Petri Net Modeling of Outpatient Waiting Time for MRI Examination

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Division of Biomedical Engineering
University of Saskatchewan
Saskatoon

By
Hengyao Lu

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Division of Biomedical Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5A9

Abstract

In Canada, access to magnetic resonance imaging (MRI) examination is limited with an outcome of long patient waiting time. It is reported that the current median waiting time for MRI examination in Saskatoon is almost double the target for waiting time, which may aggravate the disease.

This research is towards reducing the waiting time of patients for MRI examination in Canada by applying an improved management. As a first step of this effort, a comprehensive model of MRI booking and serving system is needed. The city of Saskatoon was taken as an example and the MRI booking and serving system in the city was studied. The common tools (queuing theory, system dynamics (SD) and discrete event dynamics simulation (DES)) were compared and it is found that DES is more suitable, in particular Petri nets (PNs), deemed to be the best choice for the purpose of this thesis.

The model in this research was constructed on the basis of Hierarchical Coloured Petri nets (HCPNs), a combination of two extended PNs: Coloured PNs (CPNs) and Hierarchical PNs (HPNs). The model is able to simulate and predict patients' waiting times. Given that the structure of the model developed by HCPNs is still too complex, two extensions to CPNs, Ordered CPNs (OCPNs) and Prioritized HCPNs (PHCPNs), were proposed in this study to reduce the complexity of the model. Validation of the model was performed using the data of Saskatoon Health Region - Royal University Hospital. The results have shown that the proposed model can effectively describe the real system.

The model has potential applications in decision-making for the selection of an optimal booking strategy to shorten waiting time and in the prediction of possible waiting time of the system in the future, which may assist MRI administrators in the management of medical resources and may greatly improve the quality of MRI service.

Acknowledgements

I would like to express the deepest appreciation to my supervisor, Prof. W.J. (Chris) Zhang for his esteemed guidance and endless support during this thesis work. I am also grateful to him for the financial support during the master program.

I would like to express my heartfelt gratitude to my co-supervisor, Prof. Paul Babyn for his great help in the progress of this research and priceless suggestions to my work.

I would also like to thank my committee members, Prof. FangXiang Wu and Prof. Sonia Vanderby for their valuable suggestions during my committee meeting.

In addition, I am thankful to Ms. Porter for the patient introduction of MRI department provided by her and to Mr. Dagenais for his help in the data collection.

Last but not the least, I am grateful to my family members for their support and constant belief in me.

Dedicated to my beloved parents,

Jianhao Lu and Qian Ye,

And my fiance,

Lizhi Liu,

Without whom none of my success would be possible

Contents

List of Tables

List of Figures

List of Abbreviations

MRI	Magnetic Resonance Imaging
RF	Radio Frequency
PN	Petri Net
DES	Discrete Event Dynamics Simulation
SD	System Dynamics
CPN	Coloured Petri Net
HPN	Hierarchical Petri Net
HCPN	Hierarchical Coloured Petri net
OMS	Qrdered Multi-Set
OCPN	Ordered Coloured Petri Net
PHCPN	Prioritized Hierarchical Coloured Petri Net
FIFO	First In First Out
LIFO	Last In First Out
WAWT	Weighted Average Waiting Time

Chapter 1

Introduction

1.1 Background

Magnetic resonance imaging (MRI) is a medical imaging technique to produce images of body tissues via the use of a powerful static magnetic field, radio frequency (RF) signals and particular computer software. In comparison with other X-ray based traditional medical imaging techniques (CT, X-ray), MRI does not use ionizing radiation. As the human body is predominantly made up of water, hydrogen protons are common. As the proton rotates on an axis, it produces a tiny magnetic field. Normally, the orientation of these magnetic fields are randomly distributed. However, when a patient is placed within a strong magnetic field, protons in human body may align with or against the direction of the external magnetic field. Because the number of protons aligning with the direction of the external field is greater than those aligning against it, the net result is an alignment with the external magnetic field. An RF pulse is further added, and the energy of RF is absorbed to alter the alignment of the protons. When the RF pulse is turned off, the absorbed energy is released and the direction of the protons returns to the original state. This returning process emits RF signals, which are then detected by a device and computed to render an image. Abnormal organs or tissues generate different signal, which is the basis for analysis and diagnosis of any abnormality.

MRI examination is now a non-invasive imaging modality for diagnosis allowing safely capture of high-resolution images of the body tissues and organs. It is able to scan any part of the body in any plane and orientation.

In Canada, if general practitioners or specialists need the assistance of MRI to make a diagnosis, they need to at first transmit their request for MRI examination to a booking center, providing the basic information of the patient (e.g. name, gender and medical history) and the referral. After the request is received, the attending radiologist may consult with the physician to determine the priority level and confirm some other information. After that, booking staff will schedule the patient and notify the patient of the date and time of MRI examination with pertinent information about the examination.

There are four MRI units in Saskatoon, and they are currently located in three hospitals: two in Royal University Hospital (RUH), one in Saskatoon City Hospital (SCH) and one in St. Paul's Hospital (SPH). All bookings are however accepted and scheduled at RUH only. That is to say, there is only one point of contact for booking for MRI examination in Saskatoon.

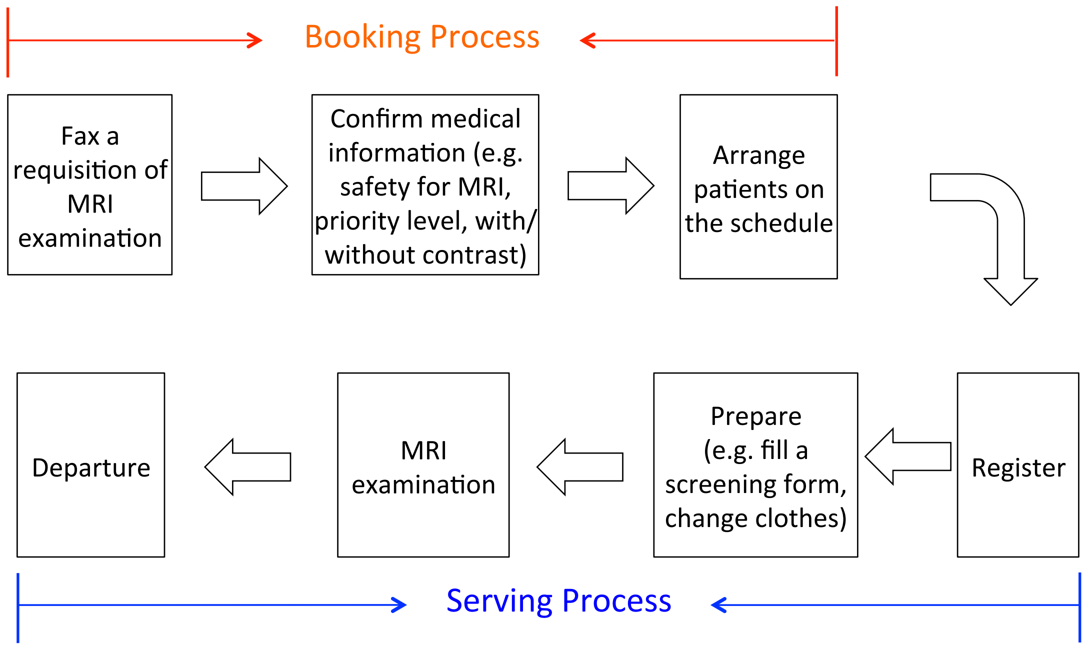

Figure 1.1: The whole process for outpatients to experience in MRI exam

On the day of examination, the patient is asked to arrive early at the MRI department typically 45 minutes prior to the examination and to register at the reception desk first. When the patient reports to the MRI department, he/she is asked to complete a screening form which is intended to further confirm whether he/she is ready and able to safely proceed with the examination. The subsequent procedures that he/she may experience are explained as well. Then, the patient is guided to change into a patient gown, leaving anything that may affect the work of MRI (e.g. keys, watches, cell phones, credit cards, etc.) in the changing room. After this, the patient waits in the waiting room. In this period, additional preparations are made as needed (e.g. intravenous injection). When the scanning starts, the patient needs to lie on the scan table and be covered with a coil to transmit/receive the RF signals, and then he/she is moved into the cylinder. The scan procedure lasts 20 minutes or longer, during which the patient has to keep still as any movement may lead to a restart of imaging. Some ancillary resources are offered to help the patient calm down and relax (e.g. movies, music, etc.). For those who cannot stop moving, sedation or anesthetic is usually used. In some cases, a contrast agent is required to highlight specific tissues. After the examination, the patient leaves the scan room, changes back into street clothes and is free to go home. If a patient is anesthetized, he/she is transferred to another area for recovery. The exam results are analyzed and reported by a radiologist and transmitted to the general practitioner or specialist who made the referral to the radiologist for the patient. The scheduling and examination process for outpatients (the patients who visit a hospital for diagnosis or treatment without staying overnight) undergoing MRI are shown in Figure 1.1. For inpatients who are admitted to live in hospital during treatment, they do not need to book the MRI scan; instead, the doctor calls the department for the service. A period of time is reserved for inpatients each day.

1.2 Motivation

As an important diagnostic technique, MRI is increasingly used for diagnosis in clinical medicine today. In Canada, the number of MRI scanners in 2010 was 281, almost doubling from the number (149) in 2003, and the number of patients scanned by MRI (in Canada) keeps increasing as well in recent years from 768,302 in April 2003 to almost 1,434,500 in October 2009 [1].

Despite the annual growth in the number of scanners and tests, access to MRI examination in Canada is far lower than many other developed countries. The report of *OECD Health Data 2010* shows that in Canada, the rate of MRI scanners per million population in 2010 was 8.24, far lower than the rate of 31.52 in the US, 22.55 in Greece and 22.02 in Italy. Besides, the data indicate that the number of MRI users per 1,000 populations in 2010 in Canada is around 46.3, which means that one scanner is shared by 5,624 patients per year, the figure being much higher than 737 in South Korea, 3,097 in the US and 3,099 in Poland [2]. Owing to the high requirement of MRI examination and reduced availability of machines in Canada, Canadians experience a longer waiting time for the MRI examination service.

There are several definitions for MRI waiting time and they need to be clarified prior to developing a model. One standard definition is the time between the date of ordering the MRI examination and the date of receiving the examination [51]. This thesis has taken this standard definition of waiting time. It should be noted that patients waiting for MRI examination may be subdivided into four groups based on the classification of Saskatoon Health Region: emergency (level 1), urgent (level 2), semi-urgent (level 3) and non-urgent (level 4). The MRI waiting times for patients in the latter three groups (as level 1 cases need to be done within 24 hours) in Saskatoon in 2011 are statistically measured [3], and the result is as follows: the median waiting times (within which 50% cases are completed) for level 2, 3 and 4 patients are 14, 77 and 95 days, respectively; while the target waiting times of the three groups of patients are 7, 30 and 90 days, respectively [3]. Though the current delay of access to MRI in Saskatoon for level 4 patients is slightly longer than the target, the delay for other patients is double the expected standard.

1.3 Problem Statement

The general problem of the present study is how to develop an accurate model for MRI examination, capturing as many real situations and features as possible, and demonstrate its potential to be used in improving MRI booking and serving management to reduce waiting time. The city of Saskatoon was taken as an example to assist in this study. Further, management here refers to planning and scheduling. The goal of planning and scheduling is to achieve a minimal idle time of the MRI scanner and a healthy workload of both medical personnel and patients. It is noted that the entire system is a kind of human-machine system. That is to say

[1] http://www.hc-sc.gc.ca/hcs-sss/pubs/system-regime/2010-fed-comp-indicat/index-eng.php
[2] http://stats.oecd.org/index.aspx?DataSetCode=HEALTH_STAT
[3] http://www.health.gov.sk.ca/diagnostic-imaging-mri-wait-times

in particular, the MRI scanner can be considered to be a part of the "team", along with all the staff (e.g. operators, nurses and patients). The two parts should be considered together.

The challenge to have a better plan and schedule includes: (1) different machines at different locations, (2) uncertainty in terms of the length of examination (due to individualized difference between patients, between nurses and between operators), (3) changes in booking, and (4) uncertainty in machine condition. It should be noted that different patients may need different lengths of times for MRI examination.

1.4 Research Objectives and Scope

To solve the foregoing problem, an effective way is to build a model of the system and analyze the system based on the model. The system in this case is the MRI booking and serving system (MRI system for short), which includes patients, medical staff, scanners, and their interactions. However, this particular work is restricted to patients and patient flow only. The whole process of patient flow was considered, which is divided into two successive processes: booking process and serving process (Figure 1.1). In the booking process, a patient is scheduled with a date, a start time and an anticipated end time. In the serving process, the examination is performed, which may end earlier or later than anticipated. Any case that the patient ends beyond the scheduled time may cause some rescheduling of the remaining patients in that day.

The overall objective of this study is towards a solution to the problem of long waiting time in Saskatchewan through an improved scheduling. The thesis focused to develop a model for the MRI booking and serving system, which is able to examine the existing planning and scheduling principles and procedure and to assist in developing a potentially better plan and schedule in the future.

The specific objectives of this work are as follows:

Objective (1): Construct a model to describe the whole process of MRI examination and to quantify the waiting time of patients.

Objective (2): Investigate the effectiveness of the model and its potential applicability in the context of planning and scheduling of MRI examination in terms of simulation and predication of the waiting time of patients.

1.5 Research Methodology

The whole process of MRI system was learned through a period of observation in RUH in Saskatoon. Based on the characteristics of MRI system, a discrete event based graphical simulation and formulization tool, Petri nets (PNs), was selected as the modeling tool to capture the dynamic behavior of the system. To the best of the author's knowledge, this is the first attempt to use PNs to model the flow of patients with MRI. There are many extensions to PNs. Taking account of the scale and data organization of the process of MRI system, one extension called Hierarchical Coloured Petri nets (HCPNs) that combine Coloured PNs (CPNs) and Hierarchical PNs (HPNs) was taken to build the present model. Given that the structure of model developed by the existing extensions of PNs is still too complex, two extensions to CPNs, Ordered CPNs (OCPNs) and Prioritized HCPNs (PHCPNs), were proposed in this study to reduce the complexity of the CPN model. With these extensions, some functions of the model can now be achieved straightforwardly rather than by complicated transformations, so the size of model can be reduced. As a validation of the proposed model, simulated results were compared with the real data acquired from the database of Saskatoon Health Region - Royal University Hospital.

1.6 Organization of This Thesis

This thesis is composed of six chapters. The remaining five chapters are laid out as follows:

Chapter 2: This chapter provides a literature review of the existing research methods of managing the patient flow in health systems. Their characteristics are described, and their applicable scopes and limitations are discussed. Through the comparisons of different methods, the reason for the selection of PNs as the modeling tool in this research is further given.

Chapter 3: This chapter introduces the background of PNs and their extensions and justifies why the extended PNs, namely HCPNs are used to develop the model for patients and the patient flow. To further reduce the complexity of the model, two new extensions to CPNs, OCPNs and PHCPNs, are defined, and they are employed in the model development.

Chapter 4: In this chapter, six pieces of semantics are described, which need to be modeled. The corresponding PN models for Semantics 1 to 4 are developed. As a foundation of other models, the model for Semantics 1 is validated using the real data acquired from the database of Saskatoon Health Region - Royal University Hospital. The models for the other three pieces of semantics are not validated due to the unavailability of data.

Chapter 5: This chapter demonstrates the potential applications of the proposed PN model with examples.

Chapter 6: The last chapter draws a conclusion from the above work and discusses the contributions and future work of this research. In particular, modeling of the remaining two pieces of semantics is outlined.

Chapter 2

Literature Review

To model and analyze the behavior of a health care system and especially patient flow, various approaches have been proposed in the literature, amongst which the most popular are queuing theory and process simulation. This chapter gives an overview of these approaches. In Section 2.1, queuing theory is introduced with its characteristics, applications and limitations. Section 2.2 focuses on the description of process simulation, which is divided into two subsections: system dynamics (SD) and discrete event dynamics simulation (DES). Further, a specific DES, Petri nets (PNs), is presented, and the reasons for choosing DES, especially PN, for the present study are described, with a conclusion in the last section.

2.1 Queuing Theory

The basic ideas of queuing theory are: (1) the customer arrives at a system; (2) the customer waits in queue; (3) the customer is served according to a pre-defined rule (e.g. first-come-first-serve, random). A classical queuing model is $M/M/s$. In $M/M/s$, the first M represents the inter-arrival time of the customer that follows an exponential distribution, the second M represents the service time that follows exponential distribution, and s is the number of servers. With this model, one can determine the probability of n customers in the queuing system (n is from 0 to the upper limit of the queue size), expected number of customers in queue or in the whole system (including those who have been served) and expected waiting time in queue or in the whole system for an individual customer. The model assumes that the queuing system is in a steady state depending neither on the initial state nor elapsed time [28].

Queuing theory has been employed in health care systems for decades with two most popular applications for resource scheduling and patient scheduling. Resource here primarily refers to beds [16, 45, 39, 22] and staff [64, 57, 24, 23]. The allocation of beds should be well determined, as otherwise the scarcity of beds may cause a block for patients to receive cares. For example, De Bruin et al. [16] used a queuing model to optimize the bed allocation in different levels of a cardiac care chain with an upper limit of refused admissions and found that the refused admissions at the First Cardiac Aid mainly resulted from a lack of available beds downstream in the care chain. McManus et al. [45] modeled the patient flow in an intensive care unit (ICU) and measured the turn-away rates. By the comparison of the simulated and observed data, they concluded that queuing theory is effective in modeling ICU bed utilization and providing an indication of an appropriate bed supply in a large system which operates close to its capacity. Besides beds, the shortage of nurses is another important reason for overcrowding in the health care system. Yankovic and Green [64] proposed a finite source queuing model with two servers (beds and nurses) to guide the nurse-staffing decision. The result has shown that the nurse-staffing level is affected by unit size, nursing intensity, bed occupancy and unit length-of-stay. Another finding is that the common criterion used for staffing, minimization of nurse-to-patient ratio, is unreliable, leading to either understaffing or overstaffing. A similar result has been discovered in the work of De Vericourt and Jennings [57], who presented a closed queuing system to investigate the optimal number of nurses and found that a fixed nurse-to-patient staffing ratio is ineffective for nurse staffing and the total number of patients should be taken into consideration.

Patient scheduling concentrates on management of patient flows in an appointment system, in which patient waiting time is chiefly impacted by appointment rules and methods. To control the waiting time for patients to stay at home before the appointed time or to stay at a clinic before the treatment, models based on queuing theory were developed in previous research [6, 1, 10, 17, 56, 13, 43]. For instance, Vasanawala and Desser [56] applied queuing theory to predict an appropriate number of time slots reserved for urgent requests in computed tomography (CT) and ultrasonography (US) with a premise that the rescheduling rate of routine cases was below a certain level. Creemers and Lambrecht [13] constructed a queue model with combination of a vacation queuing system and an appointment system to evaluate the performance of a health care system, including the size of a waiting list, waiting time at the service facility and server overtime. An application system was used to verify the model. Mardiah and Basri [43] studied the factors leading to the delay for patients to be treated in an outpatient clinic with a queue model. The results indicate that the current appointment strategy, "doctor on call", is not valid, which decreased doctor's idle time at the cost of long patient waiting time.

Queuing theory has some important limitations. First, the assumptions of queuing theory are too rigorous, narrowing down the scope of its applications. Kolker [40] summarized that the following situations are inappropriate for current queuing theory: (1) a queuing system with an unlimited queuing size (i.e. the length of a queue keeps increasing) does not have a steady state (e.g. the mean arrival rate of patients is larger than the mean service rate of all the servers); (2) patients leave the queue after waiting some time or patients jump the queue for emergent requesting for a care; (3) patients' mean arrival rates are a series of variables; (4) two situations cannot be distinguished when their service time follows different distributions with yet the same average; (5) the patients with the random and scheduled arrival time are combined into one system. It is noted that in the MRI system, it is possible that a patient cancels the appointment, and thus the available scheduled time may be occupied by another patient. Queuing models may have difficulty to capture such real life situations.

Second, the method is not easy to be used in cases where several service facilities are involved in sequence. The typical queuing models usually consider one service facility with one or more servers. However, in health care systems, patients often join more than one queue of different units to wait for various tests or treatments.

The service rate in each unit is probably different and affects the arrival rate of the subsequent units. The complexity of such system makes such a problem intractable by queuing theory. Finally, queuing theory fails to make an intuitionistic description of the original system or a description that has a clear distinction between the structure and behavior of a system. It is noted that a clear distinction of the notions such as structure, behavior and so on is a necessary condition to make an intelligent system according to [65], while queuing theory seems to operate at the behavioral level.

2.2 Process Simulation

Process simulation is the imitation of a real-world event or process with the assistance of computer software. The purpose of simulation is to describe a system dynamically and answer "what-if" questions based on the system. In comparison with queuing theory, process simulation is more flexible and tractable since no distribution assumption is needed for the inter-arrival and service time, and more details of the system can be considered in simulation with less worry about the tremendous increase of the complexity of the problem. There are two streams of process simulations: system dynamics (SD) and discrete event dynamics simulation (DES).

2.2.1 System Dynamics (SD)

System dynamics is a tool to explore how the state of a system changes over time. The core components of SD are feedback loops, levels and rates. A feedback loop exists when information runs through a system and finally returns to the initial point, which takes a potential effect on the future action. Levels are the amount of stored resources or information, and rates are the speed of accumulation of the stock. The former are sources of the dynamic behavior of a system, whereas the latter are operating policies. The resource or information moving through the system is modeled as a crowd with no consideration of individuals in a flow. A model with all the components is usually illustrated in a causal-loop diagram. In SD, the change of state is continuous, and can be discretized into difference equations by partitioning a whole period of time into small time intervals.

In process simulation with SD, the structure of a model is composed of essential process stages and other factors that have impact on the process stages (positive or negative). The linkages among the entities are set to show the interactive relationships, making some feedback loops to influence the performance of the system iteratively until a steady state is attained or the iteration is ended by some conditions (e.g. one resource is used up). For the same system of interest, SD models may differ from one to another, as there are factors contributing to dynamics which are selected subjectively by modellers. A SD model is more suitable to produce a qualitative analysis of system performance, the result of which can be treated as a guideline for the decision makers to evaluate a strategic policy.

According to Vanderby [55], there are several utilizations of SD in health care system, including:

- Wait list [60, 62],
- Demand for social services [18],
- Safety and quality issues [12, 54],
- The impact of new technology on health care [8, 3],
- Human resource concerns [37],
- Disease progression and prevalence [11, 15], and
- Disaster planning [2, 29].

Furthermore, the patient flow can be modeled in some situations with consideration of capacity and delay [55]. Wolstenholme et al. [61] developed a SD model to show that the actual utilization of health care in UK exceeded the designed capacity, which would lead to severe consequences. In their model, the admitted patient flow was traced from primary care to hospital, and the length of service was taken into account. Vanderby [55] developed a whole hospital model to show that the SD model is suitable to model the patient flow in the emergency department (ED) (before, during and after ED treatment). The patients were grouped with age cohort and Canadian Triage Acuity Scale (CTAS) level. By analysis of the model, the length of stay (LOS) at each stage of a care chain was obtained, and then the total LOS was measured.

Although SD has had some successful applications in the patient flow modeling, it is not suitable to model the patient flow in a system for the event-driven and discrete-time dynamic characteristics (e.g., the patient appointment system). In a patient appointment system, the profile of the patient is an important piece of information that may affect the behavior of the appointment system. The SD model has difficulty in capturing that individual information. Another reason that SD is unsuitable for modeling of a patient appointment system is that patients will show up to the health care system in a pre-arranged manner from an individual patient's point of view. Because a SD model takes a point of view of flow of a "group" of patients, which is a random variable perceived by a particular hospital, a stochastic model for the admission rate of patients needs to be assumed. That is to say, the nature of the original problem is altered to a problem with "artifacts" due to the use of SD, a flow view in particular, which is prone to errors or inaccuracies with a SD model. In short, SD is a tool for modeling of a continuous time flow, and it thus has an inherent difficulty in modeling any discrete event dynamics where individuals in a flow may significantly affect the dynamics of the entire flow.

2.2.2 Discrete Event Dynamics Simulation (DES)

Discrete event dynamics simulation (DES) is a tool to model the behavior of a system where a series of events may occur at random time points. There are two essential notions in defining a DES: a discrete state space and a discrete event set. The event set logs all the events that a system experiences, while the state space tracks state variations in each node of the network. In a DES model, the state updates when a new event happens (e.g. a patient comes) or an operation is carried out (e.g. a treatment is finished). The objects moving through a DES system are individuals distinguishable from each other, that is, an individual's behaviors are captured, so the DES model more suits systems such as the appointment system.

The modeling of health care systems with DES has a long history and is still in evolution. A survey of DES [38] shows its application to health care systems, including management of patient flow [25, 52, 50, 48] and optimization of the resource allocation [44, 59, 20, 4]. Patient flow can be evaluated by patient throughout, patient waiting times, length of stay, clinic overtime, staff utilization rates and physician idle times. An adequate resource allocation is the foundation of improving the quality of health care under limited resources.

Comparatively, DES is more appropriate to modeling the MRI system with three reasons. First, DES can facilitate the visualization of the process dynamics especially at the individual patient level. Second, health care systems are in essence event-driven and time-discrete, to which DES is a natural fit. Third, the individual differences in patients can be distinguished and explicitly represented, so as to make the dynamic model more accurate.

Petri Nets (PNs)

A DES model can be constructed with a variety of formalisms, one of which is Petri nets (PNs). PNs, a graphical tool with a powerful mathematical foundation to simulate discrete event dynamic systems, are widely employed in many areas, such as queuing, communication, traffic and manufacturing systems. The basic elements in PNs are “place”, “transition”, “arc” and “token”, which represent a condition, an event, linkage of the condition and event, and resource distribution, respectively. To meet the practical requirements, various extensions of PNs have been proposed. A great number of commercial software packages are now available to build up and run PN models (e.g. CPN Tools, STPN Play, Snoopy). They are developed under different operating systems and may support part of these extensions.

