		12.1	(4) 52.7	(1) 7.8	(2) 41.5	(1) 5.6	77.7	
128		30.5	(1) 26.1	(1) 23.4	(1) 23.0	(3) 45.9	15.0	
129		11.1	(1) 26.5	(1) 52.5	(3) 47.4	(10) 210.3	135.9	
130		28.7	(1) 6.7	(1) 29.3	(4) 61.5	(6) 93.4	33.3	
131		40.1	(1) 18.8	(2) 40.4	(3) 64.2	(2) 45.7	48.9	
132		29.2	(1) 68.7	(1) 11.6	(1) 9.5	(1) 83.1	81.2	
133		6.5	(2) 37.3	(3) 96.9	(4) 73.4	(1) 15.6	33.4	
134		8.0	(6) 55.5	(1) 33.2	(4) 55.2	(2) 28.3	27.2	
135		18.4	(1) 17.8	(1) 140.4	(1) 65.2	(1) 19.3	31.4	
136		5.4	(2) 32.3	(1) 6.0	(1) 33.1	(3) 50.2	100.3	
137		11.3	(1) 24.3	(1) 65.8	(1) 65.0	(1) 21.1	91.1	
138		15.8	(1) 30.7	(1) 41.8	(1) 35.1	(2) 21.5	87.3	
139		20.3	(4) 36.8	(2) 63.4	(1) 30.5	(1) 9.8	50.4	
140		31.2	(2) 24.6	(5) 95.3	(1) 22.2	(1) 24.6	56.7	
141		39.4	(2) 78.6	(4) 51.9	(1) 8.6	(5) 42.6	56.2	
142		80.5	(4) 61.1	(3) 36.0	(1) 10.6	(9) 125.2	10.3	
143		7.3	(3) 87.9	(1) 8.6	(5) 142.3	(6) 99	10.0	
144		18.2	(2) 54.7	(1) 7.0	(3) 22.8	(1) 22.9	31.9	
145		48.3	(5) 64.9	(4) 28.5	(2) 79.2	(1) 19.6	28.3	
146		61.4	(1) 44.5	(1) 47.6	(3) 47.3	(1) 33.9	87.5	
147		82.5	(2) 21.7	(2) 40.6	(3) 41.9	(1) 7.1	16.4	
148		123.2	(3) 134.3	(1) 29.2	(1) 25.8	(2) 37.6	15.9	
149		47.9	(4) 40.7	(2) 21.6	(2) 47.6	(1) 29.2	17.7	
150		41.1	(2) 49.6	(3) 48.5	(1) 11.8	(6) 112.7	54.6	
151		42.8	(1) 30.3	(1) 19.3	(1) 23.7	(2) 16.9	61.7	
152		30.5	(3) 126.4	(5) 51.9	(4) 61.3	(3) 36.6	28.7	
153		8.3	(1) 27.9	(1) 19.6	(3) 61.4	(1) 26.1	24.9	
154		20.1	(2) 43.6	(3) 44.5	(7) 91.2	(1) 21.0	38.4	
155		30.5	(1) 60.6	(2) 13.8	(8) 87.3	(2) 35.0	53.3	
156		60.3	(1) 37.5	(1) 27.1	(10) 183.4	(2) 24.5	35.3	
157		105.2	(1) 17.2	(1) 7.4	(3) 60.2	(1) 21.0	77.5	
158		91.7	(1) 28.8	(2) 21.0	(1) 22.3	(4) 34.1	59.7	
159		32.3	(2) 22.7	(1) 15.2	(2) 31.7	(2) 40.4	9.5	
160		19.9	(1) 33.7	(1) 21.3	(1) 19.8	(8) 111.5	87.3	

Appendix C

Ethics Approval Letter

The following page shows a full copy of an ethical approval letter obtained for on site studies of community pharmacies from the University of Nottingham School of Medicine.

FIGURE C.1: Ethics Approval Letter from the Nottingham School of Medicine

University of Nottingham
UK | CHINA | MALAYSIA

Email: FMHS-ResearchEthics@nottingham.ac.uk

Faculty of Medicine & Health Sciences
Research Ethics Committee
c/o Faculty PVC Office
School of Medicine Education Centre
B Floor, Medical School
Queen's Medical Centre Campus
Nottingham University Hospitals
Nottingham, NG7 2UH

24th August 2017

Matthew Naybour
PhD Student
c/o Dr Rasa Remenyte-Prescott
Assistant Professor in Risk and Reliability Engineering
Room C22 NTEC
Faculty of Engineering
Nottingham
NG7 2RD

Dear Matthew

Ethics Reference No: 55-1707 – please always quote	
Study Title: Understanding the process of dispensing in community pharmacies: A mixed methods study using observational data collection and interviews.	
Short Title: Understanding the process of dispensing in community pharmacies.	
Chief Investigator/Supervisor: Matthew Boyd, Assistant Professor in Pharmacy Practice, School of Pharmacy.	
Lead Investigators/student: Matthew Naybour, PhD Student, Civil Engineering- Risk and Reliability.	
Other Key Investigators: Rasa Remenyte-Prescott, Assistant Professor in Risk and Reliability engineering, Civil Engineering.	
Type of Study: workflow, pilot study, qualitative, mixed methods PhD Student project	
Proposed Start Date: 01/09/2017	**Proposed End Date:** 01/09/2018 12mths
No of Subjects: 4-8 Community pharmacies	**Age:** 18+years
School: Pharmacy, Engineering	

Thank you for submitting the above application which has been considered by the Committee at its meeting on 18 July 2017and the following documents were received:

- FMHS REC Application form and supporting documents version 1.0: 17 May 2017.

These have been reviewed and are satisfactory and the study has been given a favourable opinion.

A favourable opinion is given on the understanding that the conditions set out below are followed:

1. You should follow the protocol agreed and inform the Committee of any changes using a notification of amendment form (please request a form).

2. You must notify the Chair of any serious or unexpected event.

3. An End of Project Progress Report is completed and returned when the study has finished (please request a form).

Yours sincerely

Professor Ravi Mahajan
Chair, Faculty of Medicine & Health Sciences Research Ethics Committee

Appendix D

Code Examples

This Appendix will highlight some examples of how some key objects and routines were coded in C++ as a part of this thesis.

D.1 Transitions

Transitions were coded using inheritance classes. The following code snippet shows the abstract base class used to represent transitions within the code. There is only 1 virtual function (defined by being set equal to 0), calcDelay. New classes which inherit the class transitions as the base class have different versions of calcDelay to match the type of distribution they represent. Two examples of these are highlighted below this code snippet.

```
class transitions
{
   static int Tcounter;

   public:

   transitions()
   {
      enabled = 0;
      number = Tcounter;
      Tcounter++;
      memory.push_back(0);
```

```
   std::cout << "\n" << "transition " << number << "constructed." <<
'\n';
};
~transitions(){};


virtual float calcDelay(int n=0, int m= 0)=0;


bool calcMemory(int n)
{
   return memory[n];
}

float checkDelay(int n)
{
   return delays[n];
}

void setMemory(bool a, int n)
{
   memory[n] = a;
}

void pop_memory(int n)
{
   memory.erase(memory.begin()+n);
}

void pop_delay(int n)
{
   delays.erase(delays.begin()+n);
}


void initialise_memory(bool x)
{
   memory.push_back(x);
}
```

```
int check_memory_size()
{
   return memory.size();
}


float min_delay()
{   //int y;
   int n = delays.size();
   float MIN = 100000;

   for(int i = 0; i < n ; i++)
   {
      if(delays[i]<MIN && delays[i]>=0)
      {
         MIN = delays[i];
      }
   }
   return MIN;
}

int min_clock()
{
   float a = 100000;
   int smallestTransition= 1;
   int MARK = 0;
   int n= delays.size();

   for (int i = 0 ; i < n; i++)
   {
      if (delays[i]<a && delays[i]>=0)
      {
         smallestTransition=MARK+1;
         a = delays[i];
      }
      MARK++;
   }
```

```
    return smallestTransition;
}

void erase_memory()
{
    int n = memory.size();

    for(int i = 0; i< n; i++)
    {
        memory.pop_back();
    }
    memory.push_back(0);
}

void empty_delays()
{
    int n = delays.size();
    for(int i=0; i<n; i++)
    {
        delays.pop_back();
    }
}

void set_delay(float r)
{
    delays.push_back(r);
}

void countdown_delay(int n, float r)
{
    delays[n] -= r;
}

int pSource()
{
    return prescription_source;
}
```

```
int wSource()
{
   return worker_source;
}

bool is_processor()
{
   if (processor== 1)
   {
      return true;
   }
   else
   {
      return false;
   }
}

int inspect_transition()
{
   int x = memory.size();
   int y = delays.size();
   std::cout<<'\n';
   std::cout << number << " has " << x << " in memory, and " << y <<
" in delays ";
   std::cout<< "Mvec=" ;
   for (int i=0; i < x ; i++)
   {
      std::cout<< memory[i] << " ";
   }
   std::cout<< "  , D vec=";
   for (int i=0; i<y; i++)
   {

      std::cout<< delays[i] << " " ;
   }
   return 0;
}

protected:
```

```
    int enabled;       // the integer signifies how many self concurrent
    clocks are enabled at a given time

    int number;         // a numeric symbol of which transition we are
    dealing with

    int prescription_source;        // The source of prescription tokens
    passing through the transition
    // negative source values are used for places which don't have a
    direct source of tokens
    // so, if a transition has prescription_source = -1, outarcs from it
    will add new tokens into the system rather than shifting existing
    tokens around
    // -1 indicates walkin generation, -2 indicates delivery generation

    int worker_source;              // This notes the source of workers
    used to process prescriptions through out the net.
    // Worker source of -1 generates pharmacists, -2 generates new
    dispensers

    // positive values of worker source and prescription source indicate
    that tokens are taken from those places

    bool processor;                 // processor transitions ping the
    distribution attached to them a number of times equal to the
    // number of items in the prescription
    std::vector<float> delays;

    std::vector<bool> memory;       // used to indicate if the
    transition has been waiting to fire, while other transition(s) have
    been firing

    private:
};
```

The following code snippet shows the inherited class used for modelling the normal

distribution. The normal distribution is initialised with 5 variables: mean, standard deviation, psource, wsource, and Boolean variable a. The integer psource related to where prescriptions tokens passing through this transition originate from, similarly, wsource indicates where worker tokens passing through this transition originate from. The Boolean value a indicates whether this is a processor transition.

```
class normal : public transitions
{
public:
normal(float m, float sd, int psource, int wsource, bool a=0)
{
mean = m;
standard_dev = sd;
prescription_source = psource;
worker_source = wsource;
processor = a;
};

~normal() {};

float calcDelay(int n = 0, int m = 0)
{   double x =0;
double y =0;
x = (rand() % 1000000 );

x/=1000000;
y = (rand() % 1000000 );

y/=1000000;
if (x==0)
{
return mean;
}
x = sqrt(-2*log(x))*cos(2*180*y);
x = x*standard_dev;
x = x+mean;

if(x<0)
```

```
{
x = -x;
}
return x;
}

protected:
float mean;
float standard_dev;

private:

};
```

The following code snippet indicates how real data was incorporated into the CPN. The float values from p1-p12 are parameters defined from the distribution fitting analysis of in-field data.

```
class accuracy : public transitions
{
public:
accuracy(float p1, float p2, float p3, float p4,
float p5, float p6, float p7, float p8,
float p9, float p10, float p11, float p12,
int psource, int wsource, int n, bool a=0): rand_seed(n), engine(n)
{
P1_1 = p1;
P1_2 = p2;
P2_1 = p3;
P2_2 = p4;
P3_1 = p5;
P3_2 = p6;
P4_1 = p7;
P4_2 = p8;
P5_1 = p9;
P5_2 = p10;
P6_1 = p11;
P6_2 = p12;
prescription_source = psource;
```

```
worker_source = wsource;
processor = a;
};

~accuracy() {};

float calc_lognormal(float mean, float standard_dev)
{
std::lognormal_distribution<double> distribution(mean, standard_dev);
return distribution(this->engine);

}

float calc_gamma(float alpha, float beta)
{
std::gamma_distribution<double> distribution(alpha, beta);
return distribution(this->engine);
}

float calcDelay(int n = 0, int m = 0)
{   float number = 0;
if(m==1)
{
number = calc_lognormal(P1_1, P1_2);

}

if(m==2)
{
number = calc_lognormal(P2_1, P2_2);

}

if(m==3)
{
number = calc_lognormal(P3_1, P3_2);

}
```

```
if(m==4)
{
number = calc_gamma(P4_1, P4_2);
}

if(4<m && m <=8)
{
number = calc_lognormal(P5_1, P5_2);
}

if(8<m)
{
number = calc_lognormal(P6_1, P6_2);
}

if(number<0)
{
number = -number;
}
return number;
}


protected:
float P1_1;
float P1_2;
float P2_1;
float P2_2;
float P3_1;
float P3_2;
float P4_1;
float P4_2;
float P5_1;
float P5_2;
float P6_1;
float P6_2;

unsigned long rand_seed;
std::default_random_engine engine;
```

```
private:

};
```

D.2 Prescriptions

The following code snippet contains the class used to represent prescription tokens in the net. The value of b is generated using a random variable, and the value of a is determined by where the prescription enters the net from.

```
class tokens
{
public:

static int Tcounter;

tokens(int b=0, bool a=0)
{
time_to_dispense = 0;
itterations = 0;
outcome = 0;
labels = 0;
contents = 0;
application=0;
walk_in = a;
items = b;
staff = -1;
number = Tcounter;
Tcounter++;

};


~tokens(){ };

void outData()
```

```
{
std::cout<< time_to_dispense << ',' << walk_in << ','  << itterations
    << ',' << outcome << ',' << labels << ',' << contents <<','<<
    application <<',' << items << ',' << number ;
}


void addTime(float x)
{
time_to_dispense += x;
}

void setLabels(bool a)
{
labels=a;
}

void setContents(bool a)
{
contents=a;
}

void setApplication(bool a)
{
application = a;
}

void setItterations(bool a)
{
itterations +=a;
}

void setItems(int a)
{
items = a;
}

int check_items()
{
```

```
return items;
}

float Check_timeToD()
{
return time_to_dispense;
}

bool check_type()
{
return walk_in;
}

int check_itterations()
{
return itterations;
}

int check_outcome()
{
return outcome;
}

bool check_labels()
{
return labels;
}

bool check_contents()
{
return contents;
}

bool check_application()
{
return application;
}

void setOutcome(int n)
```

```
{
outcome = n;
}


protected:

float time_to_dispense;     // self explanatory

bool walk_in;               // Type of prescription; 1 for walk-ins, 0
    for deliveries

int itterations;            // Number of times a prescription had to go
    through production until success

int outcome;                // Overall outcome of the prescription:
    Near Miss, Dispensing Error or successful procedure

bool labels;                // Current state of the prescriptions
    labels, 1 is correct, 0 is error

bool contents;              // Current state of the prescriptions
    contents, 1 correct, 0

bool application;           // were labels applied correctly?

int number;                 // number prescription

int items;                  // indicator of how many items are in a
    prescription

int staff;                  // Indicator of staff type dealing with the
    prescription


private:
};
```