PNs can be regarded as an improvement of the ordinary DES in the following aspects. First, PNs have a precise defined set of tools or entities (e.g. places, transitions, arcs, tokens). The functions of these tools have been realized as function modules in the most existing PN software, which makes the modeling of any event-driven and discrete-time dynamic system formal and easy. Second, PNs have a solid foundation of mathematics, which is not included in most DES models. The ordinary DES just composes a model based on the original system. The problems in the system can only be diagnosed when they appear in simulation. But for PNs, the problems may be exposed as early as the model is completed since the properties of the model can be analyzed. PNs are actually an excellent combination of mathematics and computer science. In PNs, all components, states and behaviors have formal definitions and can be stated with a mathematical language. Some properties of the PN model are deduced and formed as theorems to guide analysis of the system. The properties can be applied to detect and prevent the abnormal situations (e.g. deadlock) prior to the computer simulation. Finally, PNs have numerous extensions, which allow realization of various functions. For example, Coloured PNs extend the function of objects moving through a system and allow them to load more information. Stochastic PNs extend the function of PNs by involving probabilistically specified delays.

Even though the DES has a prevalent utilization in health care, the usages of the PN and its extensions in this field are still not common. The original version, basic PN, is usually used to model the control of systems with small scale and less complexity. Xiong et al. [63] built ordinary and temporal PNs for an emergency medical services system to analyze the performance and improve the efficiency of the system, in which two cases were discussed: patients with and without priority. It is an early attempt to model a health care system with PNs. A similar study appears in [14]. To ease the ambulance divert, Criswell et al. modeled patient flow in ED with a two-part PN (one for Fast track and the other for "normal" ED). However, the model is too large with many details, which makes its structure complex and difficult to be reused.

Given that basic PNs can hardly meet the growing requirements in practice, extended PNs have been proposed and come into use. A Timed PN model was developed by Efstratia et al. [19] to seek solutions to the reduction of delays during the final inpatient charge in a payment's office. After simulation, it is discovered that the system performance is improved when the time period of charge for both medication and MSES (Medical and Surgical Equipment and Supply), or for surgeries, is shortened. The model can be embedded in larger and more complex nets. Bahi-Jaber and Pontier [5] addressed that PNs can be used in the illustration and analysis of epidemic models. They explored the elements affecting the extinction of directly transmitted infectious diseases via Coloured Stochastic PNs. The result has shown that, though an epidemic is doomed to vanish eventually, the time to extinction of the disease may be tremendous, extending with the population volume, the contact rate and the probability of disease transmission during a hazardous contact. In the research of Leite et al. [42], Stochastic PNs were applied to simulate the medical care in the Intensive Care Unit (ICU). Four scenarios were discussed through the comparison of the number of tokens in distinct places. Their paper [42] concluded that the process performance is influenced by the admission and discharge rate and efficiency of the doctor and the nurses, rather than the number of people waiting to occupy a bed in the ICU. Whittaker et al. [58] introduced a choice-point net of a long-term care facility for elderly people to analyze outbreak management protocols, in which the firing time of transitions is deterministic and the outcomes of firing transitions are attached with probability. This model is helpful to answer some questions about protocols. Jansen-Vullers and Reijers [34] paid attention to the reengineering of the intake process for new patients seeking non-urgent cure at a mental health care institute. The initial and redesigned systems are modeled by two Coloured PNs (CPNs) with the addition of a monitor to analyze the models. Comparing with the original system, the new one is advanced with the reduction of both flow time and service time. They declared that CPNs are suitable to modeling business process redesign.

The existing applications of PNs in medical imaging are very limited. Guglielmino et al. [26] applied a set of visual diagrammatic language (VDL) techniques to model a radiology department service, in which PNs were selected as the simulation tool. The patient flow of radiology department was modeled and the flow time of daily exams was measured. Based on the PN model, the performance of two scenarios with different resource allocations was compared and the better scenario was determined. Tavares et al. [53] developed a Stochastic PN model to describe a CT service. The execution times of the imaging acquisition and the medical report were measured. This model was also used to evaluate different scenarios and help picking out the best one. In these two studies, PNs are successfully applied to model the process of medical imaging systems and guide to select a best scenario from alternative ones based on some criteria. A main problem of both models is that they fail to capture specific patient information.

For each example above, the model scope is strictly limited since it is impossible and impractical to assemble all stages and resources into a model. Too many details can easily increase the complexity of a PN model, resulting in more difficulty in framework construction, data collection and model operation. To mitigate the complexities, it is not surprising that PN models are usually built for a specific department (e.g. ED, ICU, CT) or a particular disease (e.g. an epidemic). However, excessive simplifications are not allowed as well in that the nature of a system may be distorted and the validity of the model cannot be guaranteed. Thus, a trade-off between the complexity and validity of a model has to be made. The model in this research focuses on the process of the MRI department and some key booking rules are considered in scheduling of patients to make sure that the process modeling is valid.

A big problem in PN modeling is that the complexity of a model grows along with the increasing scale of the system, that is, the dimension of net modeling a large system may become intolerably huge. To lighten the severity of this problem and broaden the applications of PNs, various extended PNs are proposed by combining or extending functions in different degrees, two of which are selected here to model the MRI system. Two new extensions are proposed to reduce the complexity of the structure of the net, which are introduced in further detail in Chapter 3.

2.3 A Special Note to the Application of Simulation Techniques to Radiology Department

There have been few works on using the discrete event simulation technique to the department of radiology. Ramakrishnan et al. [48] used a general purpose software tool called Arena to model the process of radiology department. Arena is based on discrete event dynamics. They demonstrated the promise to use this simulation technique to identify the changes that may be needed in a particular radiology department, e.g., increasing the number of reading radiologists, etc. However, Arena seems to be based on queue theory, which therefore limits its use for more complex discrete event dynamics. In a similar vein, the tool called Tecnomatix Plant Simulation software was used by Johnston et al. [36]. This tool is also based primarily on queue theory. A comprehensive simulation study was performed by Guglielmino et al. [26] by employing a suite of tools including flow charts, IDEF0 diagrams and Petri Nets. These tools correspond to different activities in the entire business process of radiology department. Their work led to a finding of reorganizing nurses in the department.

2.4 Conclusion

This chapter reviews the literature to discuss the common modeling methods employed for managing health care systems. Both characteristics and application areas of the methods are discussed. It can be concluded that queuing theory is not an ideal choice for this study. In process simulation, SD specializes in modeling of time-continuous dynamic systems and is unsuitable to modeling an appointment system, while DES is illustrated to show its promise in modeling an appointment system; in particular, PNs are deemed to be the best choice for modeling the problem of interest in the present study. The complexity of a PN model needs to be taken into account. It is going to be shown in later discussions that the present study also found two extensions to PNs to reduce the complexity of the model in order to easily capture some pieces of semantics arising from the MRI booking and serving system.

Chapter 3

Petri Net Tool and Its Extensions

Utilizing the motivation established in the last chapter, Petri nets will be used to model and simulate the MRI booking and serving system (MRI system for short). This chapter will present details of Petri Nets. In particular this includes the state of the art of Petri nets described in Section 3.1, along with two extensions proposed by the author of this study to existing Petri nets theory with their justification to the MRI system detailed in Section 3.2. Section 3.3 is a conclusion.

3.1 Petri Nets

3.1.1 Basic Petri Nets

A Petri net (PN) is in its nature a type of directed graph that captures the states and behaviors of discrete event-driven dynamic systems. The PN was first proposed by Carl Adam Petri [47]. It can model various types of discrete event dynamics (e.g. concurrent, asynchronous and non-deterministic).

A PN is composed of two types of nodes: place and transition (see Figure 3.1).

- Place: a circle, denoted by p, representing a condition with respect to the occurring of events.
- Transition: a rectangular or a bar, denoted by t, representing an event or a transformation process.

The connection between a place and a transition is by means of a directed arrow called arc. An arc links two different kinds of nodes, i.e. either from a place to a transition or from a transition to a place (see Figure 3.1). A positive integer k labeled on an arc is called weight, which is interpreted as k numbers of arcs with an identical direction. For a unit weight (i.e. $k = 1$), the label is usually omitted. The dots located at a place are called tokens (see Figure 3.1), showing the presence of material or information resources. The number of tokens in a place indicates the amount of resources. The distribution of tokens over all places in a net is called a marking, which is a vector denoted by M. The dimension of a marking is equal to the total number of places in the net.

The formal definition of a basic PN is as follows:

Definition 3.1 ([46]) *A **Petri Net** is a five-tuple, $PN = (P, T, F, W, M_0)$, where:*

1. $P = \{p_1, \ldots, p_m\}$ *is a finite set of places.*

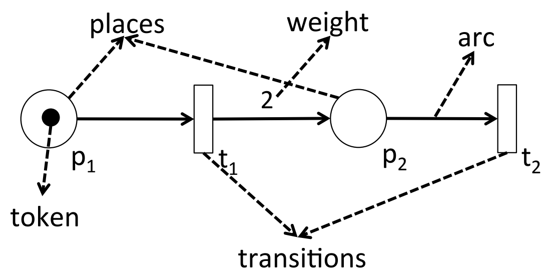

Figure 3.1: An example of a basic PN

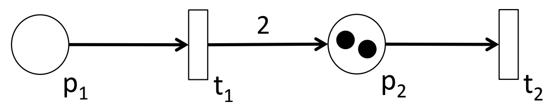

Figure 3.2: The Petri net after the firing of t_1

2. $T = \{t_1, \ldots, t_n\}$ *is a finite set of transitions.*

3. $F \subseteq (P \times T) \cup (T \times P)$ *is a set of arcs (flow relation).*

4. $W : F \rightarrow \{1, 2, \ldots\}$ *is a weight function.*

5. $M_0 : P \rightarrow \{0, 1, 2, \ldots\}$ *is the initial marking.*

6. $P \cap T = \phi$ *and* $P \cup T \neq \phi$.

A Petri net structure $N = (P, T, F, W)$ without any specific initial marking is denoted by N. A Petri net with the given initial marking is denoted by (N, M_0).

A Petri net can be used to simulate a dynamic system via the movement of tokens from place(s) to place(s), which is fulfilled by the firing of transitions. A transition is going to fire when it is enabled, i.e., the preconditions for an event to occur are satisfied. The condition for a transition t to be enabled is that the number of tokens in every input place p of t is not less than the weight of the arc directing from p to t. The firing of a transition removes token(s) from input places and adds token(s) to output places. The number of removed or added tokens is equal to the weights of the corresponding incoming or outgoing arcs of the transition. For instance, in Figure 3.1, the transition t_1 is enabled in that the number of tokens in the input place p_1 is 1, which is equal to the weight marked on the incoming arc of t_1, while the transition t_2 is not enabled. The initial marking M_0 is $[1, 0]$. Because the weights of arcs from p_1 to t_1 and from t_1 to p_2 are 1 and 2, respectively, when t_1 fires, one token is removed from p_1 and two tokens are added to p_2 (as shown in Figure 3.2. The new marking M_1 becomes $[0, 2]$. In this situation, t_1 is not enabled, but t_2 is enabled.

There are several behavioral properties of a basic PN, such as reachability, boundedness and liveness. Reachability refers to a marking that is reachable from the initial marking [46]. Boundedness emphasizes the upper limit of the number of tokens in each place in a net for any reachable marking from the initial marking [46]. Liveness detects if a PN is deadlock-free [46].

Although basic PNs have been successfully used in many areas, they have some drawbacks, such as

- Lack of an ability to distinguish tokens;
- Lack of compactness in the structure;
- Lack of time control over the firing of transitions.

To overcome these shortcomings, many extensions have been proposed on the basic PNs, boosting the innovation of this tool in the structure and function. He and Murata [27] summarized the evolutions of PNs into four generations.

- Low-level PNs: describing system control;
- High-level PNs: describing both system data and control;
- Hierarchical PNs: abstracting system structures;
- Object-oriented PNs: supporting modern system development approaches.

The first generation, low-level PNs, refer to the basic PNs. High-level PNs are developed to consider the differences in data in one condition. The most important representative of this generation of the PN model is Coloured PNs. In Coloured PNs, the concept of types of tokens is introduced; different types refer to different colors. In hierarchical PNs, the model structures are constructed with a nested pattern. To overcome the weakness of inadequate support of compositionality in the previous generations [41], the idea of object-oriented PNs (OOPNs) come into being with a combination of object-oriented technology and Petri nets. In OOPNs, the concepts such as inheritance, polymorphism and encapsulation are integrated into Petri nets. The details of each object are hidden, and communication to the details of each object is through interfaces, which makes the model reusable and easy to be maintained.

The categories of PN models above are not exhaustive and there are some additional extensions. For example, Timed PNs (TPNs) integrate time delays into net transitions and places to preserve tokens for a period of time before removal. They are a type of deterministic timed nets as the delays are deterministically given. As for the function, they are able to measure the time consuming from one state to another in a net and are suitable to deal with time related problems. Stochastic PNs (SPNs) are another type of PNs with time delays. They have a similar definition and function with TPNs except that the delays are probabilistically specified rather than deterministic [46].

All the extended PNs are designed for specific requirements in practice. For the purpose of the present work, only two extensions (Coloured PNs and Hierarchical PNs) are employed in this thesis, which are sufficient for developing a model to simulate MRI system.

3.1.2 Extended Petri Nets

Coloured Petri Nets

This extension aims to distinguish tokens in a place with different values, as the tokens are painted with various colours. Merging distinguishable tokens into one place, the number of places can be effectively reduced. The formal definition of a Coloured Petri net is as follows.

Definition 3.2 ([35]) *A non-hierarchical* ***Coloured Petri Net*** *(CPN) is a nine-tuple,* $CPN = (P, T, A, \Sigma, V, C, G, E, I)$*, where:*

1. P *is a finite set of places.*
2. T *is a finite set of transitions* T *such that* $P \cap T = \phi$.
3. $A \subseteq P \times T \cup T \times P$ *is a set of directed arcs. An arc can be denoted by two elements: the first one is the starting node and the second one is the ending node, i.e.,* (p, t) *or* (t, p)*,* $p \in P$ *and* $t \in T$.
4. Σ *is a finite set of non-empty colour sets.*
5. V *a finite set of typed variables such that* $Type[v] \in \Sigma$ *for all variables* $v \in V$.
6. $C : P \to \Sigma$ *is a colour set function that assigns a colour set to each place.*
7. $G : T \to EXPR_V$ *is a guard function that assigns a guard to each transition* t *such that* $Type[G(t)] = Bool$.
8. $E : A \to EXPR_V$ *is an arc expression function that assigns an arc expression to each arc* a *such that* $Type[E(a)] = C(p)_{MS}$*, where* p *is the place connected to the arc* a.
9. $I : P \to EXPR_\phi$ *is an initialization function that assigns an initialization expression to each place* p *such that* $Type[I(p)] = C(p)_{MS}$.

In definition, $EXPR$ is the set of expressions described with an inscription language (e.g. CPN ML in CPN Tools) and $EXPR_{V'}$ is defined as the set of expressions $e \in EXPR$ satisfying $Var[e] \subseteq V'$, where $V' \subseteq V$ and $Var[e]$ is the variables in expression e . Therefore, $EXPR_V$ refers to the set of expressions with variables, and $EXPR_\phi$ is the set of all initialization expressions without free variables. *Type* refers to the type of colour set of an object, i.e., $Type[e]$ is the type of an expression $e \in EXPR$ and $Type[v]$ is the type of a variable $v \in V$.

An example of CPNs is displayed in Figure 3.3, which shows how a specific patient moves from one place to another. The data in place p_1 load the information such as the number and name of each patient, and the data in place p_2 are a set of numbers. It can be seen that the colour set of p_1 is recorded with two fields, the first field is integer and the second is text string, while the colour set of p_2 is integer. The expression

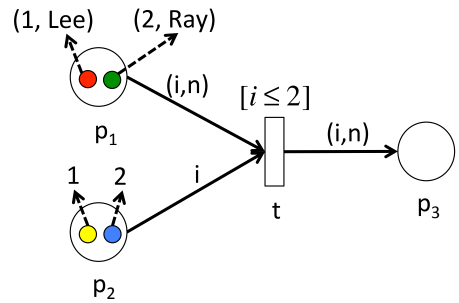

Figure 3.3: An example of CPNs

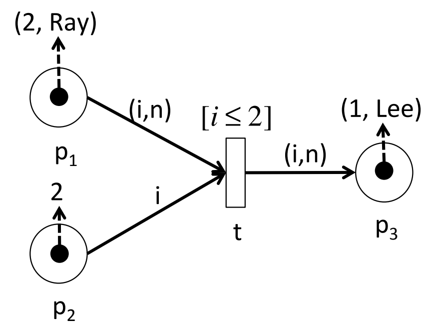

Figure 3.4: The net after the firing of t with $i = 1$

$[i \leq 2]$ on the top of t is a guard stipulating the precondition for t to fire. When a number in p_2 matches to a patient's number in p_1, the patient is allowed to move with the firing of transition t. After firing, the patient's information is removed from p_1 and added to p_3, along with the elimination of the patient's number in p_2. Figures 3.4 and 3.5 exhibit two situations after firing t once. If "i" is substituted by "1", the tokens "(1, Lee)" and "1" are removed from p_1 and p_2 respectively when t fires, then a token "(1, Lee)" is added to p_3 (Figure 3.4). If the substitution is $\{2|i, \text{Ray}|n\}$, the token deposited in p_3 is "(2, Ray)" (Figure 3.5).

When the number of colours is finite, CPNs are actually "folded" nets of the basic PNs in structure and are easily transformed into the corresponding basic PNs. The "unfolded" net of the CPN in Figure 3.3 is illustrated in Figure 3.6. The function of the new net is consistent with the original one, but the number of places and transitions dramatically increases and the net structure becomes complicated.

CPNs concentrate on the integration of diverse resources and provide a concise representation for the intricate nets. However, this improvement is not enough for the description of highly complex systems in that CPNs describe all the details of a system in a single layer, ignoring the hierarchical structure which may present in the target dynamic sysrems. This limitation can be modified by a multi-layer pattern and thus motivates the development of the notion of hierarchy to PNs.

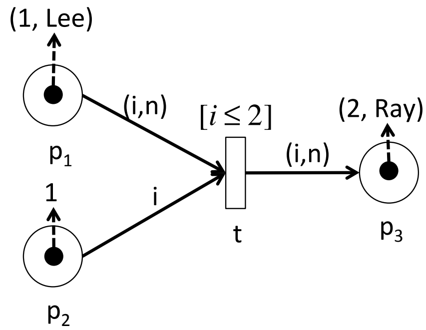

Figure 3.5: The net after the firing of t with $i = 2$

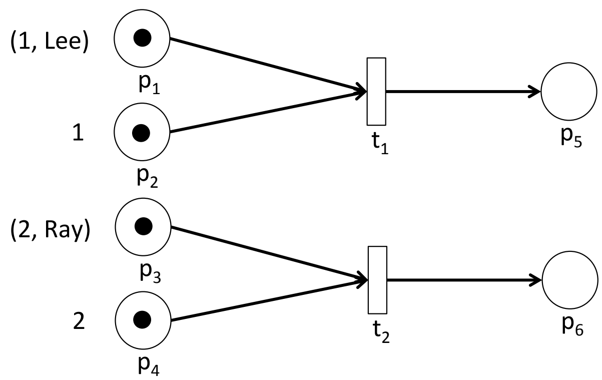

Figure 3.6: The basic PN unfolded from the CPN in Figure 3.3

Hierarchical Coloured Petri Nets

Hierarchical Petri nets encapsulate some system information to the net in a lower layer with the substitution of special transitions. This extension can be added to any type of PNs. In this study, this notion is combined with CPNs, leading to Hierarchical Coloured Petri nets (HCPNs).

Definitions 3.3 to 3.5 are about the definition for HCPNs proposed by Jensen and Kristensen [35]. Definition 3.3 introduces a concept of module, which is a key concept for HCPNs. Definition 3.4 is the formal definition for HCPNs with an extra concept of module hierarchy in Definition 3.5.

Definition 3.3 ([35]) *A **Coloured Petri Net Module** is a four-tuple,* $CPN_M = (CPN, T_{sub}, P_{port}, PT)$*, where:*

1. $CPN = (P, T, A, \Sigma, V, C, G, E, I)$ *is a non-hierarchical Coloured Petri Net.*

2. $T_{sub} \subseteq T$ *is a set of substitution transitions.*

3. $P_{port} \subseteq P$ *is a set of port places.*

4. $PT : P_{port} \rightarrow \{IN, OUT, I/O\}$ *is a port type function that assigns a port type to each port place.*

Substitution transitions are a type of special transitions in a module representing the corresponding sub-modules. The places linking to substitution transitions of a module are called sockets, which also exist in its sub-modules and are called ports. There are three types of ports with respect to their functions: input, output and both of them. For two related socket and port places, they share the same colour set and token allocation.

Definition 3.4 ([35]) *A **Hierarchical Coloured Petri Net** is a four-tuple,* $CPN_H = (S, SM, PS, FS)$ *, where:*

1. *S is a finite set of modules. Each module is a Coloured Petri Net Module*

$$s = \big((P^s, T^s, A^s, \Sigma^s, V^s, C^s, G^s, E^s, I^s), T^s_{sub}, P^s_{port}, PT^s\big) .$$

   It is required that $(P^{s_1} \cup T^{s_1}) \cap (P^{s_2} \cup T^{s_2}) = \phi$ *for all* $s_1, s_2 \in S$ *such that* $s_1 \neq s_2$*.*

2. $SM : T_{sub} \to S$ *is a sub-module function that assigns a sub-module to each substitution transition. It is required that the module hierarchy (see definition 3.5) be acyclic.*

3. *PS is a port-socket relation function that assigns a port-socket relation* $PS(t) \subseteq P_{sock}(t) \times P^{SM(t)}_{port}$ *to each substitution transition t. It is required that* $ST(p) = PT(p')$*,* $C(p) = C(p')$*, and* $I(p)\langle\rangle = I(p')\langle\rangle$ *for all* $(p, p') \in PS(t)$ *and all* $t \in T_{sub}$*.*

4. $FS \subseteq 2^P$ *is a set of non-empty fusion sets such that* $C(p) = C(p')$ *and* $I(p)\langle\rangle = I(p')\langle\rangle$ *for all* $p, p' \in fs$ *and all* $fs \in FS$*.*

In the definition, $ST(p)$ is the type of socket place p and $PT(p')$ is the type of port place p'. $C(p)$ is the colour set of place p and $I(p)\langle\rangle$ is the initial marking expression in place p, i.e., $I(p)\langle\rangle = M_0(p)$. 2^P denotes the set of all subsets of places in a net. A fusion set fs is a set of places in the same or different modules, which are glued together into one compound place. All places in a fusion set have identical colour sets and markings. The set of fusion sets is denoted by FS.

Definition 3.5 ([35]) *The **module hierarchy** for a Hierarchical Coloured Petri Net* $CPN_H = (S, SM, PS, FS)$ *is a directed graph* $MH = (N_{MH}, A_{MH})$*, where:*

1. $N_{MH} = S$ *is the set of nodes.*

2. $A_{MH} = \{(s_1, t, s_2) \in N_{MH} \times T_{sub} \times N_{MH} \,|t \in T^{s_1}_{sub} \wedge s_2 = SM(t)\}$ *is the set of arcs.*

For a substitution transition, it does not make sense to discuss its enabling and firing as it is not a real transition; instead, it is a symbol indicating the behavior of a sub-module. More specifically, when the socket places of a substitution transition are equipped with some tokens, the related port places in a sub-module obtain these tokens as well, which may enable the transitions in the sub-module. Hence, a substitution transition can be treated as a gateway from a module to its sub-module.

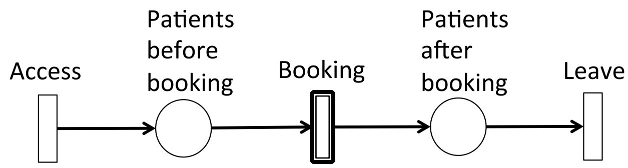

Figure 3.7: An example of HCPNs

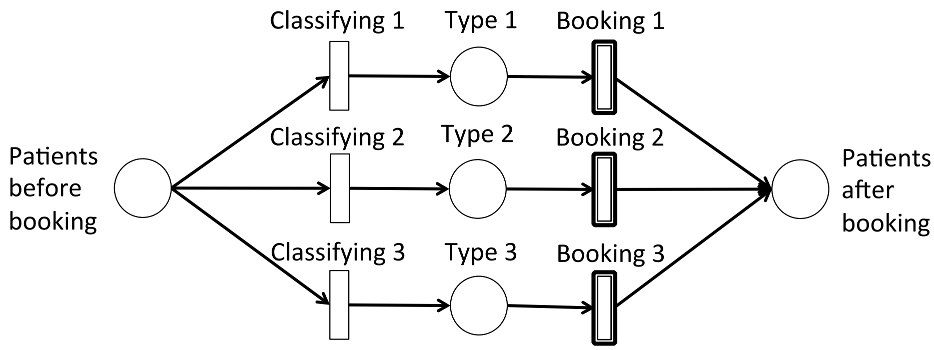

Figure 3.8: The sub-module represented by transition "Booking" in Figure 3.7

The characteristics of HCPNs are demonstrated with an example displayed in Figures 3.7 and 3.8. Figure 3.7 is a module describing a process for patients from access to leave. The transition "Booking" is a substitution transition representing the sub-module in Figure 3.8. The places "Patients before booking" and "Patients after booking" are sockets in Figure 3.7 and are ports in Figure 3.8. If a token with patient information is allocated in the socket place "Patients before booking", it is also sited in the related port place. Based on the particular information, the patient is classified into one type, and then books the service through the booking procedure for a specific type of patients. The transitions "Booking 1", "Booking 2" and "Booking 3" are substitution transitions implying three more sub-modules.

HCPNs are selected to model this system with two concerns. For one thing, the data of medical information are exclusive for patients and thus should be distinguishable in the model. For another, the hierarchical pattern is an optimal choice to logically describe the system from up to bottom by compacting the whole model.

The computer software used in this study to build and run the model is CPN Tools (Version 3.4.0), a free tool to edit, simulate and analyze CPNs. A user-friendly graphical editor is designed for the creation and revision of a net and the checking of syntax is automatically fulfilled. After editing, the simulation of model is carried out rapidly and the performance of model can be analyzed.

3.2 New Extensions to PNs

Though a few defects of the basic PNs have been made up by the foregoing extensions, great efforts are still taken to further improve the approach in other aspects that have not been considered. In this section, two new extensions are proposed, which are useful in further simplification of the model or reduction of the complexity of the model.

3.2.1 Ordered CPNs

An Ordered CPN (OCPN) is an extension based on CPNs, aiming to arrange the tokens of places in a specific order. As a type of CPNs, the tokens in OCPNs are distinguishable as well. The marking of a net is a set of ordered tokens. When a transition fires, tokens in input places are removed with a specified sequence (e.g. FIFO or LIFO) rather than randomly. The definition of OCPNs is stated in Definition 3.6.

Definition 3.6 *An **Ordered Coloured Petri Net** is a two-tuple $OCPN = (CPN, O)$, where:*

1. *$CPN = (P, T, A, \Sigma, V, C, G, E, I)$ is a Coloured Petri Net.*
2. *$O : P \to \Psi$ is an order function that assigns an order to each place $p \in P$, where $\Psi = \{FIFO, LIFO\}$ is an order set, $FIFO$ is "first-in-first-out", and $LIFO$ is "last-in-first-out".*

In OCPNs, tokens are positioned in places as ordered multi-sets, which are defined as follows.

Definition 3.7 *Let $\sigma_1, \sigma_2, \ldots, \sigma_k$ be a group of data with the same type. An **ordered multi-set** (OMS) is a combination of the data in sequence. The number of σ_i is denoted by $n(\sigma_i)$, $n(\sigma_i) \in \{1, 2, \ldots\}$, $i = 1, \ldots, k$. We have*

$$OMS = n(\sigma_1)'\sigma_1 \triangleright n(\sigma_2)'\sigma_2 \triangleright \cdots \triangleright n(\sigma_k)'\sigma_k,$$

where $\triangleright$ is an operator of ordered connection, $'$ is a separator and $n(\sigma_i)'\sigma_i$ is the i'th element of the OMS, $i = 1, \ldots, k$.

For ordered multi-sets, there are some basic operations, such as addition, comparison and subtraction. Let σ_i be a group of ordered data of an OMS, $i = 1, \ldots, k$, (σ_1 is the first one and σ_k is the last), and σ_{k+1} be another one of the same type with σ_i, we have

Addition:

$$OMS \oplus n(\sigma_{k+1})'\sigma_{k+1} = n(\sigma_1)'\sigma_1 \triangleright n(\sigma_2)'\sigma_2 \triangleright \cdots \triangleright n(\sigma_k)'\sigma_k \triangleright n(\sigma_{k+1})'\sigma_{k+1}.$$

Comparison:

$$n(\sigma_{k+1})'\sigma_{k+1} \prec n(\sigma_i)'\sigma_i \Leftrightarrow \sigma_{k+1} = \sigma_i \text{ and } n(\sigma_{k+1}) \leq n(\sigma_i), \quad i = 1, \ldots, k.$$

Subtraction:

If $n(\sigma_{k+1})'\sigma_{k+1} \prec n(\sigma_i)'\sigma_i$, $i = 1, \ldots, k$

$$\begin{aligned} &OMS \ominus n(\sigma_{k+1})'\sigma_{k+1} \\ &= n(\sigma_1)'\sigma_1 \triangleright n(\sigma_2)'\sigma_2 \triangleright \cdots \triangleright n(\sigma_{i-1})'\sigma_{i-1} \triangleright (n(\sigma_i) - n(\sigma_{k+1}))'\sigma_i \triangleright n(\sigma_{i+1})'\sigma_{i+1} \cdots \triangleright n(\sigma_k)'\sigma_k. \end{aligned}$$

The concepts of OCPNs are addressed in Definition 3.8-3.10.

Definition 3.8 *A* ***marking*** *in OCPNs is a function M that maps all $p \in P$ to a typed ordered multi-set of tokens,*

$$M(p) = M_{(1)}(p) \triangleright M_{(2)}(p) \triangleright \cdots \triangleright M_{(n)}(p),$$

where $M_{(1)}(p), \ldots, M_{(n)}(p)$ are the 1th,..., nth tokens in p. We define $M_{hd}(p)$ and $M_{tl}(p)$ as the first and last tokens in p, and $\overline{M_{hd}}(p)$ and $\overline{M_{tl}}(p)$ as the remaining tokens, respectively, i.e.,

$$M_{hd}(p) = M_{(1)}(p), \quad M_{tl}(p) = M_{(n)}(p),$$

$$\overline{M_{hd}}(p) = M_{(2)}(p) \triangleright M_{(3)}(p) \triangleright \cdots \triangleright M_{(n)}(p),$$

$$\overline{M_{tl}}(p) = M_{(1)}(p) \triangleright M_{(2)}(p) \triangleright \cdots \triangleright M_{(n-1)}(p).$$

For each $M_{(i)}(p)$, $i = 1, , \ldots, n$, it consists of two parts: the value of a token and the number of its appearance. If there is no token in p, $M(p) = \phi_{OMS}$. The colour sets of tokens in p are identical, i.e., $Type[M(p)] = C(p)_{OMS}$, where $C(p)_{OMS}$ is the colour set of p, $p \in P$.

Definition 3.9 *The* ***variables of an arc*** *(p, t) are the variables in the expressions of the arc from p to t. The set of all variables for an arc (p, t) is denoted by $Var(p, t) \subseteq V$. The* ***variables of a transition*** *t are the variables marked on the expressions of the input arcs and the guard of t. The set of all variables for a transition t is denoted by $Var(t) \subseteq V$. Let $p_1, \ldots, p_m \in P$ be the input places of transition t, we have*

$$Var(t) = Var(p_1, t) \cup Var(p_2, t) \cup \cdots \cup Var(p_m, t),$$

$$Var(p_i, t) \cap Var(p_j, t) = \phi, \quad i, j = 1, \ldots, m, \quad i \neq j.$$

Definition 3.10 *A* ***binding*** *of a transition t is a function b that maps each variable $v \in Var(t)$ to a value $b(v)$. The set of all bindings for a transition t is denoted by $B(t)$. Let $p_1, \ldots, p_m \in P$ be the input places of transition t, we have*

$$b(Var(t)) = \langle Var(p_1, t) = val(p_1), Var(p_2, t) = val(p_2), \ldots, Var(p_m, t) = val(p_m) \rangle,$$

where $val(p_k)$ is the value of tokens in place p_k matching the variables in arc (p_k, t), $k = 1, \ldots, m$. $val(p_k)$ is closely related to the order of place p_k, e.g., if $O(p_k) = FIFO$, $val(p_k)$ is the token value of $M_{hd}(p_k)$, and if $O(p_k) = LIFO$, $val(p_k)$ is the token value of $M_{tl}(p_k)$.

Definition 3.11 *Let $p_1, \ldots, p_m \in P$ be input places of a transition t. The transition t is* ***enabled*** *with a binding b in a marking M if and only if the following two properties are satisfied:*

1. *$G(t)\langle b \rangle = true$.*
2. *For $p_k \in P$, $k = 1, \ldots, m$,*

$$\text{if } O(p_k) = FIFO, E(p_k, t)\langle b \rangle \prec M_{hd}(p_k),$$

$$\text{if } O(p_k) = LIFO, E(p_k, t)\langle b \rangle \prec M_{tl}(p_k).$$

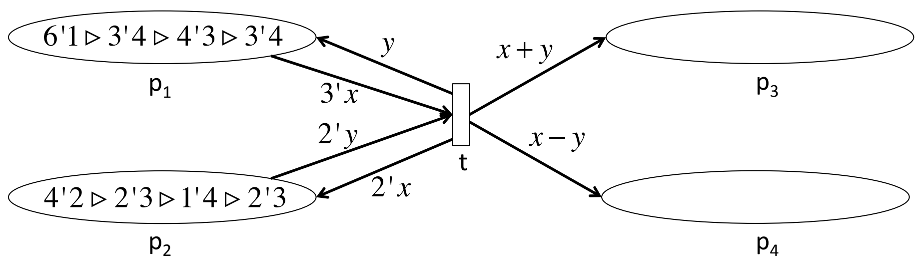

Figure 3.9: An example of OCPNs

where G is a function mapping a transition to the expression of its guard and $G(t)\langle b\rangle$ is a guard function with a binding b, the type of which is Boolean. E is a function mapping an arc to its expression and $E(p_k, t)\langle b\rangle$ is an arc expression function with a binding b.

If an enabled transition t **fires**, a new marking M' is obtained on the basis of the following rules. For a place p connected to t, $p \in P$,

$$\text{If } O(p) = FIFO, \quad M'(p) = ((M_{hd}(p) \ominus E(p,t)\langle b\rangle) \rhd \overline{M_{hd}}(p)) \oplus E(t,p)\langle b\rangle;$$

$$\text{If } O(p) = LIFO, \quad M'(p) = (\overline{M_{tl}}(p) \rhd (M_{tl}(p) \ominus E(p,t)\langle b\rangle)) \oplus E(t,p)\langle b\rangle.$$

The firing rules above are illustrated with an instance in Figure 3.9. In this net, the colour set of p_1 is integer, the same with p_2, p_3 and p_4. The current marking of the net is

$$\begin{aligned} M &= [M(p_1), M(p_2), M(p_3), M(p_4)] \\ &= [6'1 \rhd 3'4 \rhd 4'3 \rhd 3'4, 4'2 \rhd 2'3 \rhd 1'4 \rhd 2'3, \phi_{OMS}, \phi_{OMS}]\,. \end{aligned}$$

For transition t, the input places are p_1 and p_2, and the output places are p_1, p_2, p_3 and p_4.

If $O(p_i) = FIFO$, $i = 1, \ldots, 4$ the binding of t is $b = \langle x = 1, y = 2\rangle$ and we have

$$M_{hd}(p_1) = 6'1, \quad M_{hd}(p_2) = 4'2,$$

$$E(p_1,t)\langle b\rangle = 3'1, \quad E(p_2,t)\langle b\rangle = 2'2, \quad E(p_3,t)\langle b\rangle = \phi_{OMS}, \quad E(p_4,t)\langle b\rangle = \phi_{OMS},$$

$$E(t,p_1)\langle b\rangle = 1'2, \quad E(t,p_2)\langle b\rangle = 2'1, \quad E(t,p_3)\langle b\rangle = 1'3, \quad E(t,p_4)\langle b\rangle = 1'(-1).$$

There is no guard marked on t, which means no restriction is added to the firing of t, i.e., $G(t)\langle b\rangle = true$. Since $E(p_1,t)\langle b\rangle \prec M_{hd}(p_1)$ and $E(p_2,t)\langle b\rangle \prec M_{hd}(p_2)$, t is enabled. When t fires, a new marking M' is generated and we have

$$\begin{aligned} M'(p_1) &= ((M_{hd}(p_1) \ominus E(p_1,t)\langle b\rangle) \rhd \overline{M_{hd}}(p_1)) \oplus E(t,p_1)\langle b\rangle \\ &= 3'1 \rhd 3'4 \rhd 4'3 \rhd 3'4 \rhd 1'2, \end{aligned}$$

$$\begin{aligned} M'(p_2) &= ((M_{hd}(p_2) \ominus E(p_2,t)\langle b\rangle) \rhd \overline{M_{hd}}(p_2)) \oplus E(t,p_2)\langle b\rangle \\ &= 2'2 \rhd 2'3 \rhd 1'4 \rhd 2'3 \rhd 2'1, \end{aligned}$$

$$M'(p_3) = ((M_{hd}(p_3) \ominus E(p_3,t)\langle b\rangle) \rhd \overline{M_{hd}}(p_3)) \oplus E(t,p_3)\langle b\rangle$$
$$= 1'3,$$

$$M'(p_4) = ((M_{hd}(p_4) \ominus E(p_4,t)\langle b\rangle) \rhd \overline{M_{hd}}(p_4)) \oplus E(t,p_4)\langle b\rangle$$
$$= 1'(-1).$$

Thus,

$$M' = [M'(p_1), M'(p_2), M'(p_3), M'(p_4)]$$
$$= [3'1 \rhd 3'4 \rhd 4'3 \rhd 3'4 \rhd 1'2, 2'2 \rhd 2'3 \rhd 1'4 \rhd 2'3 \rhd 2'1, 1'3, 1'(-1)]\,.$$

If $O(p_i) = LIFO$, $i = 1, \ldots, 4$, the binding of t is $b = \langle x = 4, y = 3\rangle$ and we have

$$M_{tl}(p_1) = 3'4, \quad M_{tl}(p_2) = 2'3,$$

$$E(p_1,t)\langle b\rangle = 3'4, \quad E(p_2,t)\langle b\rangle = 2'3, \quad E(p_3,t)\langle b\rangle = \phi_{OMS}, \quad E(p_4,t)\langle b\rangle = \phi_{OMS},$$

$$E(t,p_1)\langle b\rangle = 1'3, \quad E(t,p_2)\langle b\rangle = 2'4, \quad E(t,p_3)\langle b\rangle = 1'7, \quad E(t,p_4)\langle b\rangle = 1'1.$$

Since $E(p_1,t)\langle b\rangle \prec M_{tl}(p_1)$ and $E(p_2,t)\langle b\rangle \prec M_{tl}(p_2)$, t is enabled. When t fires,

$$M'(p_1) = (\overline{M_{tl}}(p_1) \rhd (M_{tl}(p_1) \ominus E(p_1,t)\langle b\rangle)) \oplus E(t,p_1)\langle b\rangle$$
$$= 6'1 \rhd 3'4 \rhd 4'3 \rhd 1'3,$$

$$M'(p_2) = (\overline{M_{tl}}(p_2) \rhd (M_{tl}(p_2) \ominus E(p_2,t)\langle b\rangle)) \oplus E(t,p_2)\langle b\rangle$$
$$= 4'2 \rhd 2'3 \rhd 1'4 \rhd 2'4,$$

$$M'(p_3) = (\overline{M_{tl}}(p_3) \rhd (M_{tl}(p_3) \ominus E(p_3,t)\langle b\rangle)) \oplus E(t,p_3)\langle b\rangle$$
$$= 1'7,$$

$$M'(p_4) = (\overline{M_{tl}}(p_4) \rhd (M_{tl}(p_4) \ominus E(p_4,t)\langle b\rangle)) \oplus E(t,p_4)\langle b\rangle$$
$$= 1'1.$$

Thus,

$$M' = [6'1 \rhd 3'4 \rhd 4'3 \rhd 1'3, 4'2 \rhd 2'3 \rhd 1'4 \rhd 2'4, 1'7, 1'1]\,.$$

This extension is quite useful in the system concerned by the present study. In the booking stage of MRI service, the cases are categorized and put in a waiting list with an order, and then they are scheduled with the first-come-first-serve principle for each type, i.e., for the cases with similar medical records, the early one requesting for MRI is probably assigned an earlier time slot. The queuing of patients is straightforward using OCPNs as shown in Figure 3.10, which describes the process that patients move from "PBB (Patients before booking)" to "PAB (Patients after booking)" via the occurrence of an event "Booking". In the net, IN is a function that assigns a set of medical information to each patient pa_i, $i = 1, \ldots, n$, and da is an arc expression. Let $O(PBB) = O(PAB) = FIFO$, then patients can be booked in sequence.

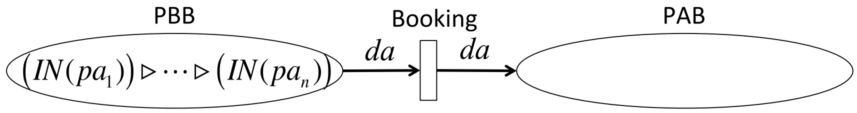

Figure 3.10: An application instance of OCPNs

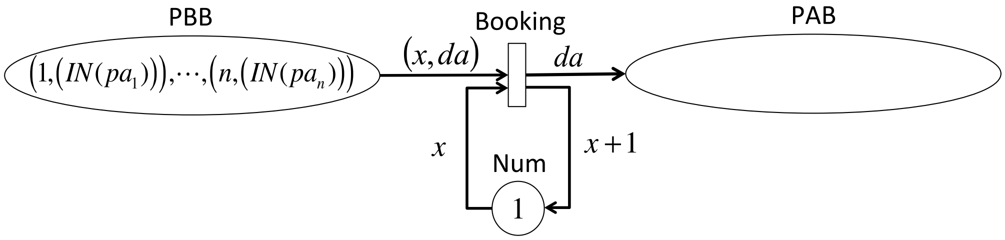

Figure 3.11: A CPN model describing the process in Figure 3.10 with a control place

The process in Figure 3.10 can be modeled with CPNs as well, which are displayed in Figure 3.11 and 3.12. In Figure 3.11, tokens are disorderly in the place "PBB". To make sure that early cases are dealt with first, cases in that place are numbered. The sequence for patients to be booked is controlled by another place "Num" (a control place) with the token of an increasing number to transfer patients orderly. However, this way brings two problems: (1) a control place is always attached when the first-come-first-serve principle is obeyed for a place, (2) the cases should be renumbered each time when they are needed to queue up. Accordingly, more nodes are required in the modeling. Figure 3.12 shows another CPN model to describe the process in Figure 3.10. In this example, a concept "list" is involved. A list is an array containing data of varying types and sizes. The token in "PBB" is a list with the components recording the information of patients. The token in "PAB" is an empty list. "p", "ps" and "$ps0$" are all variables with the type of list. "$p :: ps$" is a function setting component "p" as the head of the list "ps". "$ins\ ps0\ p$" is a function inserting component "p" at the end of the list "$ps0$". When the transition "Booking" fires, the components in "PBB" are moved to "PAB" in sequence. That is, patients can be transferred orderly. However, more arcs are required in this model. In summary, using CPNs to model the process results in an increase of the number of nodes or arcs, which are redundant and increase the complexity of the model structure.

3.2.2 Prioritized HCPNs

A Prioritized HCPN (PHCPN) is an extension of HCPNs with a modification on the substitution transitions by introduction of the notion of global priorities. The priority of a substitution transition can be passed to its related sub-module, which means that a transition cannot fire until all other transitions with higher priorities are un-enabled. This extension is defined as follows.

Definition 3.12 *A* ***Prioritized Coloured Petri Net Module*** *is a five-tuple,* $PCPN_M = (CPN, T_{sub}, T_{pri}, P_{port}, PT)$*, where:*

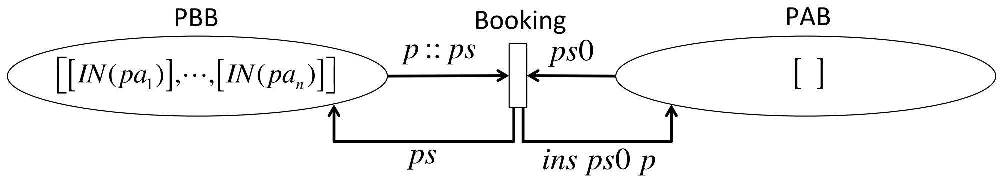

Figure 3.12: A CPN model describing the process in Figure 3.10 with a list

1. *$CPN = (P, T, A, \Sigma, V, C, G, E, I)$ is a non-hierarchical Coloured Petri Net.*

2. *$T_{sub} \subseteq T$ is a set of substitution transitions.*

3. *$T_{pri} \subseteq T_{sub}$ is a set of prioritized substitution transitions.*

4. *$P_{port} \subseteq P$ is a set of port places.*

5. *$PT : P_{port} \rightarrow \{IN, OUT, I/O\}$ is a port type function that assigns a port type to each port place.*

Definition 3.13 *A **Prioritized Hierarchical Coloured Petri Net** is a five-tuple, $PHCPN = (MO, SM, PS, FS, PR)$, where:*

1. *MO is a finite set of modules. Each module is a Prioritized Coloured Petri Net Module $mo = \big((P^{mo}, T^{mo}, A^{mo}, \Sigma^{mo}, V^{mo}, C^{mo}, G^{mo}, E^{mo}, I^{mo}), T^{mo}_{sub}, T^{mo}_{pri}, P^{mo}_{port}, PT^{mo}\big)$. It is required that $(P^{mo_1} \cup T^{mo_1}) \cap (P^{mo_2} \cup T^{mo_2}) = \phi$ for all $mo_1, mo_2 \in MO$ such that $mo_1 \neq mo_2$.*

2. *$SM : T_{sub} \rightarrow MO$ is a sub-module function that assigns a sub-module to each substitution transition. It is required that the module hierarchy is acyclic. In contrast, $SM^{-1} : MO \rightarrow T_{sub}$ is the inverse function of SM mapping a sub-module to the corresponding substitution transition.*

3. *PS is a port-socket relation function that assigns a port-socket relation $PS(t) \subseteq P_{sock}(t) \times P^{SM(t)}_{port}$ to each substitution transition t. It is required that $ST(p) = PT(p')$, $C(p) = C(p')$, and $I(p)\langle\rangle = I(p')\langle\rangle$ for all $(p, p') \in PS(t)$ and all $t \in T_{sub}$.*

4. *$FS \subseteq 2^P$ is a set of non-empty fusion sets such that $C(p) = C(p')$ and $I(p)\langle\rangle = I(p')\langle\rangle$ for all $p, p' \in fs$ and all $fs \in FS$.*

5. *$PR : T \rightarrow \{1, 2, \ldots\}$ is a priority function that assigns a priority to each transition t. It is required that $PR(t) = 1$ for all $t \in T^{mo} \backslash T^{mo}_{pri}$ and all $mo \in MO$ such that $SM^{-1}(mo) \notin T_{pri}$. If $SM^{-1}(mo) \in T_{pri}$, $PR(t) = PR(SM^{-1}(mo))$ for all $t \in T^{mo} \backslash T^{mo}_{pri}$ and $mo \in MO$. In addition, $PR(t') = PR(t) + 1$ for all $t \in T^{mo} \backslash T^{mo}_{pri}$, $t' \in T^{mo}_{pri}$ and $mo \in MO$.*

Similar to substitution transitions, it is also meaningless to discuss the enabling and firing of prioritized substitution transitions. For a transition $t \in T \backslash T_{sub}$, it is enabled when the conditions on its guard and input arc expressions are satisfied. If a transition $t \in T \backslash T_{sub}$ is enabled, it fires when $PR(t') \leq PR(t)$ for any enabled transition $t' \in T \backslash T_{sub}$ such that $t' \neq t$.

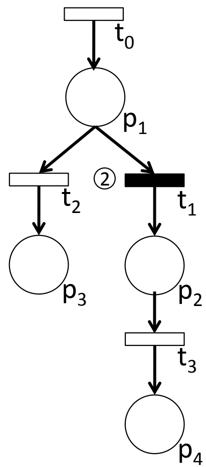

Figure 3.13: An example of PHCPNs

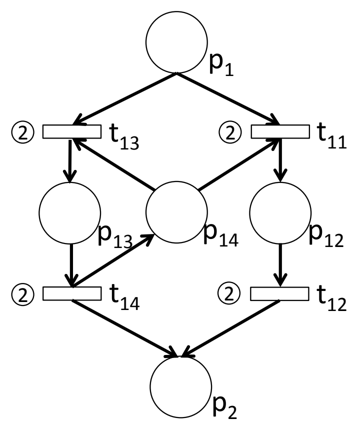

Figure 3.14: The sub-module represented by transition "t_1" in Figure 3.13

Figures 3.13 and 3.14 show an example of PHCPNs containing a module and a sub-module. In Figure 3.13, t_1 is a prioritized substitution transition with a value of priority labeled on the left. For a unit priority, the label is usually omitted. The sub-module related to t_1 is exhibited in Figure 3.14. Since t_1 is a prioritized substitution transition, the priorities of all transitions in the sub-module are equal to the priority of t_1. According to the firing rule, transition t_0, t_2 and t_3 can fire only if they are enabled and all transitions in Figure 3.14 are not enabled. For the transitions with the same priority, they fire randomly.

This extension can effectively avoid some conflicts in the modeling of this work, which is demonstrated with the following instance. Figures 3.15 and 3.16 describe a simple process for patients to be assigned time slots. As it is displayed in Figure 3.15, time slots are divided into two parts assigning to two groups of patients separately. The cases acquiring time slots are moved to the place "Patients with TS". If both parts of time slots ("TS_1" and "TS_2") are free of tokens, the remaining cases in place "Patients without TS" are moved to place "Patients in waiting list" and continue to wait. The dashed line from place "TS_1" to transition "Quitting" ending with a small circle is an inhibitor arc with a function to enable the transition when the

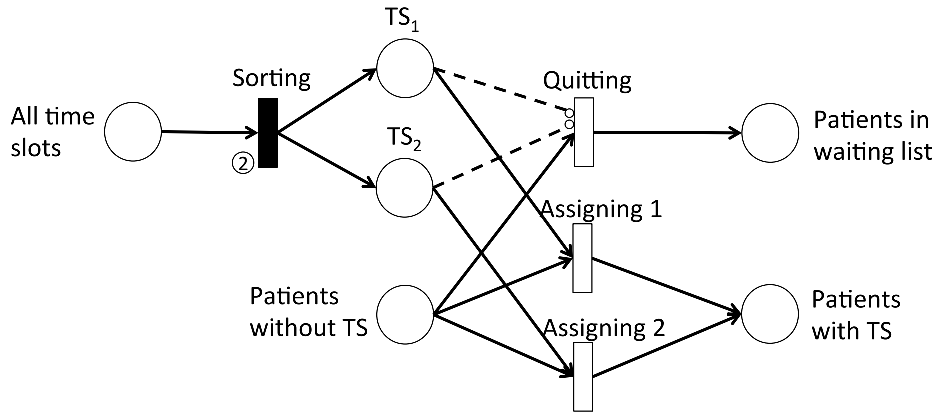

Figure 3.15: An application instance of PHCPNs

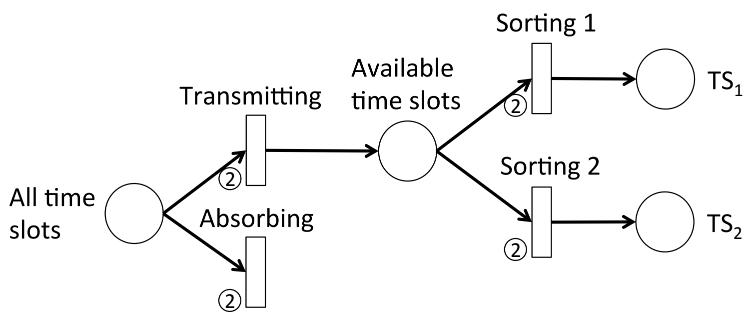

Figure 3.16: The sub-module represented by transition "Sorting" in Figure 3.13

input place has no token. The transition "Sorting" is a prioritized substitution transition representing a sub-module of time slot sifting and sorting in Figure 3.16. With the priority setting to transition "Sorting", time slots can be sorted thoroughly before being dispensed, and patients cannot be assigned only if no more time slots are available in the system, so that the purpose of time slot assignment is well achieved.