D.3 Staff

The next code snippet contains the class used to represent staff tokens in the net. The type of staff is indicated by the value of the variable worker_type. The remaining variables stored in the class correspond to how long the member of staff has spent completing each type of task.

```
class workers
{
public:
workers(int a)
{
worker_type = a;
time_advanced = 0;
recieving = 0;
time_checking =0;
time_lunch = 0;
time_primary = 0;
time_idle = 0;
time_counselling = 0;
time_storing = 0;
time_stocking = 0;
};


int check_worker_type()
{
return worker_type;
}
void outData()
{
if (worker_type ==1)
{
std::cout<< "pharmacist - "  << '\n'
<< "R-Time = " << recieving
<< ", Prim-time = " << time_primary
<< ", Check-time = " << time_checking
<< ", Adv-time = " <<  time_advanced
<< ", id-time = " << time_idle
```

```
<< ", Counsel-time = " <<  time_counselling
<< ", Storing-time = " << time_storing
<< ", Stocking-time = " << time_stocking;
}

else if (worker_type ==0)
{
std::cout << "Dispenser - " << '\n'
<< "R-Time = " << recieving
<< ", Prim-time = " << time_primary
<< ", Check-time = " << time_checking
<< ", Adv-time = " <<  time_advanced
<< ", id-time = " << time_idle
<< ", Counsel-time = " <<  time_counselling
<< ", Lunch-time = " << time_lunch
<< ", Stock-time = " << time_stocking;
}

}


void addReceive(float t)
{
recieving+= t;
}

void addPrimary(float t)
{
time_primary+= t;
}

void addChecking(float t)
{
time_checking+= t;
}

void addAdvanced(float t)
{
time_advanced +=t;
```

```
}

void addLunch(float t)
{
time_lunch+= t;
}

void addIdle(float t)
{
time_idle+= t;
}

void addCounselling(float t)
{
time_counselling+=t;
}

void addStoring(float t)
{
time_storing += t;
}

void addStocking(float t)
{
time_stocking += t;
}

float t_lunch()
{
return time_lunch;
}


float t_idle()
{
return time_idle;
}

float t_primary()
```

```
{
return time_primary;
}

float t_checking()
{
return time_checking;
}

float t_advanced()
{
return time_advanced;
}

float t_recieving()
{
return recieving;
}

float t_counsel()
{
return time_counselling;
}


float t_store()
{
return time_storing;
}

float t_stock()
{
return time_stocking;
}
protected:

int worker_type;        // 0 for dispensers, 1 for pharmacists

float recieving;
```

```
float time_primary;

float time_lunch;

float time_idle;

float time_checking;

float time_advanced;

float time_counselling;

float time_storing;

float time_stocking;

private:
};
```

D.4 Places

The next code snippet contains the class used to represent places in the net. Each place is given an initial number of tokens for each token type which is used when resetting the Petri net back to its initial conditions. There are a set of functions controlling the increments of time spent completing each type of task for staff tokens, and incrementing the time taken to dispense for prescription tokens. Other functions are used to add errors to prescription tokens, moving tokens around, resetting the number of tokens on each place, or reading information off of tokens.

```
class places
{


public:

static int Pcounter;
```

```
places(int n=0, int p =0, int ph=0, int d= 0)
{
number = Pcounter;
Pcounter++;
basic_tokens = n;
initial_basics = n;
initial_prescriptions = p;
initial_pharmacists = ph;
initial_dispensers = d;

for (int i= 0; i< initial_pharmacists; i++)
{
generate_pharmacist();
}

for (int i=0; i<initial_dispensers; i++)
{
generate_dispenser();
}

for (int i= 0; i<initial_prescriptions; i++)
{
generate_walkin(1);
}

// number of basic tokens on the place
std::cout << "\n" << "place " << number << "constructed." << '\n';

};

~places()
{
std::cout << '\n' << "place " << Pcounter << " destructed " << '\n';



};
```

```
void erase_nth(int n)
{
contents.erase(contents.begin()+n);

}

void pushBack(tokens x)
{
contents.push_back(x);
}

tokens access_nth(int n)
{
return contents[n];
}

void erase_nth_worker(int n)
{
staff.erase(staff.begin()+n);
}

void pushBack_worker(workers x)
{
staff.push_back(x);
}


workers access_nth_worker(int n)
{
return staff[n];
}


void set_stocking_time(int n, float a)
{
staff[n].addStocking(a);
}
```

```
void set_receive_time(int n, float a)
{
staff[n].addReceive(a);
}

void set_primary_time(int n, float a)
{
staff[n].addPrimary(a);
}

void set_checking_time(int n, float a)
{
staff[n].addChecking(a);
}

void set_idle_time(int n, float a)
{
staff[n].addIdle(a);
}

void set_counselling_time(int n, float a)
{
staff[n].addCounselling(a);
}


void set_storing_time(int n, float a)
{
staff[n].addStoring(a);
}

void set_advanced_time(int n, float a)
{
staff[n].addAdvanced(a);
}

void set_lunch_time(int n, float a)
{
staff[n].addLunch(a);
```

```
}

void set_time(int n, float a)
{
contents[n].addTime(a);
}

void set_labels(int n, bool a)
{
contents[n].setLabels(a);
}



void set_contents(int n, bool a)
{
contents[n].setContents(a);
}

void set_application(int n, bool a)
{
contents[n].setApplication(a);
}

void increase_itterations(int n, bool a)
{
contents[n].setItterations(a);
}

void calc_outcome(int i)
{
if (contents[i].check_contents() == 0 && contents[i].check_labels() ==0
    && contents[i].check_application() == 0 &&
    contents[i].check_itterations()==0)
{
contents[i].setOutcome(1);  // ALL CORRECT
}
if (contents[i].check_itterations()>0)
{
contents[i].setOutcome(2);  // NEAR MISS
```

```
}

if (contents[i].check_contents() == 1 || contents[i].check_labels() ==1
    || contents[i].check_application() == 1 )
{
contents[i].setOutcome(3);  // DISPENSING ERROR
}
}


void generate_pharmacist()
{
int a = 1;
staff.emplace_back(a);
}

void generate_dispenser()
{
int a = 0;
staff.emplace_back(a);
}

void generate_walkin(int b)
{


int a = 1;
contents.emplace_back(b, a);


// emplace_back takes a bunch of parameters and calls the constructor
    that best matches those parameters (or fails if no such constructor
    exist),
// using said constructor to create the object, without any copy, at
    the end of the container
};

void generate_delivery(int b)
{
```

```
int a = 0;
contents.emplace_back(b, a);
}

/* void delete_entry(int x)  // delete an entry from the vector
    contents, in place x
{
tokens *point = contents[x];

delete *point;
}
*/

int inspect_place()
{
int x = contents.size();
int y = staff.size();

std::cout <<  number << " contains " << basic_tokens << " b-tokens, "
    << x << " p-tokens, " << y << " w-tokens."  << '\n';

for (int i=0; i < x ; i++)
{
contents[i].outData();
std::cout<<'\n';
}

for (int i=0; i<y; i++)
{
staff[i].outData();
std::cout<<'\n';
}
return 0;
}

int Number_of_basics()
{
return basic_tokens;
}
```

```
int Number_of_prescriptions()
{
return contents.size();
}

int Number_of_workers()
{
return staff.size();
}

void add_basics(int n)
{
basic_tokens += n;
}

void remove_basics(int n)
{
basic_tokens -= n;
}

void reset_basics()
{
basic_tokens = initial_basics;
}

void reset_prescriptions(int a)
{
int x = contents.size();

for (int i = 0; i<x; i++)
{
erase_nth(0);
}

for (int i =0; i< initial_prescriptions; i++)
{
generate_delivery(a);
}
```

```
}

void reset_workers()
{
int x = staff.size();

for (int i = 0; i<x; i++)
{
erase_nth_worker(0);
}

for (int i = 0; i < initial_pharmacists; i++)
{
generate_pharmacist();
}

for (int i = 0; i < initial_dispensers; i++)
{
generate_dispenser();
}
}
void clear_basics()
{
basic_tokens = 0;
}
void clear_workers()
{
int x = staff.size();
for (int i = 0; i<x; i++)
{
erase_nth_worker(0);
}
}
void set_initial_pharmacists(int a)
{
initial_pharmacists = a;
}
```

```
void set_initial_basics(int a)
{
initial_basics = a;
}


void set_initial_dispensers(int a)
{
initial_dispensers = a;
}
protected:

std::vector<tokens> contents;

std::vector<workers> staff;

int initial_prescriptions;

int initial_pharmacists;

int initial_dispensers;

int initial_basics;

int basic_tokens;

int number;


private:
};
```

D.5 Main

The following code snippet outlines the main function used to simulate a CPN. The function begins by setting out all the places and transitions within two vectors before asking the user to input a checking strategy and a number of simulations. Two result files are opened to store the results of simulations. Then two 'for' loops are opened, one for the number of simulations, and the other for the maximum number of transition firings. In reality this limit is never reached, and the condition of $simTime > 28800$ (the number of seconds in the day being exceeded), ends iteration. The for loop of using the variable M runs once each time a transition fires. The functions called execute a CPN firing rule with some additional functions for adding time to tokens in the net, and applying some balancing operations when transitions with multiple outcomes take place.

```
int main()
{   srand(time(NULL));
places* walkin_generation = new places(1);
places* walkin_arrived = new places();
places* delivery_generation = new places(1);
places* dispenser_on_counter = new places();
places* mixed_prescriptions = new places();
places* available_dispensers = new places(0, 0, 0, 2);
places* labelling_station = new places(2, 0);
places* allocate_dispense = new places();
places* labelling = new places();
places* filling_apply = new places();
places* waiting_for_final_check = new places();
places* pharmacist_ready_to_check = new places();
places* acc_checking = new places(0, 0, 0, 0);
places* handing_over_to_patient = new places();
places* finished_prescriptions = new places();
places* pharmacist_available = new places(0, 0, 1);
places* pharmacist_busy = new places();
places* NMS_waiting = new places();
places* NMS_generation = new places(1);  // add 1 to implement NMS
places* stocking_generator = new places(1); // add 1 to implement
    stocking generation
places* stocking_task_ready = new places();
```

```
places* worker_stocking = new places();
places* lunch_generator = new places(1); // add 1 to implement lunch
    hours
places* lunch_ready = new places();
places* lunch_eating = new places();
places* count_arrived = new places();
places* adv_arrived = new places();

pVEC[0] = walkin_generation;
pVEC[1] = walkin_arrived;
pVEC[2] = delivery_generation;
pVEC[3] = dispenser_on_counter;
pVEC[4] = mixed_prescriptions;
pVEC[5] = available_dispensers;
pVEC[6] = labelling_station;
pVEC[7] = allocate_dispense;
pVEC[8] = labelling;
pVEC[9] = filling_apply;
pVEC[10] = waiting_for_final_check;
pVEC[11] = pharmacist_ready_to_check;
pVEC[12] = acc_checking;
pVEC[13] = handing_over_to_patient;
pVEC[14] = finished_prescriptions;
pVEC[15] = pharmacist_available;
pVEC[16] = pharmacist_busy;
pVEC[17] = NMS_waiting;
pVEC[18] = NMS_generation;
pVEC[19] = stocking_generator;
pVEC[20] = stocking_task_ready;
pVEC[21] = worker_stocking;
pVEC[22] = lunch_generator;
pVEC[23] = lunch_ready;
pVEC[24] = lunch_eating;
pVEC[25] = count_arrived;
pVEC[26] = adv_arrived;

// For P source:
```

```
// Source of -1 generates walk-ins, source of -2 generates deliveries,
    source of n >= 0 means the transitions target takes tokens from
    place n
// (Source is the last integer in the constructors)
// Source of -3 is used as an exclusionary source, where no p tokens
    will be generated or moved by a transition firing

// For W source
// Similar to above, however in this case -1 generates a pharmacist, -2
    generates a dispenser
// again a source of -3 indicates that no worker tokens will be
    generated by transitions firing

// Transitions are defined (random variables, p-source, w-source,
    processor (yes/no))

exponential* walkin_generate = new exponential(0.0033, -1, -3);
uniform* receive_walkin = new uniform(30, 60, 2, 4);
deterministic* move_to_counter = new deterministic(0, -3, 6);     //
    0.00011
deterministic* delivery_generate = new deterministic(6000, -2, -3);
deterministic* allocate_dispenser = new deterministic(0, -3, 6); //
    0.0011
uniform* choose_prescription_to_label = new uniform(5, 10, 5, 8);
deterministic* print_labels = new deterministic(15, 9, 9, 1);
normal* fill_and_apply_labels = new normal(50, 10, 10, 10, 1);
deterministic* take_pharmacist = new deterministic(0, -3, 16);
uniform* choose_prescription_to_check = new uniform(10, 15, 11, 12);
uniform* final_accuracy_check = new uniform(5, 10, 13, 13, 1);
store_or_hand_out* hand_over_to_patient_or_store_1 = new
    store_or_hand_out(0.025, 0.05, 14, 14);
deterministic* distract_pharmacist = new deterministic(0, -3, 16);
uniform* carry_out_NMS = new uniform(300, 600, -3, 17);
exponential* NMS_generator = new exponential(0.00006, -3, -3);
deterministic* stocking_arrival = new deterministic(6600, -3, -3);
deterministic* allocate_stocking = new deterministic(0, -3, 6);  //
    0.0011
uniform* finish_stocking = new uniform(300, 900, -3, 22);
deterministic* disp_lunch_1 = new deterministic(7200, -3, -3);
```

```
deterministic* disp_lunch_2 = new deterministic(0, -3, 6);    // 0.0001
deterministic* disp_finish_eating = new deterministic(3600, -3, 25);
deterministic* transfer_pharmacist = new deterministic(10, -3, 16);


tVEC[0] = walkin_generate;
tVEC[1] = receive_walkin;
tVEC[2] = move_to_counter;
tVEC[3] = delivery_generate;
tVEC[4] = allocate_dispenser;
tVEC[5] = choose_prescription_to_label;
tVEC[6] = print_labels;
tVEC[7] = fill_and_apply_labels;
tVEC[8] = take_pharmacist;
tVEC[9] = choose_prescription_to_check;
tVEC[10] = final_accuracy_check;
tVEC[11] = hand_over_to_patient_or_store_1;
tVEC[12] = distract_pharmacist;
tVEC[13] = carry_out_NMS;
tVEC[14] = NMS_generator;
tVEC[15] = stocking_arrival;
tVEC[16] = allocate_stocking;
tVEC[17] = finish_stocking;
tVEC[18] = disp_lunch_1;
tVEC[19] = disp_lunch_2;
tVEC[20] = disp_finish_eating;
tVEC[21] = transfer_pharmacist;

int checkStrategy;
int numberOFsimulations;
std::cout << "Enter the checking strategy, 1-3, you would like to use."
    << "\n";
std::cin >> checkStrategy;

std::cout << "Enter the number of simulations you would like to run."
    << "\n";
std::cin >> numberOFsimulations;

RESULTS.open ("PN sim 1 - general.csv");
```

```
RESULTS << "Simulation no#" << ','
<< "Total completed" << ','
<< "Number of deliveries" << ','
<< "Mean time to dispense (D's)" << ','
<< "Standard Deviation (D's)" << ','
<< "Number of walk-ins" << ','
<< "Mean time to dispense (WI's)" << ','
<< "Standard Deviation (WI's)" << ','
<< "No# All Correct" << ','
<< "No# Near Misses" << ','
<< "No# dispensing Errors" << ','
<< "WIs delayed" << ','
<< "WIs Near misses" << ','
<< "Average items WI delays" << ','
<< "D-lunch" << ','
<< "D-idle" << ','
<< "D-prim" << ','
<< "D-rec" << ','
<< "D-stock" << ','
<< "P-idle" << ','
<< "P-prim" << ','
<< "P-rec" << ','
<< "P-adv" << ','
<< "P-check" << ','
<< "P-counsel" << ','
<< "P-Store" << ','
<< "P-stocking" << ','
<< "Adv-comp" << ','
<< "Arrivals" << ','
<< "Adv-arrived" << ','
<< '\n';
RESULTS_2.open ("PN sim 1 - dispense times.csv");
RESULTS_2 << "Presciption no#" << ','
<< "Time to dispense" << ','
<< "Number of items" <<  ','
<< "Itterations" << ','
<< '\n';

for (int i=0; i< numberOFsimulations; i++)
```

```
{
float simTime = 0.0;
int tFiring = 0;    // tFiring is used to hold the number of the
    transition to fire next
int cFiring = 0;

for (int M= 0; M <100000 ; M++)  // M is the max number of transitions
    which can fire in a simulation
{

if(simTime>28800)
{
break;
}

prioritise_walkins(5);  // TOKEN SHUFFLE on place x, bring walk-ins to
    the top of the vector, maintains first come first served
prioritise_walkins(11);

generate_enabledVEC();

sort_memory();   // used to provide new memory values of 0 to memoryVEC
    when new clocks are added to a transition

generate_delays();   // find the minimum of delayVEC to determine which
    transition fires first

set_memory();           // Determines if a transition has remained
    enabled since the last transition firing, and needs to have its new
    delay equal to the difference between the two times.

tFiring = minTransition();      // find the transition to fire

delay = tVEC[tFiring-1]->min_delay();
cFiring = tVEC[tFiring-1]->min_clock();
WT = determine_worker_type(tFiring, cFiring);

simTime += delay;
```

```
add_dispense_time(delay);
add_accuracy_checking_time(delay);
add_advaced_services_time(delay);
add_stock_time(delay);
add_idle_time(delay);
add_labelling_and_assembling_time(delay);
add_receiving_time(delay);
add_counselling_or_storing_time(delay);

fireTransition(tFiring, cFiring);

// apply post firing conditions for only these transitions

if(tFiring == 2  || tFiring == 7 || tFiring == 8 ||
tFiring == 11 )
{
int a = rand()% 1000;
int o = rand()% 1000;
int p = rand()% 1000;
int q = generate_q(a, tFiring);

apply_post_firing_conditions(tFiring, o, p, q, WT);
}

if(tFiring == 8 && checkStrategy == 1)
{
// THIS IS NO INTERMEDIATE CHECK
{
int n = pVEC[4]->Number_of_prescriptions();
pVEC[4]->erase_nth(n-1);
pVEC[4]->remove_basics(1);
}
}

if(tFiring ==8 && checkStrategy ==2)
{
int a = rand()% 1000;
int Q = gen_Q(a);
intermediate_check(Q);    // THIS IS INTERMEDIATE CHECK A
```

```
}

if(tFiring == 8 && checkStrategy == 3)
{
int a = rand()% 1000;
int Q = gen_Q(a);
{
int r = rand()% 1000;
int o = rand()% 1000;
int p = rand()% 1000;
alt_intermediate_check(o, p, r, Q);  // THIS IS INTERMEDIATE CHECK B
}
}

}

std::cout << "The " << i+1 << "th" << " simulation lasted a total of "
    << simTime << " seconds \n";

Write_place_a_to_RESULTS(15, i+1, simTime);                    //
    Both types of info now written simultaneously
write_dispense_times_to_results();


for (int i = 0; i< NOPLACES ; i++)
{
pVEC[i]->reset_basics();
pVEC[i]->reset_prescriptions(generate_items());
pVEC[i]->reset_workers();
}
}

RESULTS.close();
RESULTS_2.close();
return 0;
}
```