However, if no priority is considered in the modeling, i.e., using HCPNs, the system may be in stuck in some situation. Without priority, all enabled transitions in the net can fire randomly, which means that the sorted time slots in "TS_1" and "TS_2" can be assigned to patients as the sorting process is in progress. Thus, it is possible that both "TS_1" and "TS_2" are vacant when their tokens are dispensed and new tokens are not obtained. Then, the backlog of demands is moved to place "Patients in waiting list". Though new tokens are added to "TS_1" and "TS_2" when the sorting of time slots is completed, they cannot be consumed any more. To solve the problem, more nodes and arcs are required to control the behavior of system. For example, a transition "Detecting" can be added in the sub-module of "Sorting" to make sure that sorted time slots are not moved to "TS_1" and "TS_2" until there are no tokens in both place "All time slots" and "Available time slots" (see Figure 3.17). Then, the sorted time slots can be assigned to patients successfully. However, with the employment of additional nodes and arcs, the size of model is enlarged.

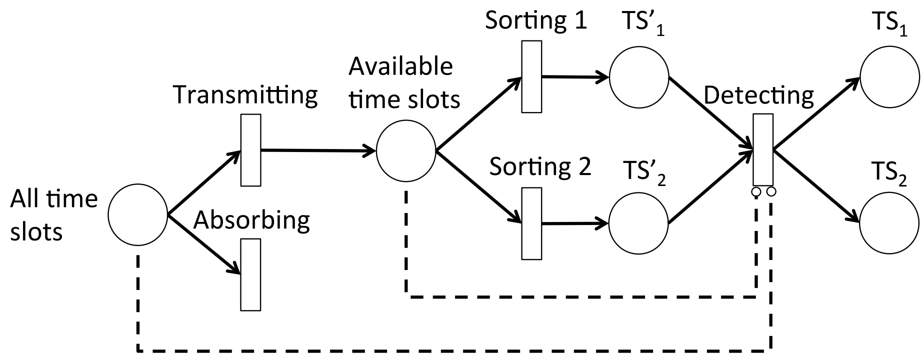

Figure 3.17: The sub-module constructed by HCPNs

3.3 Conclusion

This chapter introduced Petri nets (PNs) and their extensions, focusing on the Coloured Petri nets (CPNs) and Hierarchical Coloured Petri nets (HCPNs). On the basis of these two extensions, two new extensions (OCPNs and PHCPNs) were proposed to improve the expressing power of the PNs and the effectiveness of the operations. OCPNs were designed to describe the ordered data in a direct way cutting a few trivial components. PHCPNs impose restrictions on the firing sequence of transitions via priorities to avoid possible conflicts in event transitions. Both extensions seem to effectively refine the structure of a PN model. However, as they are not supported by the current software of PNs, the implementation of them was made in this research still with HCPNs yet by defining additional nodes and arcs in the HCPN.

Chapter 4

Model Development and Model Validation

This chapter focuses on the modeling of patient flow in the whole process of MRI examination with HCPNs and is organized as follows. Section 4.1 revisits the process of MRI system and summarizes the semantics of the target system. The PN models capturing a part of the semantics are developed in Section 4.2. Section 4.3 validates the model based on real data. A summary is given in Section 4.4.

4.1 MRI Booking and Serving Process: Revisit

MRI booking process is a period from patient request for MRI examination to patient being scheduled. If a general practitioner or specialist thinks a patient should undergo an MRI examination, he transmits a request to a booking center with the patient's basic information and the referral. Then, a radiologist may confirm the information of the patient and determine a priority level. After that, the patient may be assigned a time slot to do the scan (including scan date, start time and anticipated end time) by booking staff.

MRI serving process is a period from patient arrival for the examination to patient departure. When a patient arrives at the scan site on the appointed day, he needs to register first and do some preparations (e.g. fill a screening form, change clothes, intravenous injection, etc.). Then, the patient is asked to stay in the waiting room until the machine is available. The service time (the time period that a patient stays in the scan room) may be shorter or longer than anticipated. After the examination, the patient is free to go home.

The above procedures are experienced by an outpatient. For the inpatients, they do not need to book the service and make all the preparations as outpatients do. They can receive the service directly after their doctor calls for it. A segment of time is always reserved for them each day.

Based on the whole process of MRI examination, several pieces of semantics are summarized as follows, which are expected to be captured by the model.

Semantics 1: Both booking and serving processes of outpatients are modeled. In the booking process, an order for MRI should be confirmed before being passed to the booking staff for scheduling an appointment. This confirmation step can be ignored in the model, as it is in practice very short and not a crucial step for the system. As such, an order is directly put into the waiting list for booking as soon as a request has been received. In the booking stage, patients are booked by category (future cases and level 3, level 4 and level 2 routine cases) and assigned a time slot of either 30 or 60 minutes within the department hours. After booking, it is possible for some patients to cancel the examination and leave the system earlier than the appointed time.

In the serving process, it is assumed that patients are just ready for scan at the appointed start time and the scan lasts as long as anticipated. That is, at the moment that a patient finishes the scan, the next patient is just ready. Thus, there is no room empty time/machine idle time (i.e. the time slot from the end time of a patient to the start time of the next patient) and service overtime (i.e. the time slot from the anticipated end time to the real end time of a patient). Each patient only scans once. The machines always work well without sudden breakdown or regular maintenance and enough medical staff, including nurses, doctors, technicians and radiologists, are equipped in MRI department to guarantee that no delay is caused by the scarcity of human resources.

Semantics 2: Based on the assumptions of Semantics 1, it is further assumed that patients are allowed to choose their preferred scan site when more than one scan site is available.

Semantics 3: Based on the assumptions of Semantics 1, it is further assumed that patients are allowed to choose their preferred time slot.

Semantics 4: In Semantics 1, it is assumed that for each patient, the scan lasts as long as anticipated. This assumption is changed to be that the scan may last shorter or longer than anticipated. If the scan ends earlier than anticipated, there is a period of room empty time. If the scan ends later than anticipated, there is no room empty time; instead, there is a period of overtime in this case. Given that the department office hours in a day are limited, accumulated overtimes from several patients may eventually impede the scan of the following patients and lead to a re-ordering of the services for them.

Semantics 5: In Semantics 1, only outpatients are considered. The inpatient is a factor to influence the process of MRI system. Though inpatients do not need to book the service, they are in the serving process and may influence the service of outpatients. Furthermore, it is possible to arrange inpatients to the room empty time collected in Semantics 4 as they have been in the hospital and can be provided the service at any time. Therefore, the room empty time may be reused and more patients may be scanned.

Semantics 6: The body part to be scanned is a factor to influence booking process. Patients with different scan parts may require different service times and can be assigned different time slots. To make the assignment of the time slots more flexible and precise, the concept of time slot unit can be introduced, which is a small fraction of time (e.g. 5 minutes). A patient is assigned a specific number of continuous time slot units according to his unique feature of medical information. The subdivision of time slots may effectively compact the schedule and diminish room empty time.

The models for Semantics 1 to Semantics 4 are constructed in the next section. Semantics 5 and Semantics 6 are not modeled in the present study and are considered as future work.

Table 4.1: Assumptions of the model for Semantics 1

	Assumptions
1	Only outpatients are considered
2	Two MRI machines with the same efficiency are modeled
3	There is a limit of department hours
4	Patients in the same category are assigned the same service time
5	There are no room empty time and service overtime
6	Each Patient only scans once
7	MRI machines always work well
8	Enough medical staff are provided

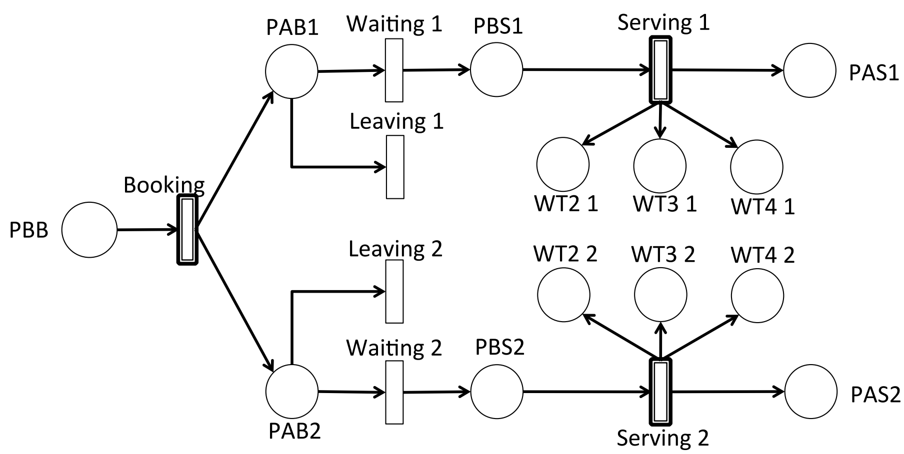

Figure 4.1: The PN model of the overall process

4.2 PN models

This section introduces the corresponding PN models. It is noted that the models exhibited in this section are all conceptual nets to explain the ideas of modeling. The real models built and simulated with the computer software are put out in Appendix A.

4.2.1 Model for Semantics 1

A PN model is developed to describe the whole process of MRI examination. The main assumptions of this model are listed in Table 4.1. To make the system logically clear, the nets of the model are organized in a hierarchical structure. Figure 4.1 shows the net of the overall process. In the model, two MRI machines (MRI_1 and MRI_2) are involved and they are assumed to have the same efficiency. The outpatients to be arranged to either of the machines are initially in the same waiting list, i.e., in the place “PBB” (patients before booking). Once they are booked, they are separated into two teams, i.e., “PAB1” (patients after booking to MRI_1) and "PAB2" (patients after booking to MRI_2), and wait for the serving process of a specific machine. For MRI_1, the place "PBS1" represents patients before serving and the transition "Serving 1" represents the serving process. After serving, the patients served by MRI_1 are moved to "PAS1" (patients after serving) and the waiting times of patients are output in "WT2 1", "WT3 1" and "WT4 1" based on the priority levels. Some patients may cancel the service and leave the system after booking, which make the transition "Leaving 1" fire. For MRI_2, the process after booking is similar to that of MRI_1.

The model in Figure 4.1 is a module with the highest hierarchy. Three substitution transitions ("Booking", "Serving 1", "Serving 2") can be found in the figure indicating that the module contains three sub-modules, which represent two processes: booking process and serving process. They are introduced in the following in details.

Booking Process

To model the booking process, the patients in the waiting list are divided into two types: future cases and routine cases. Future cases refer to the patients who are suggested to be served no earlier than a specified date for various reasons. The waiting list of future cases is organized according to the desired earliest date. To meet the special demands, future cases are booked with a priority, i.e. ahead of those cases without any specific date restriction.

Routine cases are further categorized in priority levels assigned by a radiologist, which ranks patients from level 1 to level 4 based on the urgency of examination. It is noted that the classification of four priority levels is particularly used in Saskatoon Health Region. Patients belonging to level 1 have the highest priority since they suffer from life-threatening diseases and should be examined within 24 hours. For patients in category 2, immediate treatments are not necessary, but prompt diagnoses are still required (with a target of 7 days) to provide additional information for surgical or medical management. In level 3, patients manifest symptoms of potentially life-threatening diseases and can bear a period of waiting (with a target of 30 days) with no or less negative effect on the treatment outcomes. Level 4 includes patients who undergo slight or chronic disease with a slow aggravation of the condition and who just take disease prevention, and also those for whom limited therapeutic options are available. The target delay for patients in this category to have a booking is less than 90 days [1]. Given that level 1 cases are rare and always handled in time in reality, it is not necessary to model them and discuss their waiting times. Thus, only level 2, 3 and 4 cases are considered in this model. All the cases are booked as early as possible.

Figure 4.2 shows a conceptual model for the booking process, which expands from the transition "Booking" in Figure 4.1. It can be seen from Figure 4.2 that patients in "PBB" are classified into future cases (FC) and routine cases (RC), and the routine case category is further divided into level 3 (L3), level 4 (L4) and level 2 (L2). The tokens in "PBB" are a collection of records that identify patients with the pattern $(f, (c, e), p, d_g, d_f)$. The f takes a 0-1 number denoting whether a patient is a future case ($f = 1$) or a

[1] http://www.health.gov.sk.ca/diagnostic-imaging-mri-prioritization

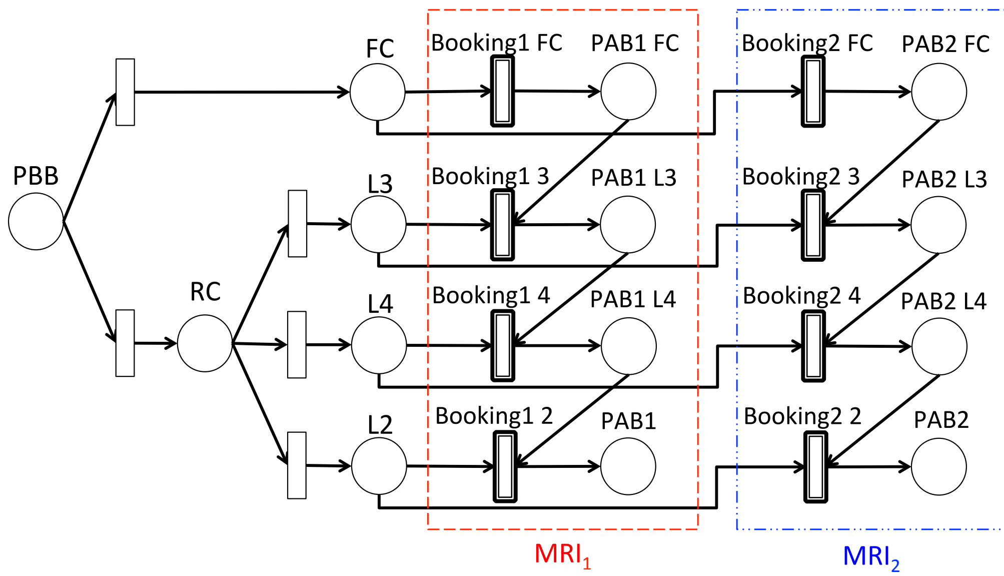

Figure 4.2: The PN model of booking process

routine case ($f = 0$). The (c, e) is a record with two elements, where c and e take both 0-1 numbers denoting whether the contrast is required ($c = 1$ if the contrast is required and $c = 0$ if not) and whether the patient's examination needs to be reviewed by radiologists ($e = 1$ if the review is needed and $e = 0$ if not), respectively. The (c, e) are another two factors to classify patients, and they will be further discussed later. The p is the priority level represented by an integer between 2 and 4. The d_g is the day when a case is generated and d_f is the earliest day that a future case is suggested to be examined. For the routine cases, $d_f = 0$.

Two machines considered in the model are scheduled separately. As shown in Figure 4.2, the part in the left rectangle is the scheduling of MRI_1 and that in the right rectangle is about MRI_2. The patients are arranged to the machine with earlier available time slots ensuring that both machines are equally busy. Since the scheduling of MRI_1 and scheduling of MRI_2 are similar, only one machine (MRI_1) is illustrated in the following. For MRI_1, the booking processes of the different categories of cases are performed with an order. The group being booked first is future cases (the patients who desire to be served no earlier than a specified date), followed by level 3 and 4 cases, and finally is level 2. Future cases are booked first because their desired earliest scan dates should be met. For level 3 and level 4 routine cases, there are usually backlog requests, which can be arranged to the schedule directly. As level 2 cases are urgent and should receive services quickly, several time slots are reserved daily to satisfy the potential requirements. Thus, level 2 cases are booked at last to the available time slots reserved for them. The completed cases of a group are considered in the booking process of the next group, so that the booking results of each group can be combined in a chronological order in place "PAB1" to wait for serving.

Let us take booking for the patients in the future case category as an example. Upon receiving the cases, they are categorized based on (1) whether the patient needs contrast and (2) whether the patient's

Table 4.2: The number of time slots for each situation on weekdays and weekends

	Weekdays	Weekends
Situation 1	5	0
Situation 2	3	3
Situation 3	5	4

examination needs to be reviewed by radiologists. This will lead to four categories: (1) with contrast and with review (WcWr), (2) without contrast and with review (WOcWr), (3) with contrast and without review (WcWOr), and (4) without contrast and without review (WOcWOr). It is noted that the serving hours of MRI department are from 8 a.m. to 11:30 p.m. on weekdays and from 8 a.m. to 5 p.m. on weekends. However, the time period with the provision of contrast is limited, and further the radiologists are not available during all these periods. Thus, the patients with different conditions are arranged to specific time periods. For patients requiring contrast, they are booked between 8 a.m. and 8 p.m. on weekdays and between 8 a.m. and 11 a.m. on weekends, while those without this need are usually booked in the remaining time slots. The patients who need to be reviewed are booked between 8 a.m. and 5 p.m. on weekdays, and those without this need can be booked in the other time. The time period from 11 a.m. to 3 p.m., which is usually reserved for inpatients, should be noted, as this period is not open to outpatients. The PN model does not include this period of time, as this study does not consider inpatients.

Based on the aforementioned discussion, daily hours are split into three segments on weekdays: 8 a.m. to 11 a.m. and 3 p.m. to 5 p.m. for patients in both WcWr and WOcWr (situation 1), 5 p.m. to 8 p.m. for patients in WcWOr (situation 2) and 8 p.m. to 11:30 p.m. for patients in WOcWOr (situation 3), and two segments on weekends: 8 a.m. to 11 a.m. for situation 2, and 3 p.m. to 5 p.m. for situation 3. The patients belonging to situation 1 are not scheduled to weekends as no radiologist is on duty at that time. As a preparation of booking, the hours on weekdays are evenly divided into several time slots in advance: 60 minutes per slot from 8 a.m. to 8:15 p.m. with a 15 minutes break and 30 minutes per slot from 8:45 p.m. to 11:30 p.m. with a 15 minutes break. It is noted that the foregoing time slot division refers to the MRI booking center in Saskatoon. On weekends, the division changes to be 60 minutes per slot from 8 a.m. to 3 p.m. and 30 minutes per slot from 3 p.m. to 5 p.m. The number of available time slots for outpatients in each situation is distributed as shown in Table 4.2.

The PN model of the booking of future cases (represented by the transition "Booking1 FC" in Figure 4.2) is displayed in Figure 4.3. In the net, a time slot is represented by a record (d, n) with two elements: d is the day and n is the time slot number in that day. Available time slots are sorted in "TS1 FC" with the destinations of places "ts1" (time slots for situation 1), "ts2" (time slots for situation 2) and "ts3" (time slots for situation 3). The future cases in "FC" are classified into three situations and assigned with the corresponding time slots. For each future case, there is a comparison between the desired earliest date and the dates of available time slots. An available time slot can be assigned to a future case only if the date of

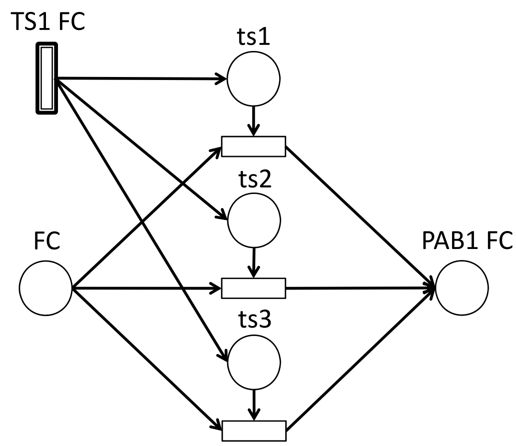

Figure 4.3: The PN model of the booking of future cases

the time slot is equal to or later than the desired earliest date. After booking, the patients are moved to "PAB1 FC" and the occupied time slots are removed from the time slot containers ("ts1", "ts2" and "ts3"). With the additional information of an assigned time slot for a particular patient, the pattern of tokens in "PAB1 FC" becomes $(f, (c, e), p, d_g, d_f, d, n)$.

The booking process of level 3 cases is shown in Figure 4.4. The model structure is similar to that of future cases with the exception of an involvement of preceding booked cases. The cases in "PAB1 FC" participate in the sifting and sorting of time slots and then transfer entirely to "PAB1 FC′", which are combined with the booked level 3 cases in "PAB1 3" before being sent to "PAB1 L3". For level 3 cases, the rule of time slot assignment is slightly different from that of future cases. A comparison is made between the request date and the dates of available time slots. The premise of time slot assignment is that the date of an available time slot is equal to or later than the request date. For each situation (situation 1, 2 and 3), cases are booked with the principle of first-come-first-serve. The booking of level 4 and level 2 cases are similar to that of level 3 and thus not repeated here.

The sub-modules of time slot sorting represented by "TS1 FC" in Figure 4.3 and "TS1 3" in Figure 4.4 are exhibited in Figure 4.5 and 4.6, respectively. The sorting process in Figure 4.5 is simply from the available time slots to the sorted ones for each situation. As future cases have the highest priority in booking, all time slots are optional for them. Comparatively, the process in Figure 4.6 is more complex with an additional step of time slot sifting. The time slots are filtered first to remove those occupied by future cases. For example, if a time slot has been assigned to a future case, the transition "Sifting N" is enabled and fires to delete the time slot in "All TS" and concurrently move the case to "PAB1 FC′", while if a time slot is unoccupied, transition "Sifting Y" can fire with the consequence to send the time slot to "Available TS" and return the case to "PAB1 FC" for the next sifting. The filtered time slots in "Available TS" can proceed with sorting as that of Figure 4.5.

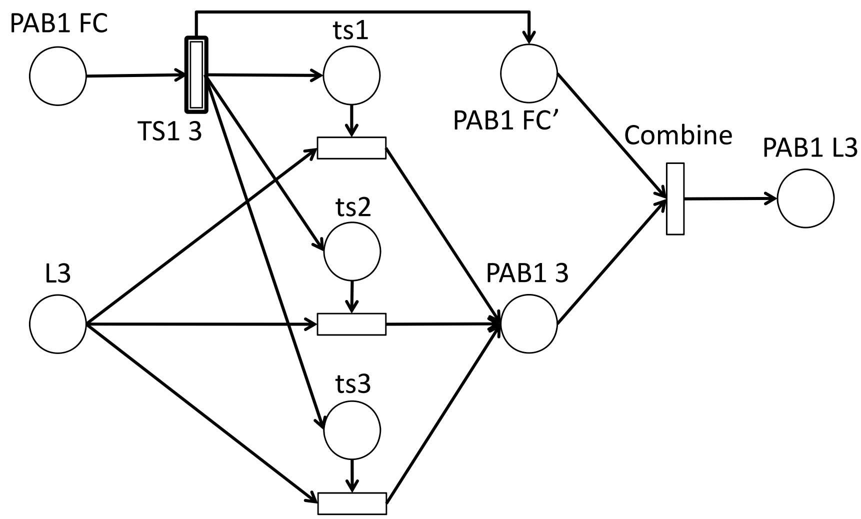

Figure 4.4: The PN model of the booking of level 3 cases

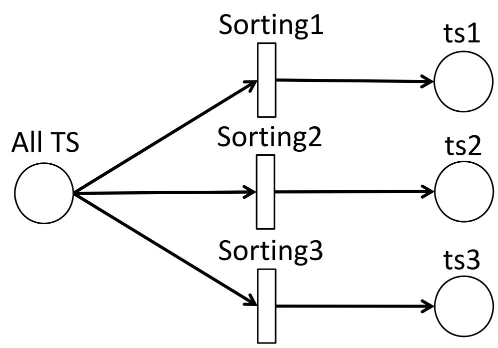

Figure 4.5: The PN model of the time slot sorting of the future cases

Serving Process

Figure 4.7 is a model of the serving process of MRI_1. As mentioned in Section 4.1, when patients come, many preparations are required to be done ahead of the examination (e.g. registration, filling screening forms, changing clothes, etc.), which are integrated into a transition "Preparing" in the PN model (Figure 4.7). After the preparation is done, patients wait in "Waiting room" and are ready to enable the transition "Scanning" when there is a token in "MRI" (i.e. the machine is available). After scanning, patients are moved to "Finished cases" with the release of machine. Then, the finished routine cases are moved to "PAS1" and leave the system with an output of their waiting times in "WT2 1", "WT3 1" and "WT4 1" based on their priority levels. For future cases, they are also moved to "PAS1" and leave the system after serving, but their waiting times are not calculated in that the delay for them is mainly caused by personal reasons (e.g. the preference of a patient, the suggestion of a specialist) rather than the problem of the system, and thus it is meaningless to test the performance of system with their waiting times.