References

Abecassis, Z. A., McElroy, L. M., Patel, R. M., Khorzad, R., Carroll IV, C., and Mehrotra, S. (2015). Applying fault tree analysis to the prevention of wrong-site surgery. *Journal of Surgical Research*, 193(1):88–94.

Afolabi, M. O. and Erhun, W. O. (2003). Patients' response to waiting time in an out-patient pharmacy in Nigeria. *Tropical Journal of Pharmaceutical Research*, 2(2):207–214.

Akaike, H. (2011). *International Encyclopedia of Statistical Science*, chapter Akaike's Information Criterion, page 25. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ali, M. M., Khompatraporn, C., and Zabinsky, Z. B. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, 31(4):635–672.

Anderson, P. M., Chintaluri, G. M., Magbuhat, S. M., and Ghajar, R. F. (1997). An improved reliability model for redundant protective systems-markov models. *IEEE Transactions on Power Systems*, 12(2):573–578.

Andrews, J. (2009). System reliability modelling: the current capability and potential future developments. *Journal of Mechanical Engineering Science*, 223(12):2281–2897.

Andrews, J. D. and Moss., T. R. (2002). *Reliability and risk assessment*. London : Professional Engineering Publishing, London, 2nd ed. edition.

Ashcroft, D., Quinlan, P., and Blenkinsopp, A. (2005). Prospective study of the incidence, nature and causes of dispensing errors in community pharmacies. *Pharmacoepidemiology and drug safety*, Volume 5(5):327–332.

Augusto, V. and Xie, X. (2014). A modeling and simulation framework for health care systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(1):30–46.

Aytug, H., Khouja, M., and Vergara, F. (2003). Use of genetic algorithms to solve production and operations management problems: a review. *International Journal of Production Research*, 41(17):3955–4009.

Baesler, F., Gatica, J., and Correa, R. (2015). Simulation optimisation for operating room scheduling. *International Journal of Simulation Modelling*, 4(2):215–26.

Bahadori, M., Mohammadnejhad, S. M., Ravangard, R., and Teymourzadeh, E. (2014). Using queuing theory and simulation model to optimize hospital pharamcy performance. *Iran Red Crescent Medical Journal*, 16(3).

Bardini, R., Politano, G., Benso, A., and Carlo, S. D. (2017). Using multi-level petri nets models to simulate microbiota resistance to antibiotics. In *IEEE International conference on Bioinformatics and Biomedicine*, Kansas City, MO, USA, 13-16 November.

Barker, K. N., Flynn, E. A., and Pepper, G. A. (2002). Observation method of detecting medication errors. *Am J Helath-Syst Pharm*, 59(23):2314–2316.

Bause, F. and Kritzinger, P. (1996). Stochastic petri nets. *Verlag Vieweg, Wiesbaden*, 26.

BBC (2018). Boots: Pharmacists under pressure? Channel 1.

Beck, R. J. and Parker, S. G. (1983). The Markov Process in Medical Prognosis. *Medical Decision Making*, 3(4):419–458.

Bell, H. M., McElnay, J. C., and Hughes, C. M. (1999). A self reported work sampling study in community pharmacy practice. *Pharm World Sci*, 21(5):210–216.

Bell Telephone Laborotories (1961). Launch control safety study. Section VII, Murrary Hill, NJ USA. Vol 1.

Blum, C., Roli, A., and Dorigo, M. (2001). HC–ACO: The hyper-cube framework for Ant Colony Optimization. In *Proceedings of 4th Metaheuristics International Conference*, volume 2, pages 399–403, Porto, Portugal, July 16-20.

Boyd, M. A. (1998). An introduction to Markov modelling: Concepts and uses. In *Reliability and Maintainability Symposium*, Anaheim, CA, USA, January 19-22.

Braga, J. A. and Andrade, A. R. (2019). Optimizing maintenance decisions in railway wheelsets: A markov decision process approach. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 233(2):285–300.

Brailsford, S. C., Gutjahr, W. J., Rauner, M. S., and Zeppelzauer, W. (2007). Combined discrete-event simulation and ant colony optimisation approach for selecting optimal screening policies for diabetic retinopathy. *Computational Management Science*, 4:59–83.

Braund, R., Chesney, K. M., Keast, E. P., Ng, L. J., Qi, S., Samaranayaka, S., and Wang, E. (2012). Are all pharmacy staff interested in potential future roles? *International Journal of Pharmacy Practice*, 20(6):417–421.

Bryman, A. (2012). *Social Research Methods: 4th edition*. Oxford University Press.

Bullnheimer, B., Hartl, R. F., and Strauss, C. (1997). A new rank based version of the ant system. a computational study. In *SFB Adaptive Information Systems and Modelling in Economics and Management Science*, Vienna University of Economics and Business Administration, Wien, Austria, 2-6 August.

Burnham, K. P. and Anderson, D. R. (1998). *Model Selection and Multimodel Inference A Practical Information - Theoretic Approach*. Springer.

Chan, C.-Y. (2017). Advancements, prospects, and impacts of automated driving systems. *International Journal of Transportation Science and Technology*, 6(3):208–216.

Chang, J.-Y., Wang, C.-C., Kang, H.-C., Shen, L.-J., and Huang, C.-F. (2017). Cost-effectiveness of the pharmacist-assisted warfarin monitoring program at a medical center in Taiwan. *International Journal for Quality in Health Care*, 29(6):817–825.

Chouaid, C., Molinier, L., Combescure, C., Daures, J., Housset, B., and Vergnenegre, A. (2004). Economics of the clinical management of lung cancer in france: an analysis using a markov model. *British journal of cancer*, 90(2):397.

Chua, S., Wong, I. C., Edmondson, H., Allen, C., Chow, J., Peacham, J., Hill, G., and Grantham, J. (2003). A Feasibility Study for Recording of Dispensing Errors and 'Near Misses' in Four UK Primary Care Pharmacies. *Drug Safety*, 26(11):808–813.

Cina, J. L., Gandhi, T. K., Churchill, W., Fanikos, J., McCrea, M., Mitton, P., Rothschild, J. M., Featherstone, E., Keohane, C., Bates, D. W., and Poon, E. G. (2006). How many hospital medication dispensing errors go undetected? *Journal on Quality and Patient Safety*, 32(2):73–80.

Cochran, G. L., Klepser, D. G., Morien, M., Lomelin, D., Schainost, R., and Lander, L. (2013). From physician intent to the pharmacy label: prevalence and description of discrepancies from a cross sectional evaluation of electronic prescriptions. *British Medical Journal*, 23(3):223–230.

Cohen, M. R., Smetzer, J. L., Westphal, J. E., Comden, S. C., and Horn, D. M. (2012). Risk models to improve safety of dispensing high-alert medications in community pharmacies. *Journal of the American Pharmacists Association*, 52(5):584.

Cornford, T., Hibberd, R., and Barber, N. (2014). Evaluation of the electronic prescription service in primary care. Technical report, University of London.

Cousineau, D., Brown, S., and Heathcote, A. (2004). Fitting distributions using maximum likelihood: methods and packages. *Behaviour Research Methods, Instruments, & Computers*, 36(4):742–756.

Crosetti, P. A. and Bruce, R. A. (1970). Commercial application of fault tree analysis. In *9th Reliability and Maintainability Conference*, Detroit, Michigan, USA, July 20.

Cullen, D. J., Bates, D. W., Small, S. D., Cooper, J. B., Nemeskal, A. R., and Leape, L. L. (1995). The incident reporting system does not detect adverse drug events: A problem for quality improvement. *Journal on Quality and Patient Safety*, 21(10):541–548.

Cummings, G. E. (1975). Application of the fault tree technique to a nuclear reactor containment system. In editors, S., editor, *Conference on Reliability and Fault Tree analysis*, Berkeley, California, USA, September 3. California University.

Dammasch, K. and Horton, G. (2007). Active tokens for modelling mental health care with coloured stochastic petri nets. In *2007 Innovations in Information Technologies (IIT)*, pages 541–545.

Dansky, K. H. and Miles, J. (1997). Patient satisfaction with ambulatory healthcare services: Waiting and filling time. *Hospital and Health Services Administration*, 42(2):165–177.

Darabi, H., Galanter, W. L., Lin, J. Y. Y., Buy, U., and Sampath, R. (2009). Modeling and integration of hospital information systems with petri nets. In *2009 IEEE/INFORMS International Conference on Service Operations, Logistics and Informatics*, pages 190–195.

Davies, J. E., Barber, N., and Taylor, D. (2014). What do community pharmacists do?: results from a work sampling study in london. *International Journal of Pharmacy Practice*, 22(5):309–318.

Dean, B. and Barber, N. (2001). Validity and reliability of observational methods for studying medication administration errors. *American Journal of Health-System Pharmacy*, 58(1):54–59.

Delignette-Muller, M. L. and Dutang, C. (2015). fitdistrplus: An R package for fitting distributions. *Journal of Statistical Software*, 64(4):1–34.

Deneubourg, J. L., Aron, S., Gross, S., and Pasteels, J. M. (1990). The self-organising exploratory pattern of the Argentine Ant. *Journal of Insect Behaviour*, 3(2):159–168.

Department of Health Chief Medical Officer (2000). *An Organisation with a Memory: Report of an Expert Group on Learning from Adverse Events in the NHS*. Stationary Office.

DOH (2016). Community pharmacy in 2016/17 and beyond: Stakeholder briefing sessions. Department of Health.

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano.

Dorigo, M. and Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, 1(1):53–66.

Dorigo, M. and Stutzle, T. (2004). *Ant Colony Optimisation*. MIT Press.

Dotoli, M., Fanti, M. P., Iacobellis, G., and Ukovich, W. (2010). Modelling and management of a hospital department via petri nets. In *Health Care Management Conference*, Venice, Italy, February 18-20.

Dotoli, M., Fanti, M. P., Mangini, A. M., and Ukovich, W. (2009). A continuous Petri net model for the management and design of emergency cardiology departments. *IFAC Proceedings Volumes*, 42(17):50 – 55. 3rd IFAC Conference on Analysis and Design of Hybrid Systems.

Dowsland, K. A. and Thompson, J. M. (2005). Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society*, 56(4):426–438.

Elliott, R. A., Camacho, E., Campbell, F., Jankovic, D., James, M. M. S., Kaltenthaler, E., Wong, R., Sculpher, M. J., and Faria, R. (2018). Prevalence and economic burden of medication errors in the NHS in England. Technical report, University of Manchester and University of Sheffield and University of York.

Emmerton, L. and Jefferson, K. (1996). Work sampling observations of community pharmacists: a review. *International Journal of Pharmacy Practice*, 4(2):75–78.

Engin, O., Figlali, A., Figlali, N., and Özcale, C. (2007). Investigation of ant system parameter interactions by using design of experiments for job shop scheduling problems; computers & industrial engineering. *Computers and Industrial Engineering*, 56(2):538–559.

Epstein, B. and Weissman, I. (2008). *Mathematical Models for Reliability*. Chapman and Hall/CRC.

Ezugwu, A. E., Adeleke, O. J., Akinyelu, A. A., and Viriri, S. (2019). A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. *Neural Computing and Applications*.

Flynn, E., Barker, K., Pepper, G., Bates, D., and Mikeal, R. (2002a). Comparison of methods for detecting medication errors in 36 hospitals and skilled-nursing facilities. *American Journal of Health-System Pharmacy*, 59(5):436–446.

Flynn, E. A., Barker, K. N., and Carnahan, B. J. (2003). National observational study of prescription dispensing accuracy and safety in 50 pharmacies. *Journal of American Pharmacists Association*, 43(2):191–200.

Flynn, E. A., Dorris, N. T., Holman, G. T., Carnahan, B. J., and Barker, K. N. (2002b). Medication dispensing errors in community pharmacies: A nationwide study. In *Proceedings of the human factors and ergonomics society*, volume 46, pages 1448–1451.

Franklin, B. D. and O'Grady, K. (2007). Dispensing errors in community pharmacy: frequency, clinical significance and potential impact of authentication at the point of dispensing. *International Journal of Pharmacy Practice*, 15(4):273–281.

Franklin, B. D., O'Grady, K., Voncina, L., Popoola, J., and Jacklin, A. (2008). An evaluation of two automated dispensing machines in uk hospital pharmacy. *International Journal of Pharmacy Practice*, 16(1):47–53.

Fruggiero, F., Lambiase, A., and Fallon, D. (2008). Computer simulation and swarm intelligence organisation into an emergency department: a balancing approach across ant colony optimisation. *International Journal of Services Operations and Informatics*, 3(3):142–161.

Fusco, J. A., Paulus, E. J., Shubat, A. R., and Miah, S. (2015). Warfarin and rivaroxaban duplication: A case report and medication error analysis. *Drug Safety*, 2(1).

Gabriel, K. R. and Neumann, J. (1962). A markov chain model for daily rainfall occurrence at tel aviv. *Quarterly Journal of the Royal Meteorological Society*, 88(375):90–95.

Gambardella, L. M. and Dorigo, M. (1995). Ant-q: A reinforcement learning approach to the traveling salesman problem. In *Machine Learning Proceedings 1995*, pages 252–260. Elsevier.

Garrick, B. (1988). The approach to risk analysis in three industries: nuclear power, space systems, and chemical process. *Reliability Engineering & System Safety*, 23(3):195 – 205.

Gertman, D. I., Blackman, H. S., Marble, J. L., Smith, C., and Boring, R. L. (2005). The spar-h human reliability analysis method. *Office of Nuclear Regulatory Research*.

Gibson, W., Hicking, B., and Kirwan, B. (2006). Feasibility study into the collection of human error probability data. Technical report, Eurocontrol experimental centre.