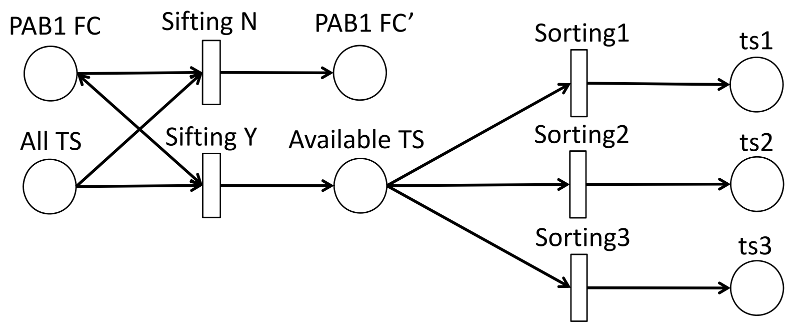

Figure 4.6: The PN model of the time slot sorting of level 3 cases

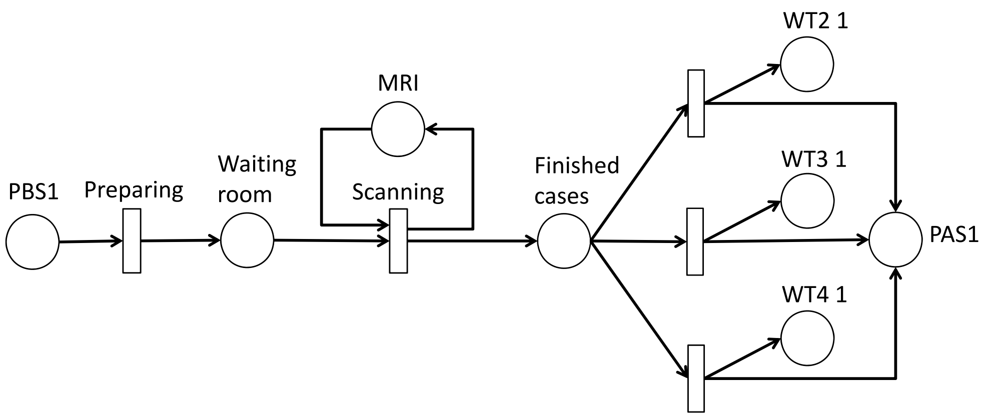

Figure 4.7: The PN model of serving process for MRI_1

4.2.2 Model for Semantics 2

In the model for Semantics 1, the location of two machines is not considered. If two machines are located in different sites (i.e. one machine in one site), it is necessary to allow patients to choose a preferred scan site. To realize this function, the available scan sites can be numbered in advance. The numbers are stored in an additional place "PSS" in the overall process (see Figure 4.8). The coming patients are originally put in "POS" and are allowed to choose the number of preferred scan site from available ones in "PSS", which is added to the patients' information. Hence, the tokens in "PBB" are in the pattern of $(f, (c, e), p, d_g, d_f, s)$, ($s$ is the scan site number). Meanwhile, the number of chosen machines is returned to "PSS" for the selection of the next patient. For example, if two sites are modeled, three options are provided as $s = 0, 1, 2$, where "1" refers to site 1 and "2" is about site 2, while "0" means no choice (i.e. with the preference of the first available site).

4.2.3 Model for Semantics 3

In this model, the patients' preference for time slot is considered. Taking the booking process of level 3 cases for MRI_1 for example (Figure 4.9), patients can choose a time slot among available ones in "ts1", "ts2" and

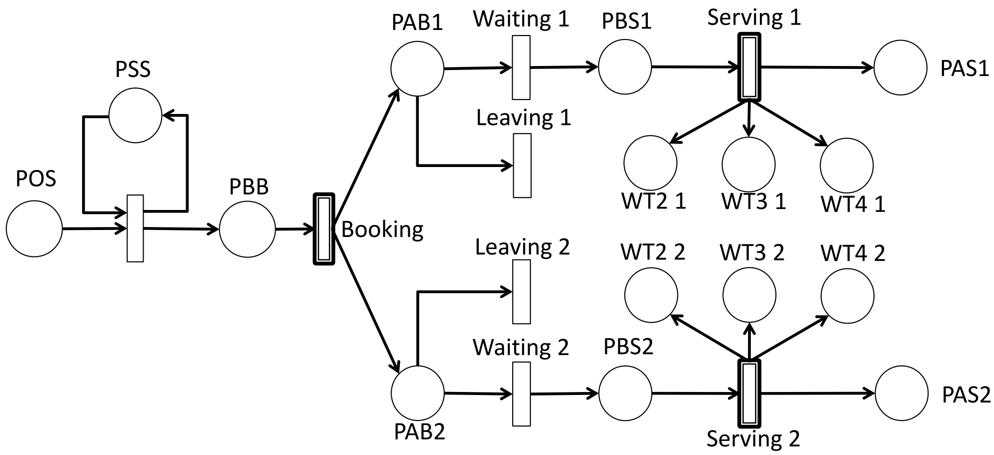

Figure 4.8: The PN model of the overall process with optional scan site

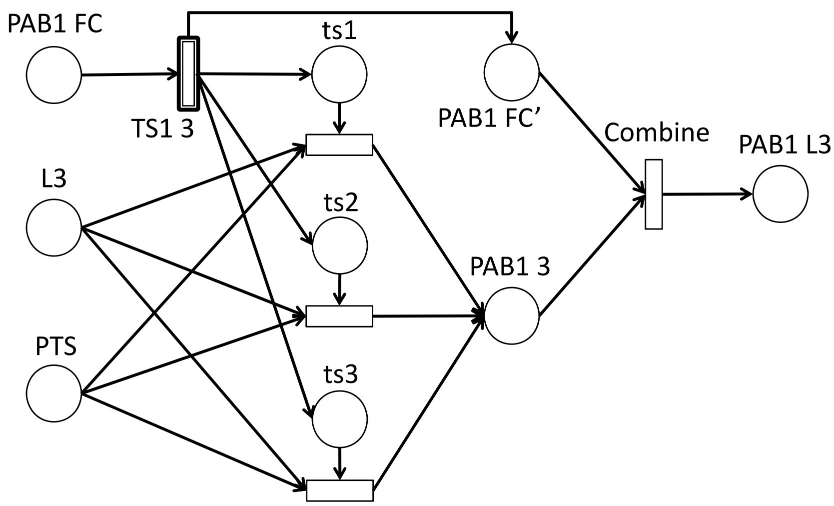

Figure 4.9: The PN model of the booking of level 3 cases with optional time slot

"ts3". The selected time slot is then stored in the place "PTS" with the pattern (d, n) (d is the date and n is the number of time slots) and assigned to the patient who requires it. Meanwhile, the occupied time slot is removed from the container of available time slots. For those patients who do not have any preference for scan time, they can input $(0, 0)$ in "PTS", which means that they would like to be booked to the first available time slot.

4.2.4 Model for Semantics 4

With the assumption that a scan may last shorter or longer than anticipated (i.e. the service time is unpredictable), the start time and end time of the service should be recorded. Figure 4.10 exhibits the model of the serving process for MRI_1. In the figure, the start time of a case is stored in "Start time". Patients in "Waiting room" enable the transition "Scanning" if there are tokens in both "MRI" and "Start time". When

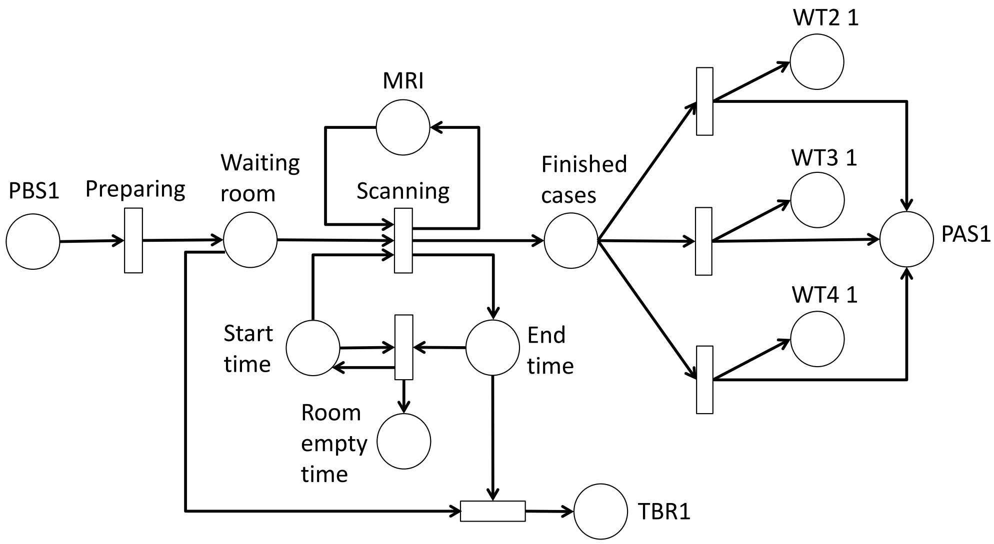

Figure 4.10: The PN model of serving process for MRI_1 with service re-ordering

the transition “Scanning” fires, the service time of the case is generated. After scanning, patients are moved to “Finished cases” with the release of machine and an output of “End time” that is equal to the summation of start time and service time. A threshold is set (according to the department hours) as an upper limit of end time. If the end time does not exceed the threshold, there are two possibilities: (1) the end time is earlier than the start time of the next patient, (2) the end time is later than or equal to the start time of the next patient. For the first possibility, there is a period of room empty time recorded in the place “Room empty time”. As for the second possibility, there is no room empty time and the value is recorded as 0 in “Room empty time”, but there is a period of overtime. Then, the service of the next patient is delayed. If the end time exceeds the threshold, the following patients in “Waiting room” have to be transferred to “TBR1” to re-order the service. The tokens in “TBR1” restore the initial pattern $(f, (c, e), p, d_g, d_f)$ removing the obtained time slot. As an outgoing place of transition “Serving 1”, the place “TBR1” also appears in the net of overall process shown in Figure 4.11 to move the impeded patients to waiting list for re-ordering. These patients are put in the head of waiting list, so that they are given a priority to be booked in an early date.

4.3 Model Validation

This section focuses on the validation of the model for Semantics 1. The models for Semantics 2 to 4 are not validated due to lack of data. However, the validation of the model for Semantics 1 can well support the effectiveness of the models for Semantics 2 to 4, as the model for Semantics 1 is the foundation of the entire model. In the following, the collected data for validation are introduced and deficiencies of the data are analyzed. Then, the model-simulated waiting times are compared with the real data. A statistical analysis of the comparison results has shown that the two are sufficiently close.

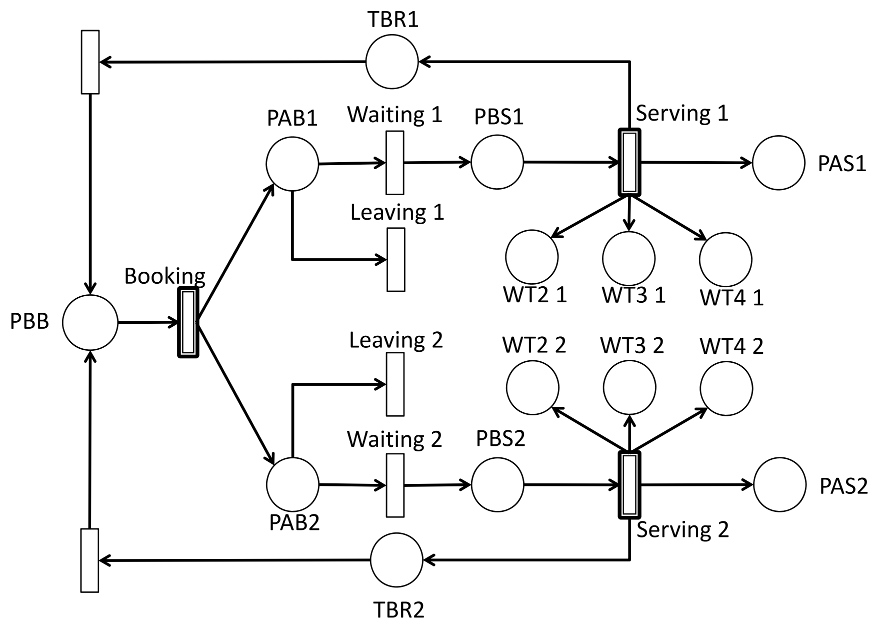

Figure 4.11: The PN model of the overall process with service re-ordering

Table 4.3: Overview of collected real data

	PL	RD	ESD	ED	ST	WT	MA
1	…	…	…	…	…	…	…
2	…	…	…	…	…	…	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2720	…	…	…	…	…	…	…

4.3.1 Data Collection

The data used in the model are the data of outpatient requests for MRI examination from January 1, 2012 to September 30, 2012 stored in the database of Saskatoon Health Region - Royal University Hospital. The data are related to 2,720 outpatients served by two MRI machines (MRI_1 and MRI_2) in RUH, and the data contain the information of (1) priority level (PL), (2) date of request for MRI (RD), (3) the earliest scan date of future case (ESD), (4) date of the examination (ED), (5) start time of the examination (ST), (6) waiting time (WT) and (7) service machine (MA) (see Table 4.3). Unfortunately, another two groups of data: with contrast or not and with review or not, which are required in the model, are not recorded in the database. They were then produced by the experience. In practice, these two sets of data are also possibly obtained from the patients' medical documents.

Apart from the problem of missing data, there are also some other deficiencies in the obtained data. First, the start times of some cases are out of department hours. They may be sudden or special cases that seldom occur, so they were treated as outliers and eliminated from the data set. Second, several cases are repeatedly

Table 4.4: Summary of the deficiencies of collected data

Deficiency of Data	Number of Cases
Cases with start times out of department hours	8
Cases with repeated records	14
Cases recorded with different earliest scan dates	37
Cases with negative waiting times	5

Table 4.5: The number of patients in each category with respect to each machine

		MRI_1	MRI_2
Future Case		245	295
Routine Case	Level 2	379	478
	Level 3	374	400
	Level 4	234	251

recorded. Only one record was kept. Third, some cases are recorded more than once with different earliest scan dates. For this situation, only the record with the latest update was taken. Finally, a few abnormal records are found in which the date a case is served is earlier than the date the case is generated and the waiting times for them are reported as negative numbers. These cases are impossible to happen in reality; their presence may result from some input error. Thus, they were removed from the data as well. The number of data with deficiencies is summarized in Table 4.4.

After the aforementioned treatment, the data relating to 2,656 patients remain, which are further shown in Table 4.5. As mentioned in Section 4.2, three situations are considered to classify the patients. The percentages of patients in situation 1, 2 and 3 are 37.7%, 29.1% and 33.2%, respectively. After booking, 4.5% of the patients leave the system without being served. They may have in fact cancelled the service in advance or just not show at the appointed time.

4.3.2 Results

In this section, the information of patients that receive MRI services in a specific period is collected from the real data and taken as input to our model. The model simulates the real booking and serving processes, from which the scheduled time of each patient can be obtained and the simulated waiting time of each patient can be computed. To show the effectiveness of the proposed model, the simulated waiting times are compared with the actual ones. Since the waiting time as exhibited in reality as well as in simulated system involves much noise, a statistical approach is employed to compare the simulated data and real data.

In the real data, scheduled times of patients are recorded from June to September in 2012. The patients whose waiting times simulated in the model are level 2 patients who have received services in June and all

Table 4.6: The comparisons of waiting times in each category

		Simulation Mean (days)	Real Mean (days)	t	df	p-value
MRI_1	Level 2	12.83416	17.67781	3.5677	690.275	0.000385
	Level 3	66.07538	67.12991	0.4262	718.233	0.6701
	Level 4	82.47464	89.93464	1.9529	300.259	0.05176
MRI_2	Level 2	12.43846	17.37799	3.5192	800.164	0.0004573
	Level 3	61.1455	64.51064	1.687	416.693	0.09235
	Level 4	84.77512	87.83673	0.92	178.504	0.3588

patients who have received services from July to September. In practice, those patients of level 3, level 4 and future cases that received services in June may be scheduled before June and thus are not simulated in the model. Instead, their scheduled times in the real data are used directly in the model as the internal states of the system, i.e., the schedule times for those patients in the model taking exactly the same as that in the real data. Since in practice, level 2 cases are urgent and usually receive services quickly, their scheduled times in June are also simulated by the model. In this way, the model has a similar initial internal state with the real data such that the comparison of the simulated waiting times and corresponding real ones can reflect the effectiveness of the model.

Since the simulation should start in a state that some future cases and level 3 and 4 routine cases have been scheduled, and several available time slots are reserved for the coming of potential level 2 patients, the real booking situations in June 2012 (except for those of level 2 routine cases) are set up in the model as initial internal states and level 2 cases are booked first when the simulation start. The simulated waiting times for level 3 and 4 routine cases booked between July and September are compared with the real situations collected in this time period. As for level 2 cases, the waiting times to be compared span over four months from June to September in that the results in June are produced by simulation as well.

The simulated waiting times and real ones are compared on the basis of student t-test at the significant level of 0.05. The hypotheses are H_0: On average, the waiting times of these two sets are the same, and H_1: On average, the waiting times of these two sets are different. If a p-value is less than 0.05, H_0 is rejected.

The simulation on the two machines was run more than 30 times and their results are quite similar (according to [30], 30 times of run are quite sufficient). This result may be explained by the following fact: both being with the same inputs in each run and there are very small variations in the scheduling process for the two machines. In fact, the only uncertainty occurs when the two machines have the same number of available time slots and in this case, the next patient is randomly assigned to either one of the two machines. Nevertheless, this situation seldom occurs and thus this uncertainty has slight effects on simulated waiting times. The results of one run are shown in Table 4.6 and Table 4.7, respectively.

From Table 4.6, it can be seen that the mean values of simulated waiting times of each level are all smaller than the real ones, which may be caused by some uncontrolled events that could occur in reality but are

Table 4.7: Statistical summary of simulation and real waiting times in each category

			Sample Size	Standard Deviation
Simulation	MRI_1	Level 2	404	17.90434
		Level 3	398	34.58759
		Level 4	276	36.61787
	MRI_2	Level 2	390	18.38562
		Level 3	378	13.42639
		Level 4	209	15.17279
Real	MRI_1	Level 2	329	18.58368
		Level 3	331	32.11039
		Level 4	153	38.59006
	MRI_2	Level 2	418	21.47534
		Level 3	329	33.94356
		Level 4	147	38.28874

not considered in the model (e.g. maintenance of machine, careless management of the process, etc.). These neglected events may delay the process and increase patient waiting times. However, the differences for most types of cases are not significant. The p-values of t-test for level 3 and level 4 cases in MRI_1 and those in MRI_2 are all larger than 0.05, so there is no significant difference between the simulation and real results for level 3 and level 4 cases. Unfortunately, for level 2 cases in both machines, the simulation and real waiting times are significantly different. This may result from the interference of a few overly long waiting times in the real situations. The largest waiting times for level 2 cases booked to MRI_1 and MRI_2 are 117 and 188 days, respectively, longer than three months, which are unreasonable and unacceptable since patients in level 2 are urgent and usually cannot bear a long waiting time. Thus, the overly long waiting times in the data are considered as outliers and ignored during the testing. By removing these outliers, the remaining data are compared with the simulation data again, which are shown in Table 4.8 and Table 4.9. It can be seen from Table 4.8 that p-values for level 2 cases in both machines are larger than 0.05, implying that without the effects of outliers, there is no significant difference between the simulation and real results. It is also indicated that, for both machines, the average waiting times obtained from simulation are shorter than those from real system.

In summary, the above discussions have shown that on average the difference between simulated waiting times and real waiting times is not significant, which demonstrates the effectiveness of the proposed model. The model is able to capture the main processes and factors of the real system and its performances can approach to the reality to some degree. Therefore, it should be a good model based on which many applications can be done and various characteristics of the real system can be explored.

Table 4.8: The comparisons of waiting times in level 2 cases

	Simulation Mean (days)	Real Mean (days)	t	df	p-value
MRI_1	12.83416	14.08013	1.2035	634.227	0.2292
MRI_2	12.43846	13.95262	1.4492	579.171	0.1478

Table 4.9: Statistical summary of simulation and real waiting times in level 2 cases

		Sample Size	Standard Deviation
Simulation	MRI_1	404	17.90434
	MRI_2	390	18.38562
Real	MRI_1	312	9.319791
	MRI_2	401	9.497381

4.4 Conclusion

This chapter developed models to describe the whole process of MRI system from booking to serving. Six pieces of semantics were identified to be captured in the model, in which the first semantics is a foundation of the other five. The PN models for the first four pieces of semantics were constructed. This chapter omitted many details including arc expression, guard of transition and other nodes aiding to realize some functions, which are all contained in the real model (see Appendix A). Though the models for Semantics 5 and Semantics 6 were not developed in this study, it is not difficult for them to be constructed based on the model for Semantics 1 in future. In the model validation, the model for Semantics 1 was validated through the comparison of the waiting times obtained from the model simulation and real record. To validate the models for Semantics 2 to 4, more real data are required. However, the validation of the model for Semantics 1 can well support the effectiveness of the models for Semantics 2 to 4, as the model for Semantics 1 is the foundation of the entire model.

Chapter 5

Model Applications

The application of this model is explored mainly in two aspects: decision making and prediction. Each application is explained with an example. Section 5.1 demonstrates how the model can be used to guide the decision makers to select an optimal plan. Section 5.2 shows that the model can be used to forecast the future waiting time when the current distribution of regional patient comings changes. The conclusion of this chapter is given in Section 5.3.

5.1 Decision Making

As an example of decision-making, the model was used to guide the selection of a scheduling strategy to determine the proportion of MRI time slots assigned to each category of patients. Since the capacity of MRI machine is limited, only a couple of time slots are available daily for patients. If patients in one category occupy too many time slots, those in other categories own less access and have to wait longer. Hence, short waiting times of a part of patients are at the expense of long waiting times of others. To balance the waiting times of patients in different categories, an ideal scheduling strategy of time slot allocation is important for the MRI decision makers. However, it is impractical to try for all possible strategies in MRI department to find out the best one. With the developed PN model, the problem can be solved.

The data of patients' arrivals used in this application are generated from the distribution of real data. Though the real data can be applied directly, there are some advantages to employ the generated data. First, comparing to the limited number of real data, the computer program can generate as many data as possible to satisfy the requirement of simulation. Second, the real data are too specific for one region within a fixed time period and may be disturbed by some outliers, while the generated data focus on the distribution to catch the main features of data. Finally, it is easier for decision makers to master the generated data through an alteration of parameters of the distribution when the pattern of patients' arrivals changes.

The probability mass function of the number of patients' requests for MRI is fitted with a mixed Poisson distribution. A Poisson distribution describes the probability of a given number of events happening in a fixed interval of time. The histogram of frequency of the number of daily requests for MRI is plotted in Figure 5.1. It can be seen from the figure that the frequency of a small number of requests is in a largest value up to 0.2. With the increase of requests, the frequency plunges into the bottom, close to 0 for 10 requests, and

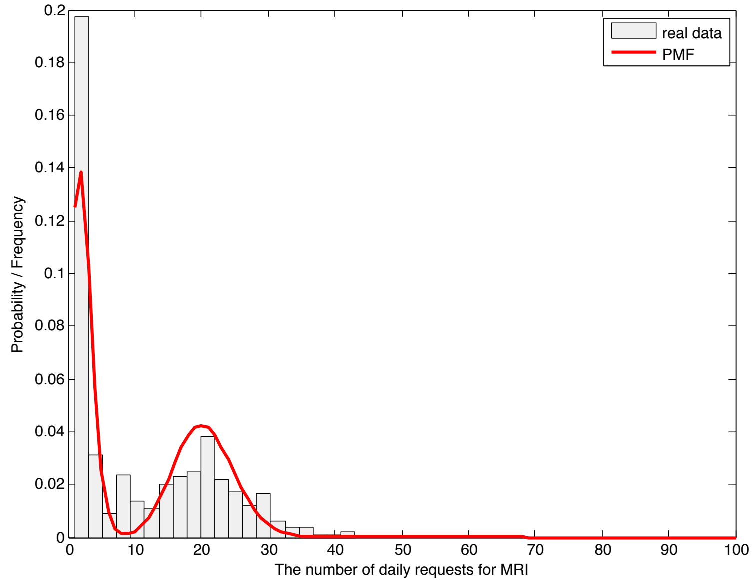

Figure 5.1: The distribution of the number of daily requests for MRI and the fitted probability mass function

bounces back to a peak at 0.04 where 20 requests come, and then declines to 0 for around 35 requests before leveling off. From the figure, it is found that the data do not belong to the population of a single Poisson distribution; instead, they can be considered coming from two Poisson distributions with different parameter values. Therefore, to combine these two distributions, a mixed Poisson distribution is proposed to fit the data.

Let $x_1, \ldots, x_n$ be the observed number of patients coming each day. We assume that they can be divided into two groups. For each observation x_i, a random variable z_i is defined to indicate the group that x_i belongs to, i.e.,

$$x_i \in \text{group 1} \quad \text{if } z_i = 1 \quad \text{and} \quad x_i \in \text{group 2} \quad \text{if } z_i = 0$$

We assume that z_i follows a Bernoulli distribution with the parameter of probability of success, γ, i.e. $z_i \sim Bernoulli(\gamma)$. Conditioning on different values of z_i, x_i has a mixed Poisson distribution, i.e.,

$$x_i | z_i = 1 \sim Poisson(\lambda_1) \quad \text{and} \quad x_i | z_i = 0 \sim Poisson(\lambda_2)$$

In order to generate data from the distribution of x_i, the parameters $\theta = \{\gamma, \lambda_1, \lambda_2\}$ need to be estimated. As the observations of the latent variables z_i are unknown, the famous Expectation Maximization (EM) algorithm is used here. EM algorithm is a technique used in point estimation and very suitable for the case containing latent variables or missing values. It consists of two major steps: the E-step and M-step. E-step computes the expectation of log likelihood of completed data with respect to the latent variables conditioning on the observations and current estimate of the parameters. M-step updates the estimate by maximizing the expectation function obtained in the E-step. Given initial values of those parameters, EM algorithm iterates these two steps to update the parameters such that the log likelihood function keeps increasing until it converges (see Appendix B for more details).

Using the EM algorithm, parameters of the mixed Poisson distribution are estimated as $\gamma = 0.5184$, $\lambda_1 = 2.2198$ and $\lambda_2 = 20.4564$. The fitted line is plotted in Figure 5.1. It can be seen that the mixed Poisson distribution is a good fit for the real data and hence suitable to generate our simulation data. The parameter λ in Poisson distribution represents the mean value of the number of events occurring in a specific time interval, which is the average number of patients requesting MRI each day. The estimated results indicate that the average numbers for the first group is quite small, only around 2. In contrast, the average number for group 2 is very large, around 20. Interestingly, the possibility of the number of patients requesting MRI each day falling into either group is almost equal, as 0.5184 for group 1. This phenomenon reveals that the requests in each day are extremely unstable or uneven, which may be either pretty rare or pretty abundant.

In the received requests, the ratio between future cases and routine cases is determined on the basis of real data. For future cases, the probability mass function of time interval between the request date and the earliest scan date is fitted with a geometric distribution expressing the probability of the first success of Bernoulli trials after a series of failure. Bernoulli trials refer to the independent repeated trials of an experiment with only two outcomes: success and failure. In this problem, whether a future case is available to be served at one day after the request date can be regarded as the outcomes of Bernoulli trials, with "yes" and "no" mapping to success and failure, respectively. The outcome of the daily trial is always failure until the earliest scan date comes when the first success occurs. The histogram in Figure 5.2 shows the frequency of the number of days between the request date and the earliest scan date. With the increase of days, the frequency decreases dramatically from near 0.09 to 0.01 at 20, and then continues dropping gradually and keeps in a low level with slight fluctuations, complying with the pattern of geometric distribution.

The only parameter p in the distribution is estimated as $p = 0.0907$ and the fitted curve is plotted in Figure 5.2. The parameter p in the geometric distribution is the probability of success for each trial, i.e. the probability for a future case to be available for booking each day. A small estimated value of p ($p = 0.0907$) indicates that there is little chance for the earliest scan date of a future case to be fixed at a day. The result implies that though most future cases require for a short time interval between the request date and the earliest scan date, some cases prefer to receive the service after a long time (e.g. two months).

Except for the earliest scan date, a factor that is unique for future cases, the other factors involving in the model (e.g. priority level, with/without contrast) exist in both future cases and routine cases and are evaluated here on the basis of their proportions in real data.

With the complete information for all the requests, the booking and serving process can be simulated with the model. In the booking stage, the number of all available time slots each day is fixed as mentioned in the model section (13 on weekdays and 7 on weekends) and it is further assumed that the proportion of time slots reserved for each category of cases every day is fixed. Six strategies are considered as the alternatives of time slot allocation as shown in Table 5.1. For each strategy, the allocations on weekdays (WD) and Weekends (WE) are different. For instance, in Strategy 1, the proportion of time slots assigned to future cases and level

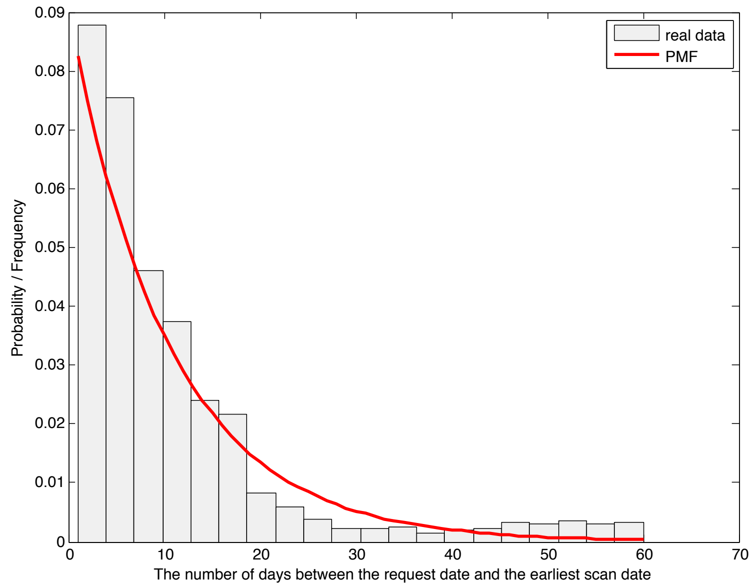

Figure 5.2: The distribution of the number of days between the request date and the earliest scan date and the fitted probability mass function

Table 5.1: The proportion of time slots assigned to each category of cases by strategy

		Strategy 1		Strategy 2		Strategy 3		Strategy 4		Strategy 5		Strategy 6	
		WD	WE	WD	WE	WD	WE	WD	WE	WD	WE	WD	WE
Future Case		5	2	5	2	5	2	5	2	5	2	5	2
Routine Case	Level 2	3	2	3	2	2	2	4	2	1	0	6	4
	Level 3	3	2	4	2	4	2	3	2	6	4	1	0
	Level 4	2	1	1	1	2	1	1	1	1	1	1	1

2, 3 and 4 routine cases on WD is 5:3:3:2, while on WE, it changes to be 2:2:2:1. In all strategies, the number of time slots reserved for future cases is the same, either on WD or on WE, since we are not interested in the change of waiting time for future cases, which is not even measured in the model; instead, the variation of waiting time for routine cases in different priority levels is focused on when the allocation of time slots alters. Furthermore, the first four strategies are similar with slight adjustments to the proportion on WD but not on WE. Though it does make sense to do some alterations on WE, the partial adjustments are enough to manifest how time slot allocation impacts on waiting times. The Strategy 5 and 6 are two extreme situations with most available time slots assigned to one category of cases (to level 3 patients in Strategy 5 and to level 2 patients in Strategy 6). Apparently, they are not good strategies for booking. To investigate if they are as unreasonable as expected, their performances are compared in this example.

After simulation, the average waiting times of level 2, 3 and 4 cases under each strategy is measured and plotted in histograms in Figure 5.3, 5.4 and 5.5, respectively. The red line in each figure is the error

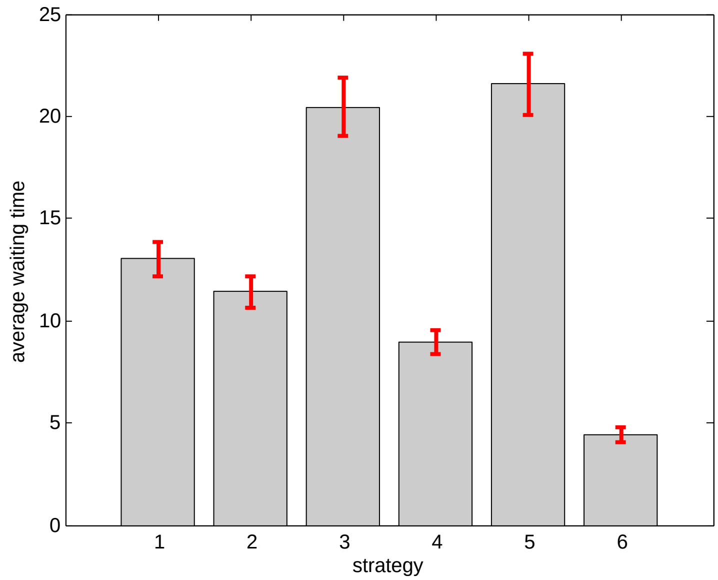

Figure 5.3: The average waiting time (days) of level 2 cases versus strategy

bar (i.e., the interval is 95% confidence interval). In Figure 5.3, the strategy produces shortest waiting time is Strategy 6, around 5 days, followed by Strategy 4, 2 and 1, all between 6 and 15 days. Strategy 3 and Strategy 5 force level 2 cases to wait over 20 days, in which Strategy 5 is worse with a longest delay close to 22 days. According to the resource allocation in Table 5.1, the number of time slots assigned to level 2 cases in Strategy 6 is the largest (6 on WD and 4 on WE), which decreases gradually in Strategy 4, 2, 1 and 3, and gets to the smallest in Strategy 5 (1 on WD and 0 on WE). It indicates that the length of waiting time is negative correlated with the number of time slots. On the basis of shortest waiting time, Strategy 6 is the best choice and Strategy 5 is the worst for level 2 cases. Figure 5.4 shows the results of level 3 cases. As expected, Strategy 5, with an assignment of most time slots (6 on WD and 4 on WE), generates the shortest waiting time, near 40 days, while the longest waiting time appears in Strategy 6 with only one time slot reserved on WD and none on WE, up to around 118 days. Strategy 1 and 4, with the same time slot allocation (3 on WD and 2 on WE), produce similar waiting times, so do Strategy 2 and 3, but shorter in that one more time slot is added on WD. Undoubtedly, for level 3 cases, Strategy 5 is the first choice and Strategy 6 is the last. As for level 4 cases, given that Strategy 1 and 3 lead to almost the same short waiting times (see Figure 5.5), either one can be selected. The values of average waiting time for all levels of cases by strategy are exhibited in Table 5.2. The data in parentheses are the standard deviations.

As no best choice can be made to minimize the average waiting times of all levels of cases, it is thought to combine the waiting times and pick up a strategy with a minimal average of the combination. However, the waiting times of different levels of cases cannot be combined directly. Though they are all measured in the unit of day, the severity of one day's delay for level 2 cases is higher than for level 4 cases. A waiting of a couple of weeks or months is acceptable for level 4 cases, while for level 2 cases, a few days' delay may be life-threatening. To eliminate the distinction in the severity of delay, the waiting time of urgent cases should

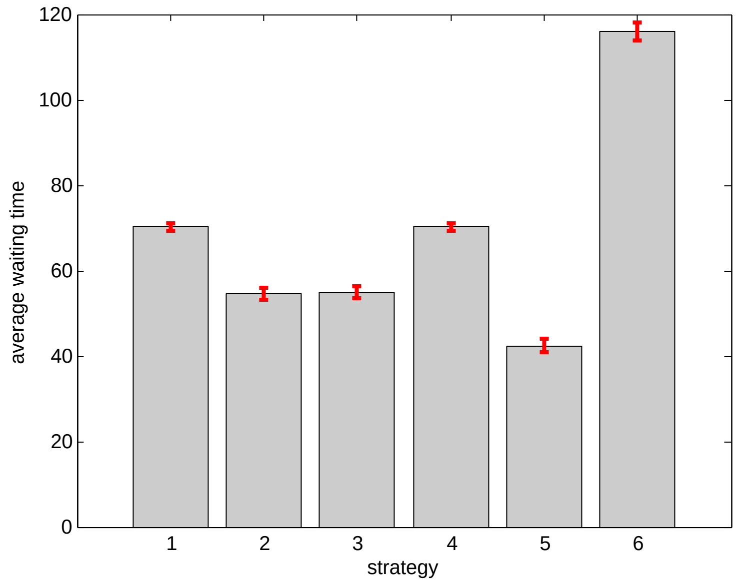

Figure 5.4: The average waiting time (days) of level 3 cases versus strategy

Table 5.2: Means and standard deviations of waiting times (days) for all levels of cases by strategy

	Strategy 1	Strategy 2	Strategy 3	Strategy 4	Strategy 5	Strategy 6
Level 2	13.02	11.4	20.46	8.96	21.59	4.42
	(0.42)	(0.41)	(0.72)	(0.29)	(0.77)	(0.19)
Level 3	70.2	54.68	54.79	70.23	42.41	116.06
	(0.43)	(0.71)	(0.71)	(0.43)	(0.83)	(1.05)
Level 4	47.08	77.86	46.85	78.34	74.07	77.6
	(1.12)	(0.50)	(1.13)	(0.50)	(0.61)	(0.48)

be equipped with a larger weight. The weights can be understood as coefficients for the conversion of waiting times between two levels of cases. For instance, if the weights of level 2 and 4 are 10 and 1, respectively, then one day's delay for level 2 cases is equivalent to ten days' delay for level 4 cases. The weighted waiting times are measured in the same standard and can be combined, so that the weighted average waiting time can be calculated.

Let a, b and c be the weights of waiting times of level 2, 3 and 4 cases, the weighted average waiting time (WAWT) is

$$\mathrm{WAWT} = \frac{a \times \mathrm{AWT}_2 \times n_2 + b \times \mathrm{AWT}_3 \times n_3 + c \times \mathrm{AWT}_4 \times n_4}{n_2 + n_3 + n_4},$$

where AWT_i is the average waiting time of level i cases and n_i is the number of cases in level i, $i = 2, 3, 4$. The weights a, b and c can be determined by experts based on their experience. In this example, we suppose the weights are $a = 16$, $b = 4$ and $c = 1$. The WAWT of each strategy is measured and plotted in Figure 5.6 and the value is summarized in Table 5.3. It can be seen that Strategy 3 gives the longest WAWT, followed

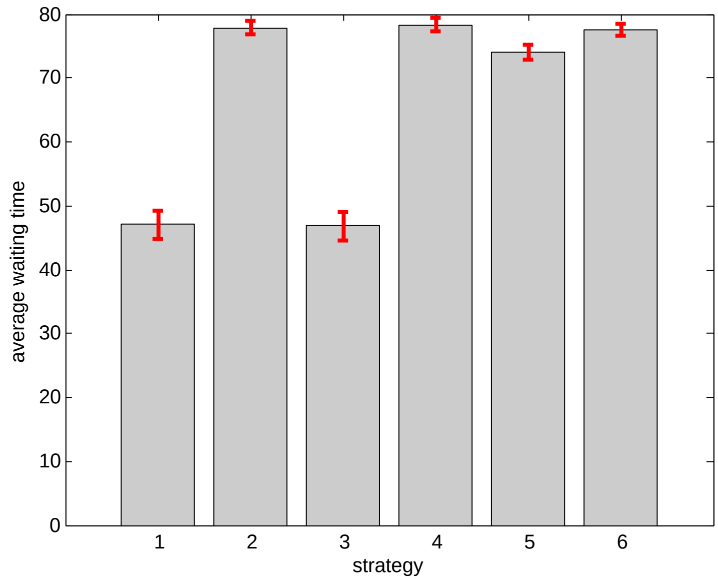

Figure 5.5: The average waiting time (days) of level 4 cases versus strategy

Table 5.3: Means and standard deviations of weighted waiting times (days) by strategy

	Strategy 1	Strategy 2	Strategy 3	Strategy 4	Strategy 5	Strategy 6
WAWT	194.36	176.84	203.08	191.28	200.57	185.31
	(2.79)	(2.63)	(4.00)	(2.36)	(4.46)	(4.35)

by Strategy 5, a little bit shorter, and then Strategy 1, 4 and 6, which reduce slightly in order. The shortest waiting time is generated via Strategy 2, far lower than others. Though Strategy 2 is not an ideal choice in the discussion of any single level, its general performance is the best and can be selected to balance the waiting times. What makes us surprised is that the performances of the two extreme situations, Strategy 5 and 6, are not so bad. Especially for Strategy 6, it provides a second shortest average waiting time. Thus, an intuitive judgment may mislead the decision, and comparatively, the result obtained by modeling is more reliable.

5.2 Prediction

Another example is to explain how the model is used to predict the future waiting time when the number of patients' requests for MRI changes. In this example, we assume that the average number of requests is doubled. To compare the results, two groups of data on patients' requests are involved. The first group is the data of current patients' requests generated in the preceding section by the mixed Poisson distribution with the parameters estimated as $\gamma = 0.5184$, $\lambda_1 = 2.2198$ and $\lambda_2 = 20.4564$. The second group is the data of increased patients' requests generated with two parameters, λ_1 and λ_2, doubled, i.e. $\gamma = 0.5184$, $\lambda_1 = 4.4396$

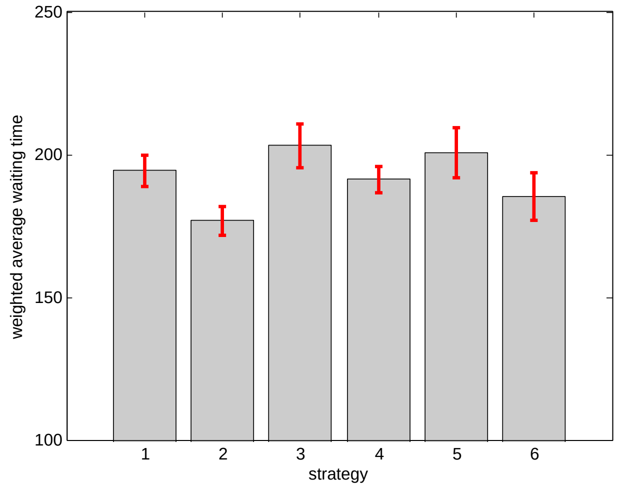

Figure 5.6: The weighted average waiting time (days) versus strategy

and $\lambda_2 = 40.9128$, and the factors of patients' information are evaluated with the same proportions as the first group does. For the allocation of time slots, we use Strategy 2, which provides the shortest WAWT under the given weights.

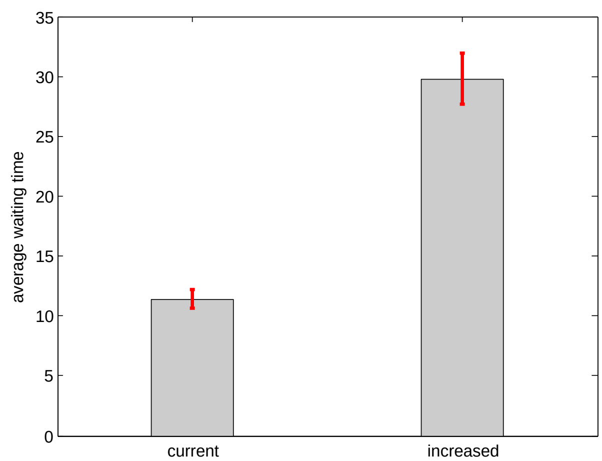

Figure 5.7: The average waiting time (days) of level 2 cases on current and increased requests

The simulated average waiting times of level 2, 3 and 4 cases are plotted in Figure 5.7, 5.8 and 5.9, respectively, and the values are summarized in Table 5.4. It can be seen that the average waiting times of the situation with increased requests in each level are far larger than those of the current situation. More specifically, for both level 2 and level 3 cases, the ratios of the values of increased requests to those of current requests are over 2:1, while for level 4 cases, the ratio is a little bit lower than 2:1. The result reveals that a sudden increase in patients' requests leads to a heavy attack on patients' access to MRI service, and comparatively, the impact on the level 2 and 3 cases is greater than on level 4 cases.

Combining the waiting times with the weights set in the preceding section, the WAWTs are calculated as

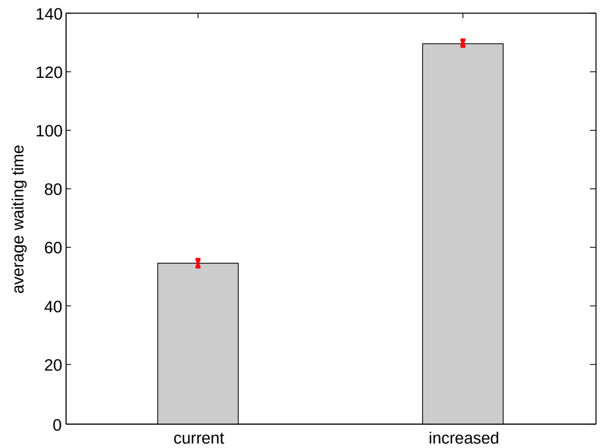

Figure 5.8: The average waiting time (days) of level 3 cases on current and increased requests

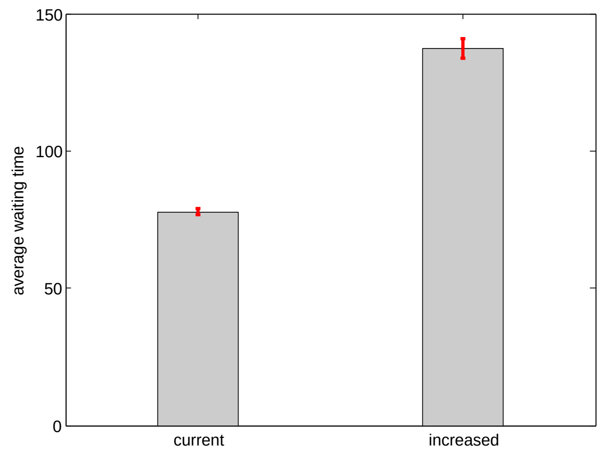

Figure 5.9: The average waiting time (days) of level 4 cases on current and increased requests

shown in Table 5.4 and plotted in Figure 5.10. Based on the data, it is found that the WAWT of increased requests are up to 2.5 times of that of current situation.

With the above simulation, the direction and magnitude of variation of average waiting times can be predicted. The simulation results provide some evidences to warn the impending high waiting times and leave time for administrators to take some effective precautions, e.g. purchasing more machines or referring a part of patients to other sites.

Besides the change of the number of patients' requests for MRI, the composition of other factors may vary (e.g. the proportion of level 2 cases rises, the proportion of cases with contrast goes down), and its effects on waiting times is predictable as well, so that the related settlements can be determined and carried out before it is too late.

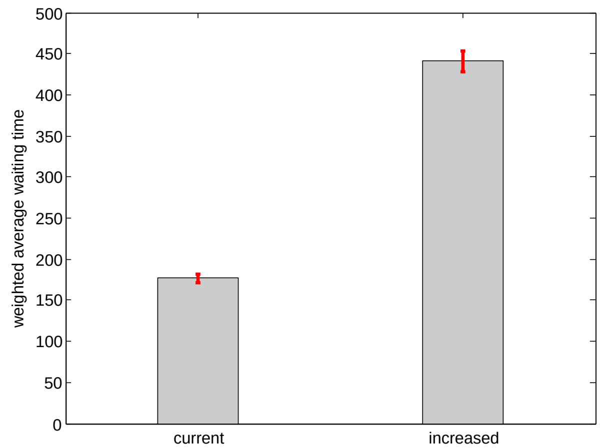

Figure 5.10: The weighted average waiting time (days) on current and increased requests

Table 5.4: Means and standard deviations of waiting times (days) for each level and their weighted waiting times (days) on current and increased requests

	Current	Increased
Level 2	11.4	29.8
	(0.41)	(1.08)
Level 3	54.68	129.6
	(0.71)	(0.58)
Level 4	77.86	137.17
	(0.50)	(1.80)
WAWT	176.84	440.75
	(2.63)	(6.64)

5.3 Conclusion

In this chapter, two main applications of the proposed model were introduced. In the first case, it can be used to analyze the performance of different strategies and help the administrators to make an informed decision. In the second case, it can forecast the waiting time in the condition that the number or composition of patients' requests for MRI changes and make it possible for administrators to take some necessary precautions. In general, the model is useful to bring benefits to the management of the MRI booking and serving system.

Chapter 6

Conclusion

6.1 Overview

This thesis presented a study on the modeling of MRI booking and serving system (MRI system for short) with Petri nets, based on which the processes in MRI system can be simulated and waiting times of the system can be measured. The model has potential applications in decision-making and prediction and is useful for improvements of the management of MRI systems.

This research put an emphasis on the booking process of the MRI system, which has not been considered in the existing literature to the author's best knowledge. This booking process is crucial for an appointment-oriented system and worth being studied, as it is a key process responsible for waiting times. In the condition that medical resources are limited, booking strategies are sensitive to waiting times. A proper booking strategy is able to optimize the scheduling of patients and reduce their waiting times.

As the research scope of MRI system was determined, the research methodology was selected through comparisons among several popular methods. Related literature was reviewed to analyze the pros and cons of each method, and it was found that Petri nets (PNs), a graphic tool with powerful mathematical foundations to simulate discrete event dynamic systems, should be a sound choice to model the MRI system.

The overall objective of this study was to provide a solution to the problem of long waiting time in Saskatchewan through a better scheduler and the specific objectives are restated as follows.

Objective (1): Construct a model to describe the whole process of MRI examination and to quantify the waiting time of patients.

Objective (2): Investigate the effectiveness of the model and its potential applicability in the context of planning and scheduling of MRI examination in terms of simulation and predication of the waiting time of patients.

The above objectives have been achieved. For objective (1), a PN model was constructed to describe the booking and serving systems with an output of patients' waiting times. To achieve objective (2), the simulated waiting times were compared with real ones by student t-test, the results of which demonstrate the effectiveness of the model. The applications of the model were presented with examples in two aspects: decision making and prediction.

With an effective model, our goal may be achieved. For one thing, the performances of different management rules (or strategies) can be compared, which guides the MRI administrators to select an optimal strategy to shorten waiting times. For another, when the number or composition of patients' requests for MRI changes, the waiting times of the system can be predicted, which can caution the MRI administrators to make a quick and informed decision on if more resources (e.g. machines) are required.

6.2 Contributions

There are four main contributions out of this research.

First, the whole process of MRI system, including booking and serving processes, is considered. Some former studies are found to model and analyze the serving process in radiology department only [9, 36, 33]. However, combining booking and serving processes in one model is necessary in that there is a tight interaction between them. In the booking process, patients' waiting times are almost determined, while in the serving process, some accidents may occur (e.g. if a patient cannot be served at the appointed date, his/her service has to be postponed and the waiting time is prolonged), which may further affect the booking process. Modeling the whole process can establish the connections for these two processes and make the model more effective.

Second, a complete PN model is designed to model the MRI system. To select a method to describe the MRI system, the common modeling methods are compared and analyzed. It is found that queuing theory is unsuitable for this study owing to the limitations in the description of real systems and the lack of a clear distinction between the structure and behavior of a system. SD is not an ideal choice as well in that it is not good at modeling a patient appointment system and cannot capture an individual patient's information, while DES, in particular PNs, deems to be the best choice for modeling the problem of interest in the present study.

Third, two new extensions to the PN formalism are proposed to refine the structure of model constructed by the existing extended PNs. One extension, OCPN, is defined to arrange the resources in order, and the other, PHCPN, is to control the occurrence of events with priorities to avoid some conflicts. The two extensions can simplify the model structure by using fewer nodes to realize the same functions.

Fourth, on the basis of the model for Semantics 1, the models for Semantics 2 to 4 are developed. In this study, several pieces of semantics are proposed before modeling to guide the development of model. Semantics 1 guides the modeling of the basic booking and serving process of MRI system. Based on Semantics 1, Semantics 2, 3 and 4 are described and the corresponding models are constructed to enhance the performance of the system in some degree. Semantics 2 and 3 may improve the way of booking to make the system more flexible and patient friendly. Semantics 4 discusses a situation that commonly happens in reality to make the model more accurate to the real situation.

Besides, the model for Semantics 1 is verified to be effective to describe the real system. The model has important applications in the decision-making of the selection of booking strategies and the prediction of possible waiting times of the system in the future, which will assist MRI administrators in the management of medical resources and improve the quality of MRI service.

6.3 A Special Note to the Need of PN Model

The purpose of this research was to develop a more accurate model for describing the MRI examination process to capture as many real situations as possible and demonstrate its potential application to reduce the MRI waiting time. The MRI system is composed of the entities (patients, medical staff, MRI machine) and events (e.g. service re-ordering). Based on this model, the various possible approaches to schedule MRI examination can be simulated to find the best one in terms of the shortest waiting time with respect to individual patients.