Goss, S., Aron, A., Deneubourg, J. L., and Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76:579–581.

Government, U. (1968). Medicines act 1968, section 85.

Gregório, J., Cavaco, A. M., and Lapão, L. V. (2017). How to best manage time interaction with patients? community pharmacist workload and service provision analysis. *Social and Administrative Pharmacy*, 13(1):133–147.

Hollnagel, E., Woods, D. D., and Leveson, N. (2006). *Resilience Engineering*. Ashgate Publishing Limited.

Hsueh, K.-S. and Mosleh, A. (1996). The development and application of the accident dynamic simulator for dynamic probabilistic risk assessment of nuclear power plants. *Reliability Engineering & System Safety*, 52(3):297 – 314. Reliability and safety analysis of dynamic process systems.

Hughes, M., Carson, E. R., Makhlouf, M., Morgan, C. J., and Summers, R. (2000). Modelling a progressive care system using a coloured-timed Petri net. *Transactions of the Institute of Measurement and Control*, 22(3):271–283.

Hyman, W. A. and Johnson, E. (2008). Fault tree analysis of clinical alarms. *Journal of Clinical Engineering*, 33(2):85–94.

Jensen, K. (1996). *Coloured Petri nets : basic concepts, analysis methods, and practical use*. Monographs in theoretical computer science. Berlin : Springer, Berlin, 2nd ed. edition.

Judge, G. G. and Swanson, E. R. (1962). Markov chains: Basic concepts and suggested uses in agricultural economics. *Australian Journal of Agricultural Economics*, 6(2):49–61.

Knudsen, P., Herborg, H., Mortensen, A. R., Knudsen, M., and Hellebek, A. (2007). Preventing medication errors in community pharamcy; frequency and seriousness of medication errors. *Qual Saf Health Care*, 16(4):291–296.

Kucukkoc, I. and Zhang, D. (2015). Type-E parallel two-sided assembly line balancing problem: Mathematical model and ant colony optimisation based approach with optimised parameters. *Computers & Industrial Engineering*, 84:56–69.

Langley, C. A. (2009). *Applied pharmaceutical practice / Christopher A. Langley, Dawn Belcher.* Fasttrack (London, England). London : Pharmaceutical Press, London, 2nd edition.

Lee, W. S., Grosh, D. L., Tillman, F. A., and Lie, C. H. (1985). Fault tree analysis, methods, and applications; a review. *IEEE Transactions on Reliability*, R-34(3):194–203.

Li, X., Baki, M. F., and Aneja, Y. P. (2010). Ant colony optimization metaheuristic for machine-part cell formation problems. *Computers & Operations Research*, 37(12):2071–2081.

Li, Z. and Pack, S. (2004). An application of markov models in estimating transition probabilities for postmenopausal women with osteoporosis. *Drug Information Journal*, 38(1):41–46.

Lisnianski, A., Elmakias, D., Laredo, D., and Haim, H. B. (2012). A multi-state markov model for a short-term reliability analysis of a power generating unit. *Reliability Engineering & System Safety*, 98(1):1 – 6.

Littlewood, B. (1975). A reliability model for systems with markov structure. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):172–177.

Liu, F., Heiner, M., and Gilbert, D. (2017a). Coloured Petri nets for multilevel, multiscale and multidimensional modelling of biological systems. *Briefings in Bioinformatics*, 20(3):877–886.

Liu, X., Zhang, J., and Zhu, P. (2017b). Modeling cyber-physical attacks based on probabilistic colored Petri nets and mixed strategy game theory. *International Journal of Critical Infrastructure Protection*, 16:13–25.

Liu, Z., Xin, N., Yiliu, L., Qinglu, S., and Yukun, W. (2011). Gastric esophageal surgery risk analysis with a fault tree and markov integrated model. *Reliability Engineering & System Safety*, 96(12):1591–1600.

Lusseau, D. (2003). Effects of tour boats on the behavior of bottlenose dolphins: using markov chains to model anthropogenic impacts. *Conservation Biology*, 17(6):1785–1793.

Mahulea, C., Mahulea, L., Soriano, J. M. G., and Colom, J. M. (2017). Modular Petri net modeling of healthcare systems. *Springer Science & Buisness*, 30(1-2):329–357.

Makajic-Nikolic, D., Petrovic, N., Belic, A., Rokvic, M., Radakovic, J. A., and Tubic, V. (2016). The fault tree analysis of infectious medical waste management. *Journal of Cleaner Production*, 113:365–373.

Marsan, M. A. (1990). Stochastic Petri nets: An elementary introduction. In Rozenberg, G., editor, *Advances in Petri Nets 1989*, pages 1–29, Berlin, Heidelberg. Springer Berlin Heidelberg.

Martens, D., Van Gestel, T., De Backer, M., Haesen, R., Vanthienen, J., and Baesens, B. (2010). Credit rating prediction using ant colony optimization. *Journal of the Operational Research Society*, 61(4):561–573.

Marx, D. A. and Slonim, A. D. (2003). Assessing patient safety risk before the injury occurs: an introduction to sociotechnical probabilistic risk modelling in health care. *BMJ Quality & Safety*, 12(2):33–38.

McCann, L., hughes, C. M., and Adair, C. G. (2009). A self-reported work-sampling study in community pharmacy practice: a 2009 update. *Pharm World Sci*, 32(4):536–543.

Melnikow, J., Birch, S., Slee, C., McCarthy, T. J., Helms, L. J., and Kuppermann, M. (2008). Tamoxifen for breast cancer risk reduction: Impact of alternative approaches to quality-of-life adjustment on cost-effectiveness analysis. *Medical Care*, 46(9):946–953.

Melnikow, J., Kuenneth, C., Barnato, A., Kuppernann, M., Birch, S., and Nuovo, J. (2006). Chemoprevention: Drug pricing and mortality - the case of tamoxifen. *Cancer*, 107(5):950–958.

Mullan, Z., Benham, L., Cumber, H., Dehnel, T., and Clark, S. (2011). To err is human. *The Lancet*, 378(9794):861–861.

Narayanan, M. and Cherukuri, A. K. (2018). Verification of Cloud Based Information Integration Architechture using Coloured Petri nets. *International Journal of Computer Network and Information Security*, 10(2):1–11.

Naybour, M., Remenyte-Prescott, R., and Boyd, M. (2018a). Evaluation of a community pharmacy dispensing process using a coloured petri net. In *Proceedings of the European Safety and Reliability Conference, ESREL*, Trondheim, Norway, June 17-21.

Naybour, M., Remenyte-Prescott, R., and Boyd, M. (2018b). Reliability modelling of dispensing processes in community pharmacy. In *10th IMA International Conference on Modelling in Industrial Maintenance and Reliability (MIMAR)*, Manchester, UK, June 13-15.

Netjasov, F. and Janic, M. (2008). A review of research on risk and safety modelling in civil aviation. *Journal of Air Transport Management*, 14(4):213 – 220.

Neuman, H. B., Elkin, E. B., Guillem, J. G., Paty, P. B., Weiser, M. R., Wong, D. W., and Temple, L. K. (2009). Treatment for Patients with Rectal Cancer and a Clinical Complete Response to Neoadjuvant Therapy: A Decision Analysis. *Diseases of the Colon and Rectum*, 52(5):863–871.

Niu, Q., Peng, Q., and ElMekkawy, T. Y. (2013). Improvement in the operating room efficiency using tabu search in simulation. *Business Process Management Journal*, 19(5):799–818.

NPSA (2007). *Design for patient safety: A guide to the design of the dispensing environment*. National Patient Safety Agency.

O'Connor, P. D. T. and Kleyner, A. (2012). *Practical Reliability Engineering*. John Wiley & Sons.

Ojeleye, O., Avery, A., Gupta, V., and Boyd, M. (2013). The evidence for the effectiveness of safety alerts in electronic patient medication record systems at the point of pharmacy order entry: a systematic review. *BMC Medical Informatics and Decision Making*, 13(1):69.

ONS (2015). Annual survey of hours and earnings. Office for National Statistics.

Oses, N. (2008). Markov chain applications in the slot machine industry. *OR Insight*, 21(1):9–21.