To find an appropriate modeling method for the problem, several popular methods were analyzed and compared. Besides the queuing theory and system dynamics discussed in Chapter 2, another method, queuing network, is commonly used in complex system modeling as well. Queuing network [31, 32, 21, 7, 49] studies a system of a finite number of queues which are organized with specific relations. Customers in the system travel through the network to be served at service facilities. In open networks, customers can join and leave the system; whereas in closed networks, the number of customers in the system is fixed. Queuing network is an important application of queuing theory as many assumptions and conclusions in queuing theory are used in queuing network. Compared with queuing theory, queuing network is more powerful in system modeling since more than one service facility can be considered.

However, queuing network is not a good choice for modeling MRI examination scheduling systems for the purpose of this study as well as the future extended study due to the following reasons. First, queuing network cannot model the booking process, in which time slots are assigned to patients by rules. Second, queuing network cannot describe some particular dynamic events (e.g. leaving the queue after waiting some time, jumping the queue for emergent requesting for a care). Third, as queuing network assumes a service rule for each queue (e.g. FIFO, LIFO), it cannot model the flow of future cases, who are scheduled and served based on the desired earliest scan date rather than a regular order such as FIFO. Finally, in queuing network, if two patients leave their respective queues and join the same next queue, no order between them is considered. That is, for these two patients, it cannot identify which person should be served earlier. As mentioned in Semitics 4, the last patient in a day may need to re-order the service, who cannot be identified in a queueing network. Given that the booking process is a key process considered in Semantics 1 to 6 and future cases are involved in all the semantics, so here with queue network theory, Semantics 1 to 6 cannot be captured.

Comparatively, PNs are more suitable to model the process of MRI system. First, PNs do not have the above limitations and are able to capture various real situations of the system. In this thesis, six pieces of semantics were drawn from the real system and the first four are captured by PN models. Though the models for Semantics 5 and 6 are not modeled, they will be considered in the future work. Second, PNs have a solid mathematics foundation providing theoretical support to many properties. The analysis of properties can effectively detect the malfunction and redundancy of the model. For example, boundedness analyzes the upper and lower bounds of tokens appearing in all places. According to the results of this analysis, the place without tokens (i.e. the upper bound of tokens is 0) can be treated as a redundant place and can be deleted. Liveness detects the deadlock of a model, which should be fixed. The model properties will be analyzed in the future work.

However, with respect to the problem as considered in this thesis as well as in the near future, PNs still have some drawbacks. First, a large amount of data are required in simulation. Since various aspects of the real system are considered in PN models, a large amount information needs to be loaded into the model. One way to alleviate this problem is to further study the real system and pick up essential features or processes of the system and remove those redundant parts of the model such that only effective information in the data are collected. Second, the model structure is prone to be complex, i.e., the size of our model (the number of nodes and arcs) is very large. To reduce this complexity, the extensions to CPNs proposed in this research may be applied. Despite the above shortcomings, PNs are still the best choice in the modeling of MRI booking and serving system and they will be used to achieve the future work as well.

6.4 Future Work

This research has the following future work.

First, related real data can be obtained to validate the models for Semantics 2 to 4. The difficulty of this work is that some of the required data (e.g. service time) are not regularly collected and thus do not exist in the database. To collect these data, a long-term observation is needed.

Second, the model for Semantics 5 can be developed. This model will include the inpatients' information. The source of inpatients is complex relating to multiple departments. To investigate the arrivals of inpatients, all the related departments need to be observed and the inter-departmental relations should be learned, which is complicated but possible to be done.

Third, the model for Semantics 6 can be developed. For this semantics, the body part of scan is considered. This factor has effects on the anticipated service time and further affects the assignment of time slots. However, the anticipation of service time for each body part is an empirical value based on radiologists' experience and is hard to be obtained. To get the information, more efforts should be taken to consult the radiologists.

Finally, the properties of developed models, such as boundedness, liveness and reachability, can be analyzed using the existing theories in PNs to give some idea of the system without performing simulation.

References

[1] SusanL. Albin, Jeffrey Barrett, David Ito, and JohnE. Mueller. A queueing network analysis of a health center. *Queueing Systems*, 7(1):51–61, 1990.

[2] C. Arboleda, D. Abraham, and R. Lubitz. Simulation as a tool to assess the vulnerability of the operation of a health care facility. *Journal of Performance of Constructed Facilities*, 21(4):302–312, 2007.

[3] Rifat A. Atun, Ioannis Kyratsis, Gordan Jelic, Drazenka Rados-Malicbegovic, and Ipek Gurol-Urganci. Diffusion of complex health innovations–implementation of primary health care reforms in bosnia and herzegovina. *Health Policy Plan*, 22(1):28–39, Jan 2007.

[4] F.F. Baesler, H.E. Jahnsen, and M. DaCosta. The use of simulation and design of experiments for estimating maximum capacity in an emergency room. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, volume 2, pages 1903–1906 vol.2, 2003.

[5] Narges Bahi-Jaber and Dominique Pontier. Modeling transmission of directly transmitted infectious diseases using colored stochastic petri nets. *Mathematical Biosciences*, 185(1):1 – 13, 2003.

[6] N. Bailey. A study of queues and appointment systems in hospital outpatient departments, with special reference to waiting-times. *Journal of the Royal Statistical Society*, A14:185–199, 1952.

[7] Forest Baskett, K. Mani Chandy, Richard R. Muntz, and Fernando G. Palacios. Open, closed, and mixed networks of queues with different classes of customers. *J. ACM*, 22(2):248–260, 1975.

[8] Steffen Bayer, James Barlow, and Richard Curry. Assessing the impact of a care innovation: telecare. *System Dynamics Review*, 23(1):61–80, 2007.

[9] Neal M. Bengtson and Paul Meehan. Simulation of patient flow in a picture archival and communications system network. *International Journal of Bio-Medical Computing*, 23(12):137 – 153, 1988.

[10] M. Brahimi and D. J. Worthington. Queueing models for out-patient appointment systems – a case study. *The Journal of the Operational Research Society*, 42(9):pp. 733–746, 1991.

[11] Ewart R Carson. Decision support systems in diabetes: a systems perspective. *Computer Methods and Programs in Biomedicine*, 56(2):77 – 91, 1998.

[12] R. Cook and J. Rasmussen. "going solid": a model of system dynamics and consequences for patient safety. *Qual Saf Health Care*, 14(2):130–134, Apr 2005.

[13] Stefan Creemers and Marc Lambrecht. Queueing models for appointment-driven systems. *Annals of Operations Research*, 178(1):155–172, 2010.

[14] M. Criswell, I. Hasan, R. Kopach, S. Lambert, M. Lawley, D. McWilliams, G. Trupiano, and N. Varadarajan. Emergency department divert avoidance using petri nets. In *System of Systems Engineering, 2007. SoSE '07. IEEE International Conference on*, pages 1–6, 2007.

[15] Brian C. Dangerfield, Yongxiang Fang, and Carole A. Roberts. Model-based scenarios for the epidemiology of hiv/aids: the consequences of highly active antiretroviral therapy. *System Dynamics Review*, 17(2):119–150, 2001.

[16] Arnoud M de Bruin, A C van Rossum, M C Visser, and G M Koole. Modeling the emergency cardiac in-patient flow: an application of queuing theory. *Health Care Management Science*, 10(2):125 – 137, 2007.

[17] P DeLaurentis, Renata Kopach, Ronald Rardin, Mark Lawley, Kumar Muthuraman, Hong Wan, Leyla Ozsen, and Paul Intrevado. Open access appointment schedulingan experience at a community clinic. In *Proceedings of 2006 IIE annual conference and exposition, Orlando, FL*, 2006.

[18] Mitul Shivam Desai, Marion Louise Penn, Sally Brailsford, and Max Chipulu. Modelling of hampshire adult services–gearing up for future demands. *Health Care Manag Sci*, 11(2):167–176, Jun 2008.

[19] Mourtou Efstratia, Abdel-badeeh M. Salem, Abdel-badeeh M. Salem, and Pavlidis George. Modelling and analyzing a hospital procedure using a petri-net approach. In *Proceedings of World Academy of Science Engineering and Technology, 2007*, pages 98–102, 2007.

[20] D.M. Ferrin, M.J. Miller, S. Wininger, and M.S. Neuendorf. Analyzing incentives and scheduling in a major metropolitan hospital operating room through simulation. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 2, pages 1975–1980 vol.2, 2004.

[21] William J. Gordon and Gordon F. Newell. Closed queuing systems with exponential servers. *Operations Research*, 15(2):254–265, 1967.

[22] L. V. Green and V. Nguyen. Strategies for cutting hospital beds: the impact on patient service. *Health Serv Res*, 36(2):421–442, Jun 2001.

[23] Linda V. Green, Peter J. Kolesar, and Ward Whitt. Coping with time-varying demand when setting staffing requirements for a service system. *Production and Operations Management*, 16(1):13–39, 2007.

[24] Linda V. Green, Joo Soares, James F. Giglio, and Robert A. Green. Using queueing theory to increase the effectiveness of emergency department provider staffing. *Acad Emerg Med*, 13(1):61–68, Jan 2006.

[25] S. Groothuis, G. G. van Merode, and A. Hasman. Simulation as decision tool for capacity planning. *Comput Methods Programs Biomed*, 66(2-3):139–151, Sep 2001.

[26] Maria Gabriella Guglielmino, Giovanni Celano, Antonio Costa, and Sergio Fichera. Modelling a radiology department service using a vdl integrated approach. *J Health Organ Manag*, 23(4):376–395, 2009.

[27] Xudong He and Tadao Murata. High-level petri nets - extensions, analysis, and applications. In *The Electrical Engineering Handbook*, pages 459 – 475. Academic Press, Burlington, 2005.

[28] Frederick S. Hillier and Gerald J. Lieberman. *Introduction To Operations Research (IBM)*. Mcgraw-Hill College, 1995.

[29] Mama Hoard, Jack Homer, William Manley, Paul Furbee, Arshadul Haque, and James Helmkamp. Systems modeling in support of evidence-based disaster planning for rural areas. *Int J Hyg Environ Health*, 208(1-2):117–125, 2005.

[30] Robert V Hogg and Elliot A. Tanis. *Probability and Statistical Inference (7th Edition)*. Prentice Hall, 2005.

[31] James R. Jackson. Networks of waiting lines. *Operations Research*, 5(4):518–521, 1957.

[32] James R. Jackson. Jobshop like queueing systems. *Management Science*, 50(12):1796–1802, 2004.

[33] F. J. A. Jansen, L. F. P. Etman, J. E. Rooda, and I. J. B. F. Adan. Aggregate simulation modeling of an mri department using effective process times. In *Simulation Conference (WSC), Proceedings of the 2012 Winter*, pages 1–12, 2012.

[34] M.H. Jansen-Vullers and H.A. Reijers. Business process redesign at a mental healthcare institute: a coloured petri net approach. In *Sixth Workshop on Practical Use of Coloured Petri Nets and the CPN Tools, Aarhus, Denmark*, pages 21–38, 2005.

[35] Kurt Jensen and Lars M. Kristensen. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer, 2009.

[36] MJ Johnston, P Samaranayake, A Dadich, and JA Fitzgerald. Modelling radiology department operation using discrete event simulation. In *MODSIM, International Congress on Modelling and Simulation*, pages 678–684, 2009.

[37] Catherine M. Joyce, John J. McNeil, and Johannes U. Stoelwinder. More doctors, but not enough: Australian medical workforce supply 2001-2012. *Med J Aust*, 184(9):441–446, May 2006.

[38] J. B. Jun, S. H. Jacobson, and J. R. Swisher. Application of discrete-event simulation in health care clinics: A survey. *The Journal of the Operational Research Society*, 50(2):pp. 109–123, 1999.

[39] Naoru Koizumi, Eri Kuno, and Tony E Smith. Modeling patient flows using a queuing network with blocking. *Health Care Management Science*, 8(1):49 – 60, 2005.

[40] A Kolker. Queuing analytic theory and discrete events simulation for healthcare: Right application for the right problem. *Childrens Hospital of Wisconsin*, 2008.

[41] Charles Lakos. Object oriented modelling with object petri nets. In GulA. Agha, Fiorella Cindio, and Grzegorz Rozenberg, editors, *Concurrent Object-Oriented Programming and Petri Nets*, volume 2001 of *Lecture Notes in Computer Science*, pages 1–37. Springer Berlin Heidelberg, 2001.

[42] C.R.M. Leite, D.L. Martin, G.R.M.A. Sizilio, K.E.A. dos Santos, B.G. de Araujo, R.A. de M Valentim, A. D D Neto, J.D. De Melo, and A. M G Guerreiro. Modeling of medical care with stochastic petri nets. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 1336–1339, 2010.

[43] Fatma Poni Mardiah and Mursyid Hasan Basri. The Analysis of Appointment System to Reduce Outpatient Waiting Time at Indonesias Public Hospital. *Human Resource Management Research*, 3(1):27–33, 2013.

[44] Brian J Masterson, Thomas G Mihara, George Miller, Stephen C Randolph, M Emma Forkner, and Andrew L Crouter. Using models and data to support optimization of the military health system: a case study in an intensive care unit. *Health Care Management Science*, 7(3):217 – 224, 2004.

[45] Michael L. McManus, Michael C. Long, Abbot Cooper, and Eugene Litvak. Queuing theory accurately models the need for critical care resources. *Anesthesiology*, 100(5):1271–1276, May 2004.

[46] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[47] Carl A Petri. Communication with automata: Volume 1 supplement 1. Technical report, DTIC Document, 1966.

[48] S. Ramakrishnan, K. Nagarkar, M. DeGennaro, K. Srihari, A.K. Courtney, and F. Emick. A study of the ct scan area of a healthcare provider. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 2, pages 2025–2031 vol.2, 2004.

[49] M. Reiser and S. S. Lavenberg. Mean-value analysis of closed multichain queuing networks. *J. ACM*, 27(2):313–322, 1980.

[50] S. Samaha, W.S. Armel, and D.W. Starks. The use of simulation to reduce the length of stay in an emergency department. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, volume 2, pages 1907–1911 vol.2, 2003.

[51] Claudia A Sanmartin. Toward standard definitions for waiting times. In *Healthcare Management Forum*, volume 16, pages 49–53. Elsevier, 2003.

[52] S. Takakuwa and H. Shiozaki. Functional analysis for operating emergency department of a general hospital. In *Simulation Conference, 2004. Proceedings of the 2004 Winter*, volume 2, pages 2003–2011 vol.2, 2004.

[53] E. Tavares, J. Aleixo, P. Maciel, D. Oliveira, E. Heyde, R. Araujo, L. Maia, A. Duarte, and M. Novaes. Performance evaluation of medical imaging service. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 1349–1354, New York, NY, USA, 2012. ACM.

[54] Pieter Toussaint, Margaret Williamson, Geoff McDonnell, and Oystein Nytro. Modelling the communication dimension of clinical work and medical errors. In *24th International Conference on Systems Dynamics, The System Dynamics SOCiety, Nijmegen, The Netherlands*, 2006.

[55] Sonya Adriana Vanderby. *An investigation into the suitability of applying system dynamics to patient flow modeling*. PhD thesis, University of Toronto, 2009.

[56] Shreyas S. Vasanawala and Terry S. Desser. Accommodation of requests for emergency us and ct: applications of queueing theory to scheduling of urgent studies. *Radiology*, 235(1):244–249, Apr 2005.

[57] Francis de Vricourt and Otis B. Jennings. Nurse staffing in medical units: A queueing perspective. *Operations Research*, 59(6):1320–1331, 2011.

[58] S.-J. Whittaker, K. Rudie, J. McLellan, and S. Haar. Choice-point nets: A discrete-event modelling technique for analyzing health care protocols. In *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pages 652–659, 2009.

[59] A. Wiinamaki and R. Dronzek. Using simulation in the architectural concept phase of an emergency department design. In *Simulation Conference, 2003. Proceedings of the 2003 Winter*, volume 2, pages 1912–1916 vol.2, 2003.

[60] Frank Windmeijer, Hugh Gravelle, and Pierre Hoonhout. Waiting lists, waiting times and admissions: an empirical analysis at hospital and general practice level. *Health Econ*, 14(9):971–985, Sep 2005.

[61] Eric Wolstenholme, David Monk, Douglas McKelvie, and Steve Arnold. Coping but not coping in health and social care: masking the reality of running organisations beyond safe design capacity. *System Dynamics Review*, 23(4):371–389, 2007.

[62] David Worthington. Hospital waiting list management models. *The Journal of the Operational Research Society*, 42(10):pp. 833–843, 1991.

[63] Huanxin Henry Xiong, MengChu Zhou, and C.N. Manikopoulos. Modeling and performance analysis of medical services systems using petri nets. In *Systems, Man, and Cybernetics, 1994. Humans, Information and Technology., 1994 IEEE International Conference on*, volume 3, pages 2339–2342 vol.3, 1994.

[64] Natalia Yankovic and Linda V. Green. Identifying good nursing levels: A queuing approach. *Operations Research*, 59(4):942–955, 2011.

[65] W.J. Zhang, Y. Lin, and Niraj. Sinha. On the function-behavior-structure model for design. In *The 2nd CDEN Conference, Alberta, Canada*, July 2005. CD ROM.

Appendix A

The PN Model in CPN Tools

CPN Tools is created and maintained by the CPN Group at Aarhus University from 2000 to 2010 and then transferred to the AIS group, Eindhoven University of Technology, The Netherlands (http://cpntools.org/). It is free software and can be downloaded from the website.

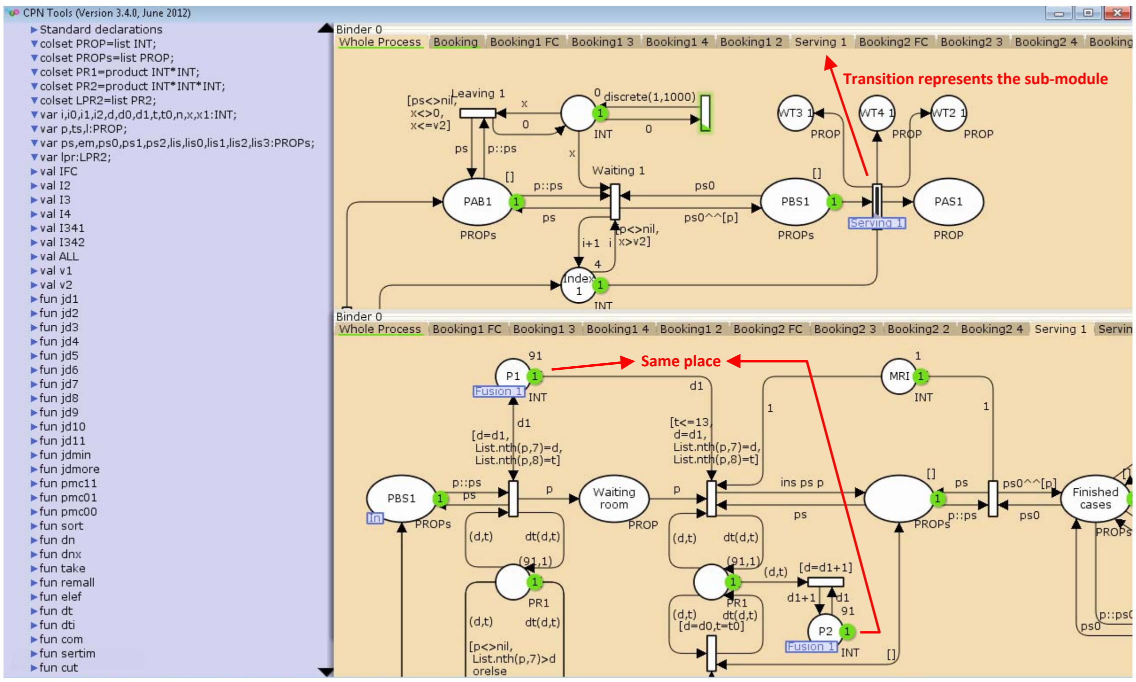

Figure A.1: The screen of CPN Tools

Figure A.1 is the screenshot of CPN Tools. The left part is the ML codes and the right part shows two nets. There are two special nodes in the nets, which are explained as follows.

- Substitution transitions: represented by double-layered rectangles and act as substitutions of the corresponding sub-modules. For example, in the upper right net of Figure A.1, the substitution transition “Serving 1” represents the sub-module “Serving 1”.
- Fusion of places: the same places located in different position (in the same net or different nets). For example, in the lower right net of figure A.1, the places “P1” and “P2” are the same place.

The ML codes and all the nets of the model are displayed as follows.

ML codes for the Petri net model:

```
colset PROP=list INT;

colset PROPs=list PROP;

colset PR1=product INT*INT;

colset PR2=product INT*INT*INT;

colset LPR2=list PR2;

var i,i0,i1,i2,d,d0,d1,t,t0,n,x,x1:INT;

var p,ts,l:PROP;

var ps,em,ps0,ps1,ps2,lis,lis0,lis1,lis2,lis3:PROPs;

var lpr:LPR2;

val v1=10;

val v2=45;

fun jd1 p=
   if (List.nth(p,6)=0
       orelse
       List.nth(p,6)=1)
   then true
   else false;

fun jd2 p=
   if (List.nth(p,6)=0
       orelse
       List.nth(p,6)=2)
   then true
   else false;

fun jd3 (p,[],[],[])=false
  | jd3 (p,lis1,lis2,lis3)=
      if (List.nth(p,1)=1
          andalso
          lis1=nil)
      then true
      else
         if (List.nth(p,1)=0
             andalso
             List.nth(p,2)=1
             andalso
             lis2=nil)
         then true
         else
            if (List.nth(p,1)=0
```

```
                andalso
                List.nth(p,2)=0
                andalso
                lis3=nil)
            then true
            else false;

fun jd4 (ps,lis1,lis2,lis3)=
   if (lis1=nil andalso
       lis2=nil andalso
       lis3=nil)
   then true
   else
      if ps=nil
      then true
      else false;

fun jd5 (ps,d,t,i)=
   if (i<>0
      andalso d<>0
      andalso t<>0)
   then
      if ps=nil
      then true
      else
         if (List.nth(hd ps,7)>d
             orelse
             List.nth(hd ps,8)>t)
         then true
         else false
   else false;

fun jd6 (p,ps)=
   if ps=nil
   then false
   else
      if List.nth(p,4)>List.nth(hd ps,4)
      then true
      else jd6 (p,tl ps);

fun jd7 (lis1,lis2,lis3)=
   if (lis1=nil andalso lis2=nil andalso lis3=nil)
   then false
   else true;

fun jd8 (l,n)=
   if List.nth(l,n) mod 7=6
   then false
   else
      if List.nth(l,n) mod 7=0
      then false
      else true;

fun jd9 d=
```

```
   if d mod 7=6
   then false
   else
      if d mod 7=0
      then false
      else true;

fun jd10 (d1,x)=
   if jd9 d1
   then
      if x>630
      then true
      else false
   else
      if x>300
      then true
      else false;

fun jd11 (d1,x)=
   if jd9 d1
   then
      if x<=630
      then true
      else false
   else
      if x<=300
      then true
      else false;

fun jdmin (d1,x1,minweekday,minweekend)=
   if jd9 d1
   then
      if x1=minweekday
      then true
      else false
   else
      if x1=minweekend
      then true
      else false;

fun jdmore (d1,x1,minweekday,minweekend)=
   if jd9 d1
   then
      if x1>minweekday
      then true
      else false
   else
      if x1>minweekend
      then true
      else false;

fun pmc11 (l,n)=
   if jd8 (l,n)
   then
```

```
      if mem [1,2,3,4,5] (List.nth(l,1))
      then true
      else false
   else
      false;

fun pmc01 (l,n)=
   if jd8 (l,n)
   then
      if mem [6,7,8] (List.nth(l,1))
      then true
      else false
   else
      if mem [1,2,3] (List.nth(l,1))
      then true
      else false;

fun pmc00 (l,n)=
   if jd8 (l,n)
   then
      if mem [9,10,11,12,13] (List.nth(l,1))
      then true
      else false
   else
      if mem [4,5,6,7] (List.nth(l,1))
      then true
      else false;

fun sort (p,ps,i)=
   if i=0
   then
      if List.nth(p,4)>List.nth(hd ps,4)
      then [hd ps]^^[p]^^tl ps
      else sort (p,ps,i+1)
   else
      if List.nth(p,4)>List.nth(List.nth(ps,i),4)
      then [List.nth(ps,i)]^^[p]^^List.take(ps,i)^^List.drop(ps,i+1)
      else sort (p,ps,i+1);

fun dn (d,1)=1`(d,0)
  | dn (d,m)=1`(d,0)++dn (d+1,m-1);

fun dnx (d,1)=1`(d,0,0)
  | dnx (d,m)=1`(d,0,0)++dnx (d+1,m-1);

fun take x=
   let
   val g=length x
   in
   List.take(x,g)
   end;

fun remall (l::lis,d)=
      if d=hd l
```

```
      then remall (lis,d)
      else l::remall (lis,d)
  | remall ([],d)=[];

fun elef (x,i)=
   let
   val g=hd x
   in
   List.nth(g,i)
   end;

fun dt (d,t)=
   if t<13
   then (d,t+1)
   else (d+1,1);

fun dti (d,t,i)=
   if jd9 d
   then
      if t<13
      then (d,t+1,i)
      else (d+1,1,i)
   else
      if t<7
      then (d,t+1,i)
      else (d+1,1,i);

fun com (x0::x,y0::y,i,j)=
   if List.nth(x0,i)=List.nth(y0,i)
   then
      if List.nth(x0,j)>List.nth(y0,j)
      then com ([y0]^^[x0]^^x,y,i,j)
      else x0::com (x,y0::y,i,j)
   else
      if List.nth(x0,i)<List.nth(y0,i)
      then x0::(com (x,y0::y,i,j))
      else com (([y0]^^[x0]^^x),y,i,j)
  | com (x,[],i,j)=x
  | com ([],y,i,j)=y;

fun sertim p=
   if jd8 (p,7)
   then
      if mem [1,2,3,4,5,6,7,8] (List.nth(p,8))
      then discrete(50,70)
      else discrete(20,40)
   else
      if mem [1,2,3] (List.nth(p,8))
      then discrete(50,70)
      else discrete(20,40);

fun cut []=[]
  | cut ps=
   [List.take(hd(ps),6)]^^(cut (tl ps));
```