Ostrom, L. T. and Wilhelmsen, C. A. (2012). *Risk Assessment*. Wiley.

Pande, S., Hiller, J. E., Nkansah, N., and Bero, L. (2013). The effect of pharmacist-provided non-dispensing services on patient outcomes, health service utilisation and costs in low and middle-income countries. *Cochrane Database of Systematic Reviews*, 2(2).

Park, A. and Lee, S. J. (2009). Fault tree analysis on handwashing for hygiene management. *Food Control*, 20(3):223–229.

Perraudin, C., Vaillant, M. L., and Pelletier-Fleury, N. (2013). Cost-Effectiveness of a Community Pharmacist-Led Sleep Apnea Screening Program - A Markov Model. *Public Library of Science*, 8(6):e63894.

Petri, C. A. (1962). *Kommunication mit Automaten*. PhD thesis, Technischen Hoschule Darmstadt.

Prescott, D. and Andrews, J. (2015). Investigating railway track asset management using a markov analysis. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 229(4):402–416.

Prescribing and Medicines Team (2017). *Prescriptions Dispensed in the Community, Statistics for England - 2006 - 2016*. Office for National Statistics.

PSNC (2017). PSNC Briefing 040/17: NHS community pharmacy Advanced Services – information for general practitioners and practice staff.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Reason, J. (2000). Human error: models and management. *British Medical Journal*, pages 768–770.

Reay, K. A. and Andrews, J. D. (2002). A fault tree analysis strategy using binary decision diagrams. *Reliability Engineering & System Safety*, 78(1):45–56.

Reed, S., Remenyte-Prescott, R., and Rees, B. (2017). Effect of venepuncture process design on efficiency and failure rates: A simulation model study for secondary care. *International Journal of Nursing Studies*, 68(73-82):2017.

Rice, J. A. (2007). *Mathematical Statistics and Data Analysis*. Thomson Brooks/Cole.

Ross, S. (2010). *A First Course in Probability*. Pearson.

RPS (2011). *Medicines, ethics and practice : the professional guide for pharmacists*. London : Pharmaceutical Press, London, edition 35. edition.

Salimifard, K., Hosseini, S. Y., and Moradi, M. S. (2013). Improving emergency department processes using coloured petri nets. In *34th International Conference on Application and Theory of Petri Nets and Concurrency*, Milano, Italy, June 24-28.

Schafheutle, E. I., Samuels, T., and Hassell, K. (2008). Support staff in community pharmacy: who are they and what do they want? *International Journal of Pharmacy Practice*, 16(2):57–63.

Schneeweiss, W. G. (1999). *Petri nets for reliability modeling: (in the fields of engineering safety and dependability)*. Hagen: Lilole-Verlag.

Selvi, V. and Umarani, R. (2010). Comparative Analysis of Ant Colony and Particle Swarm Optimisation Techniques. *International Journal of Computer Applications*, 5.

Shmygelska, A. and Hoos, H. H. (2005). An ant colony optimisation algorithm for the 2d and 3d hydrophobic polar protein folding problem. *BMC Bioinformatics*, 6(30).

Silva, B. A. D. and Krishnamurthy, M. (2016). The alarming reality of medication error: a patient case and review of pennsylvania and national data. *Journal of Community Hospital Internal Medicine Perspectives*, 6(4).

Smotherman, M. and Zemoudeh, K. (1989). A non-homogeneous markov model for phased-mission reliability analysis. *IEEE Transactions on Reliability*, 38(5):585–590.

Sánchez, A. M. (2013). Medication errors in a spanish community pharmacy: nature, frequency and potential causes. *International journal of clinical pharmacy*, 35(2):185–189.

Somani, S. M., Daniels, C. E., and Jermstad, R. L. (1982). Patient satisfaction with outpatient pharmaceutical services. *American Journal of Hospital Pharmacy*, 39(6):1025–1027.

Sonnenberg, F. A. and Beck, J. R. (1993). Markov models in medical decision making: A practical guide. *Medical Decision Making*, 13(4):322–338. PMID: 8246705.

Stültze, T. and Hoos, H. (1997). MAX-MIN Ant System and local search for combinatorial optimisation problems. In *2nd International Conference on Metaheuristics*, Sophia-Antipolis, France, July 21-24.

Stültze, T. and Hoos, H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems*.

Stojkovic, T., Rose, O., Woltersdorf, R., Marinkovic, V., Manser, T., and Jaehde, U. (2017). Prospective systemic risk analysis of the dispensing process in german community pharmacies. *International Journal of Health Planning and Management*, 33(1):e320–e332.

Tan, P., He, W.-t., Lin, J., Zhao, H.-m., and Chu, J. (2018). *Design and Reliability, Availability, Maintainability, and Safety Analysis of a High Availability Quadruple Vital Computer System*, pages 481–496. Springer Singapore.

Tan, W. S., Chua, S. L., Yong, K. W., and Wu, T. S. (2009). Impact of pharmacy automation on patient waiting time: an application of computer simulation. *Annals academy of medicine singapore*, 38:501–507.

Thode, H. (2002). *Testing For Normality*. Statistics, textbooks and monographs. Taylor & Francis.

Vesely, W., Dugan, J., Fragola, J., Minarick, J., and Railsback, J. (2002). Fault tree handbook with aerospace applications. NASA Office of Safety and Mission Assurance, NASA Headquarters, Washington, DC 20546.

Vesely, W. E. and Narum, R. E. (1970). PREP and KITT computer code for the automatic evaluation of a fault tree. Technical report, Idaho Nuclear Corporation.

Vila-Parrish, A. R., Ivy, J. S., King, R. E., and Abel, S. R. (2012). Patient-based pharmaceutical inventory management: a two-stage inventory and production

model for perishable products with markovian demand. *Operational Research Society*, 1(1):69–83.

Viswanadham, N. and Narahari, Y. (1987). Coloured petri net models for automated manufacturing systems. In *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, volume 4, pages 1985–1990.

Wang, L.-C. (1996). Object-oriented petri nets for modelling and analysis of automated manufacturing systems. *Computer Integrated Manufacturing Systems*, 9(2):111 – 125.

Waterfield, J. (2008). *Community pharmacy handbook*. London : Pharmaceutical Press, London.

Weisstein, E. W. (N/a). Log normal distribution. MathWorld - http://mathworld.wolfram.com/LogNormalDistribution.html.

WHO (2012). Safer primary care a global challenge summary of inaugural meeting. World Health Organization.

WHO (2016). Medication errors: Technical series on safer primary care. World Health Organization.

Wilk, M. B. and Gnanadesikan, R. (1968). Probability plotting methods for the analysis for the analysis of data. *Biometrika*, 55(1):1–17.

Wreathall, J. and Nemeth, C. (2004). Assessing risk: the role of probabilistic risk assessment (pra) in patient safety improvement. *BMJ Quality & Safety*, 13(3):206–212.

Xin-She, Y. (2007). *Introduction to Mathematical Optimization : From Linear Programming to Metaheuristics*. International Science.

Xiong, H. H., Zhou, M., and Manikopoulos, C. N. (1994). Modeling and performance analysis of medical services systems using petri nets. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2339–2342 vol.3.

Yeh, J.-Y. and Lin, W.-S. (2007). Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department. *Expert Systems with Applications*, 32(4):1073–1083.

Yu, J., Shah, B., Ip, E., and Chan, J. (2013). A Markov Model of the Cost-Effectiveness of Pharmacist Care for Diabetes in Prevention of Cardiovascular Diseases: Evidence from Kaiser Permanente Northern California. *Journal of Manages Care Pharmacy*, 19(2):102–114.

Zecchin, A. C., Simpson, A. R., Maier, H. R., and Nixon, J. B. (2005). Parametric study for an ant algorithm applie to water distribution system optimisation. *IEEE Transactions on Evolutionary Computation*, 9(2):175–191.

Zegordi, S. H. and Davarzani, H. (2012). Developing a supply chain disruption analysis model: Application of colored petri-nets. *Expert Systems with Applications*, 39(2):2102 – 2111.

Zheng, X., Bolton, M. L., Daly, C., and Feng, L. (2017). A formal human reliability analysis of a community pharmacy dispensing procedure. 61(1).