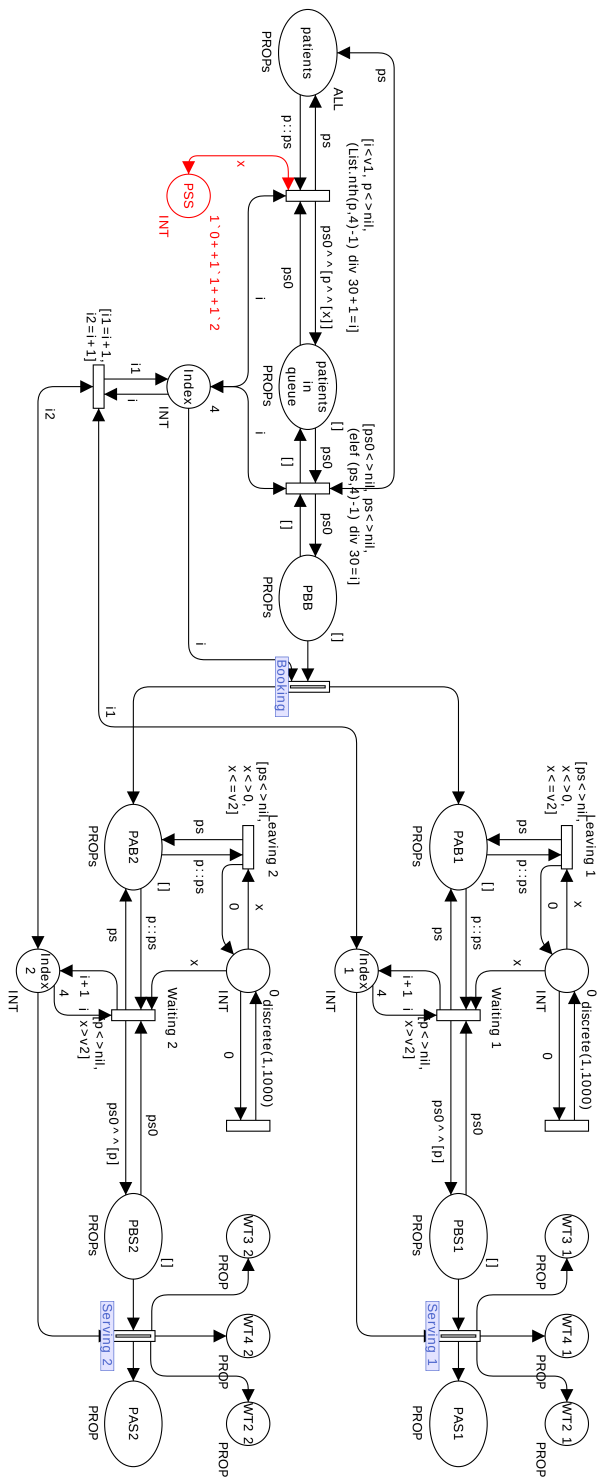

Figure A.2: The PN model of the whole process in CPN Tools

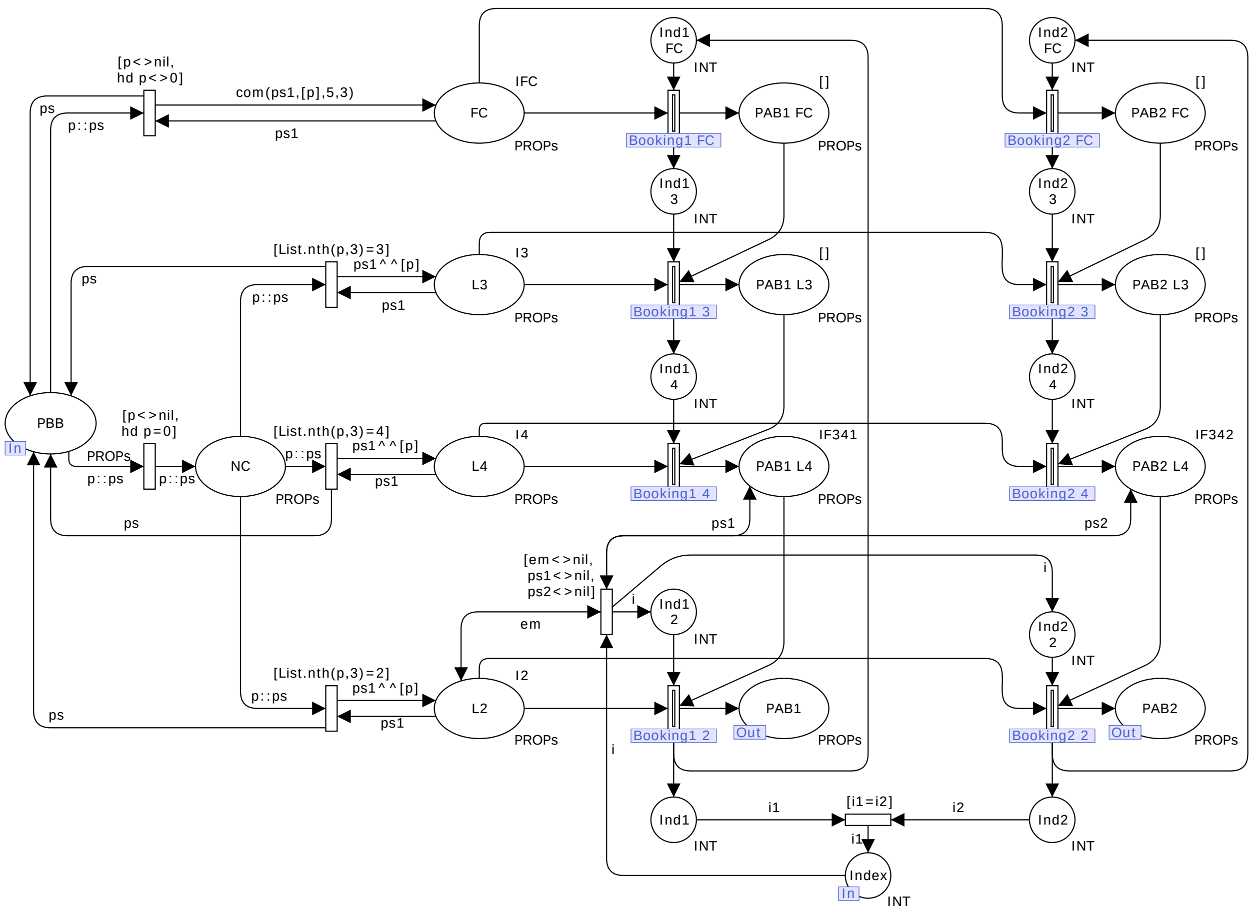

Figure A.3: The PN model of booking process in CPN Tools

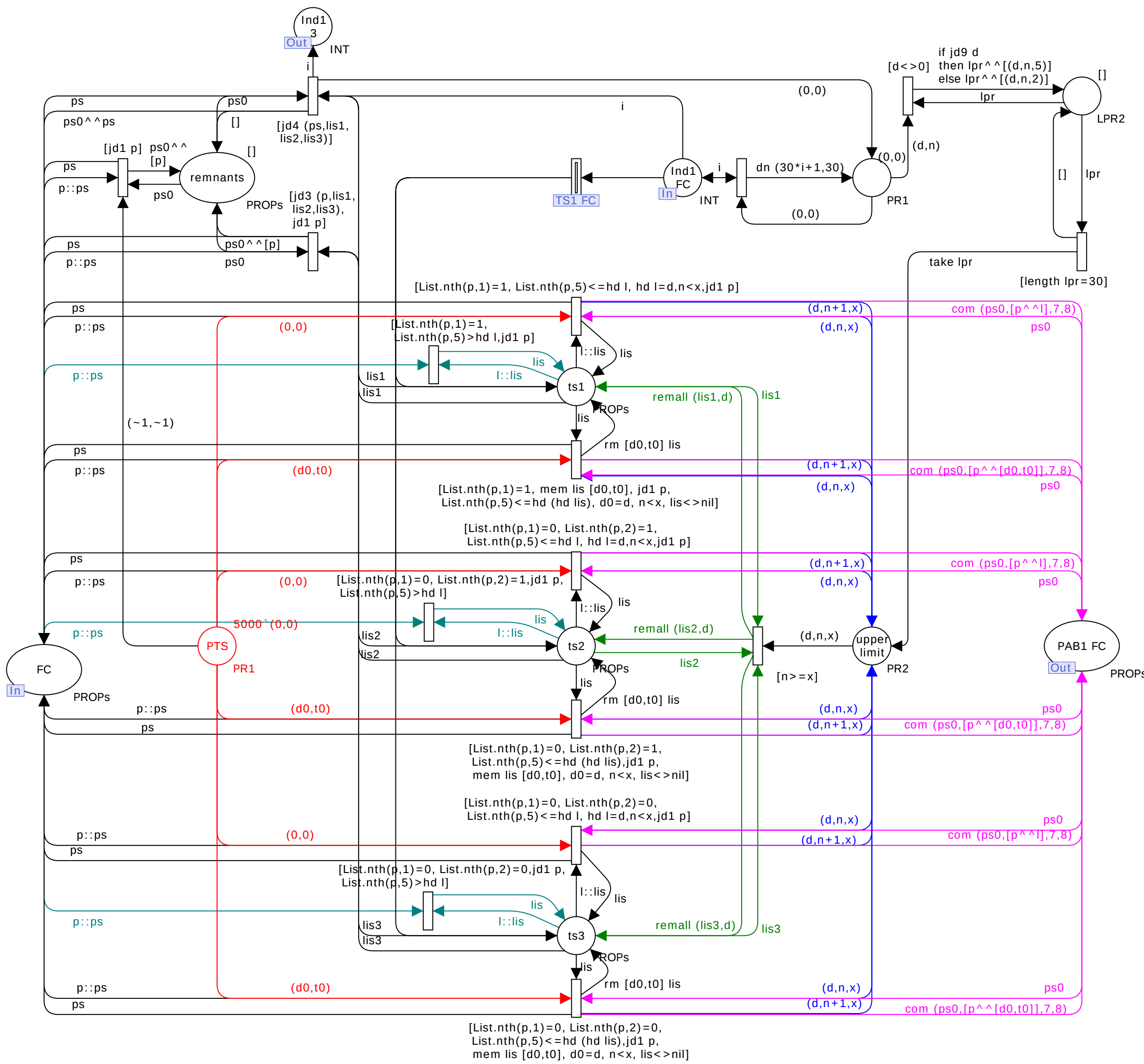

Figure A.4: The PN model of the booking of future cases for MRI_1 in CPN Tools

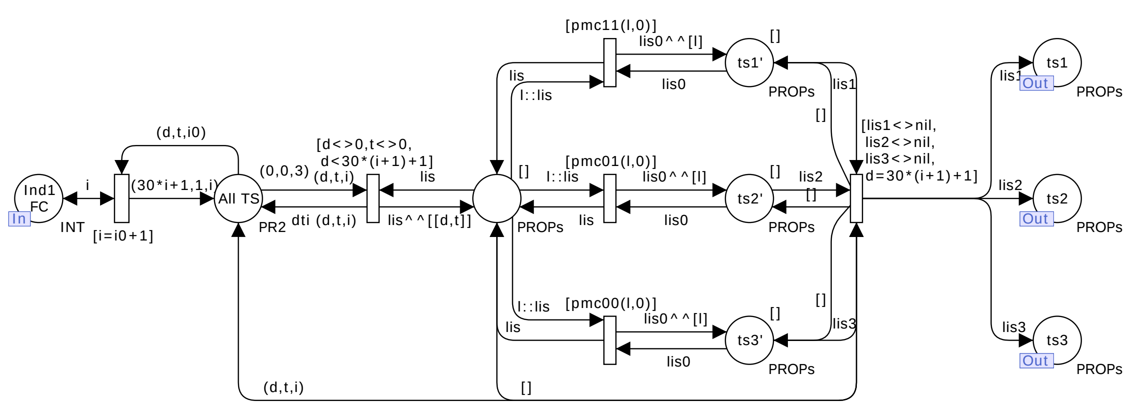

Figure A.5: The PN model of the time slot sorting of future cases for MRI_1 in CPN Tools

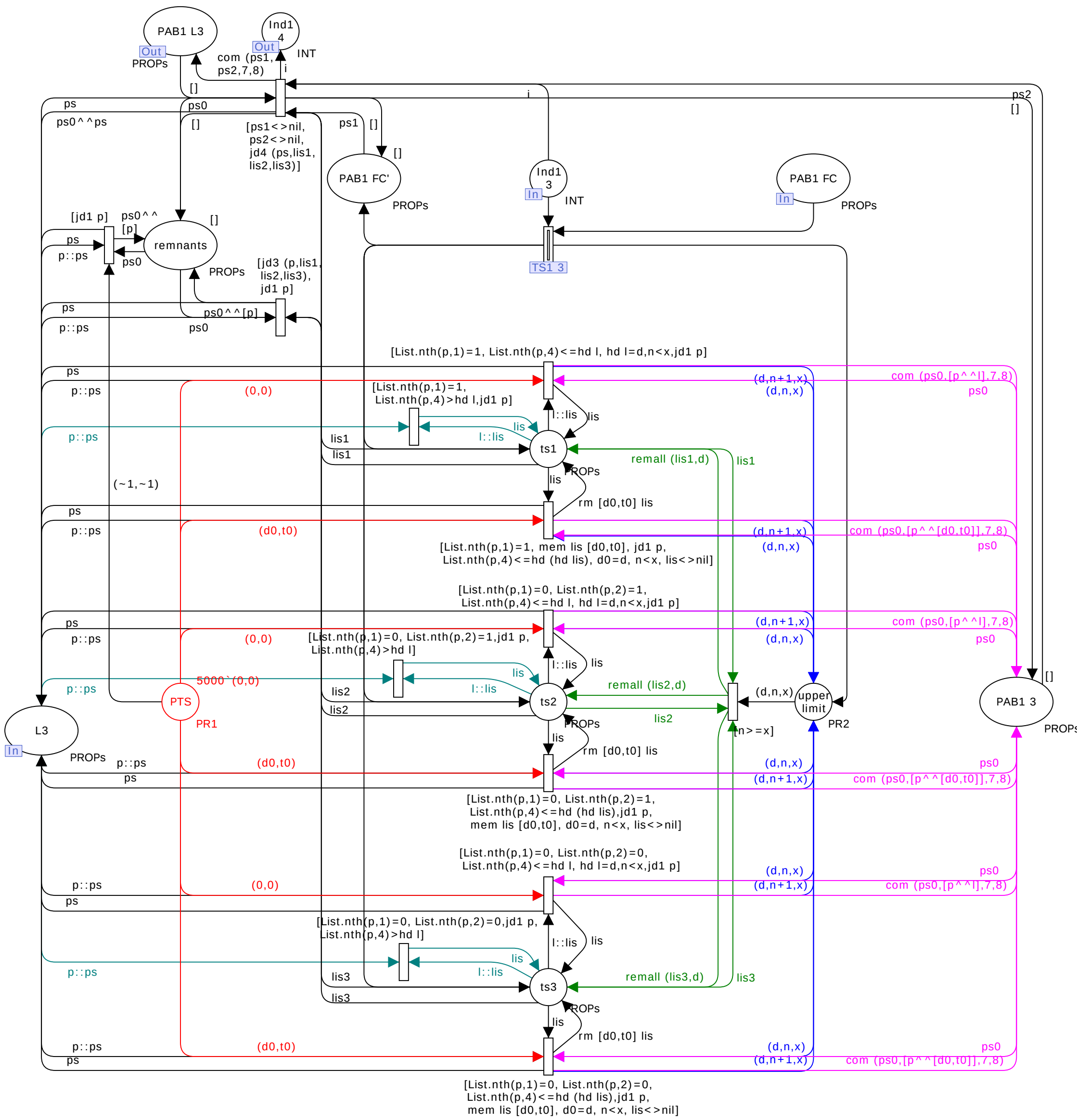

Figure A.6: The PN model of the booking of level 3 cases for MRI_1 in CPN Tools

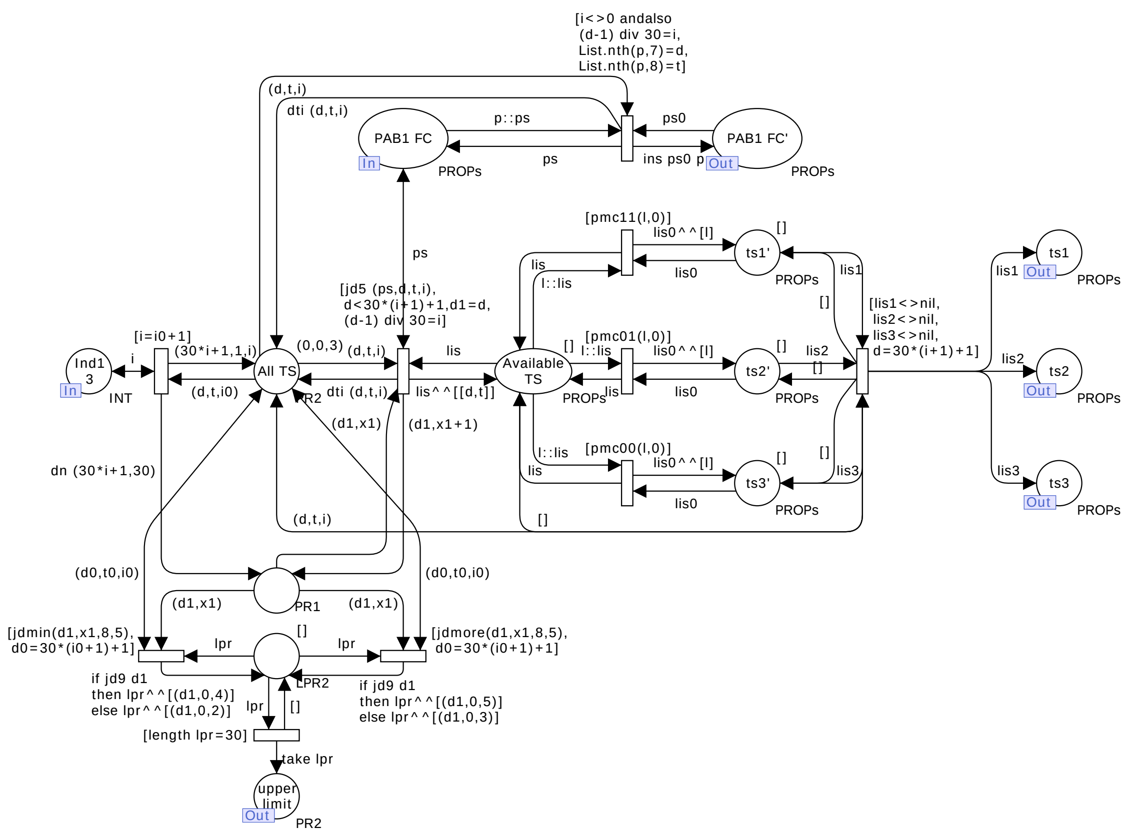

Figure A.7: The PN model of the time slot sorting of level 3 cases for MRI_1 in CPN Tools

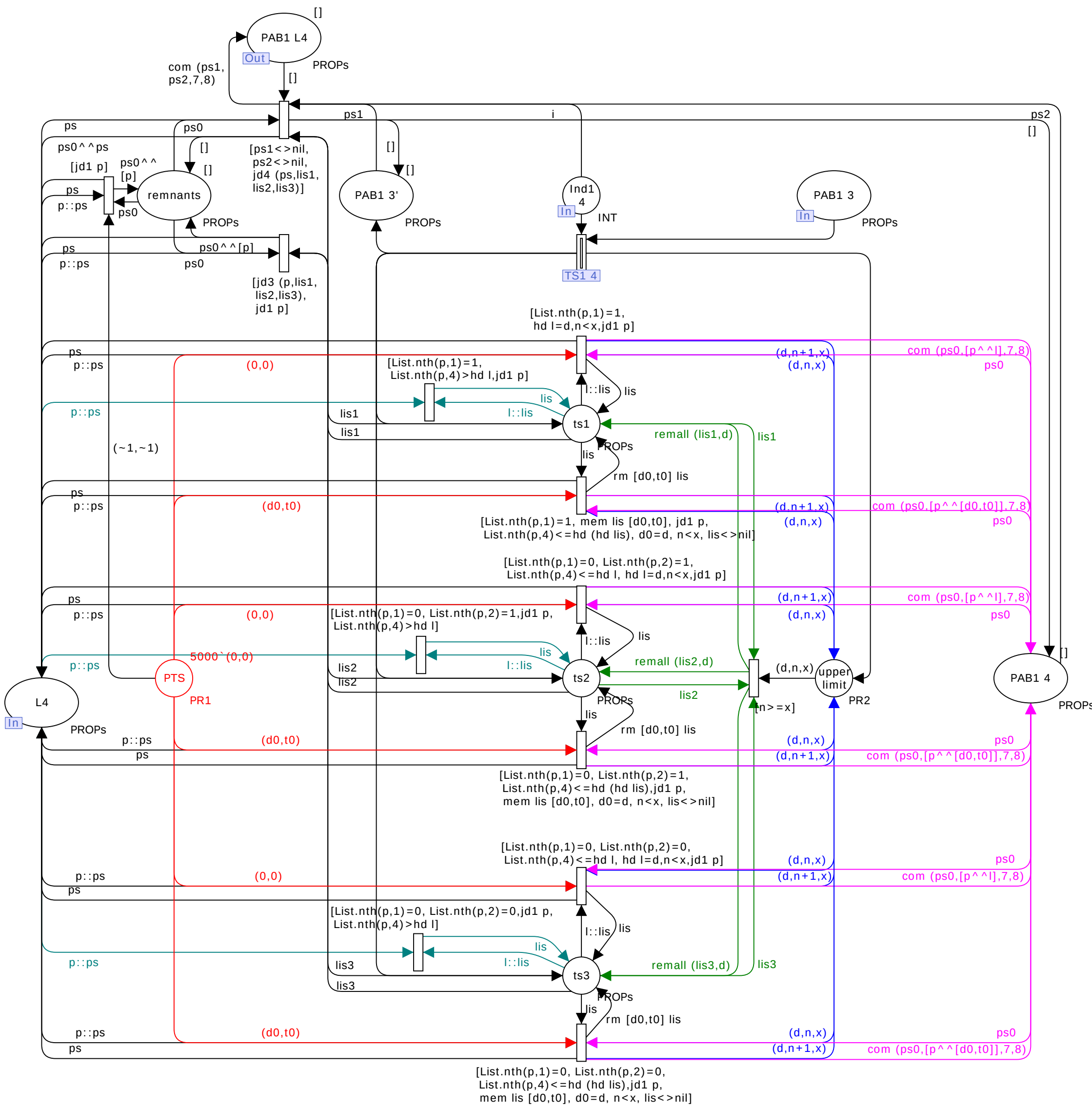

Figure A.8: The PN model of the booking of level 4 cases for MRI_1 in CPN Tools

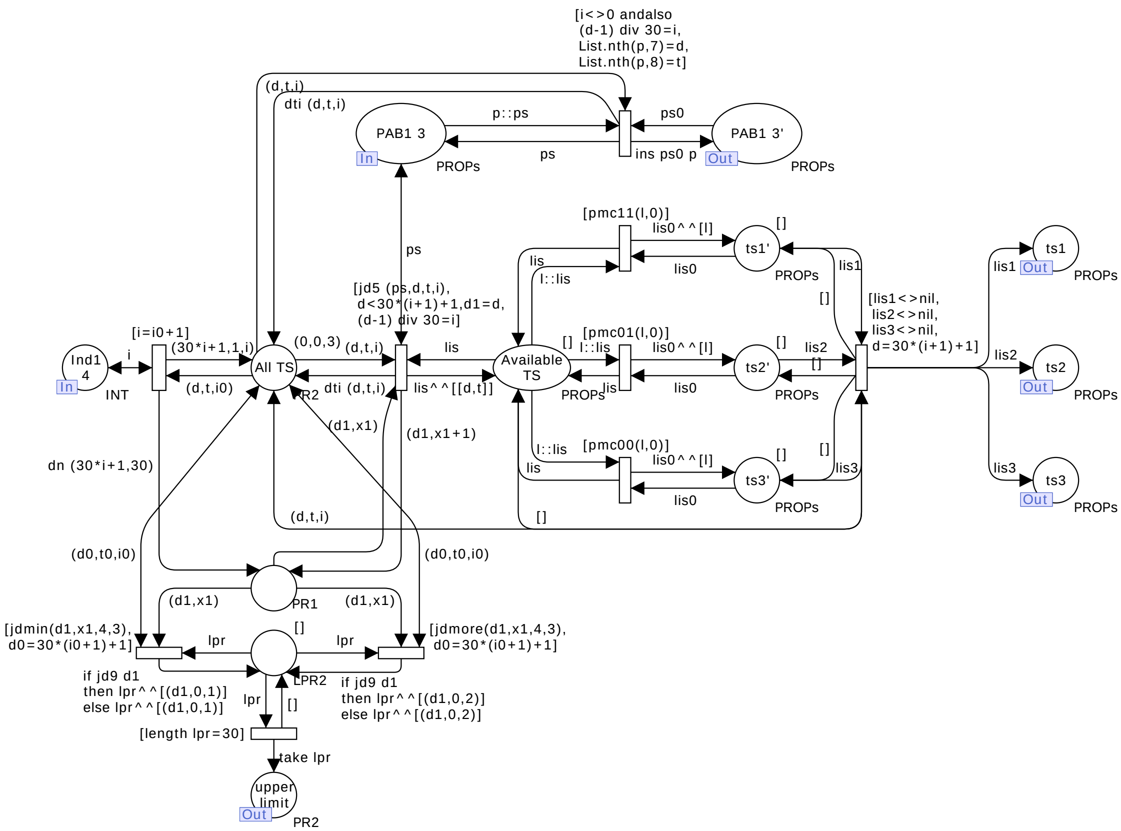

Figure A.9: The PN model of the time slot sorting of level 4 cases for MRI_1 in CPN Tools

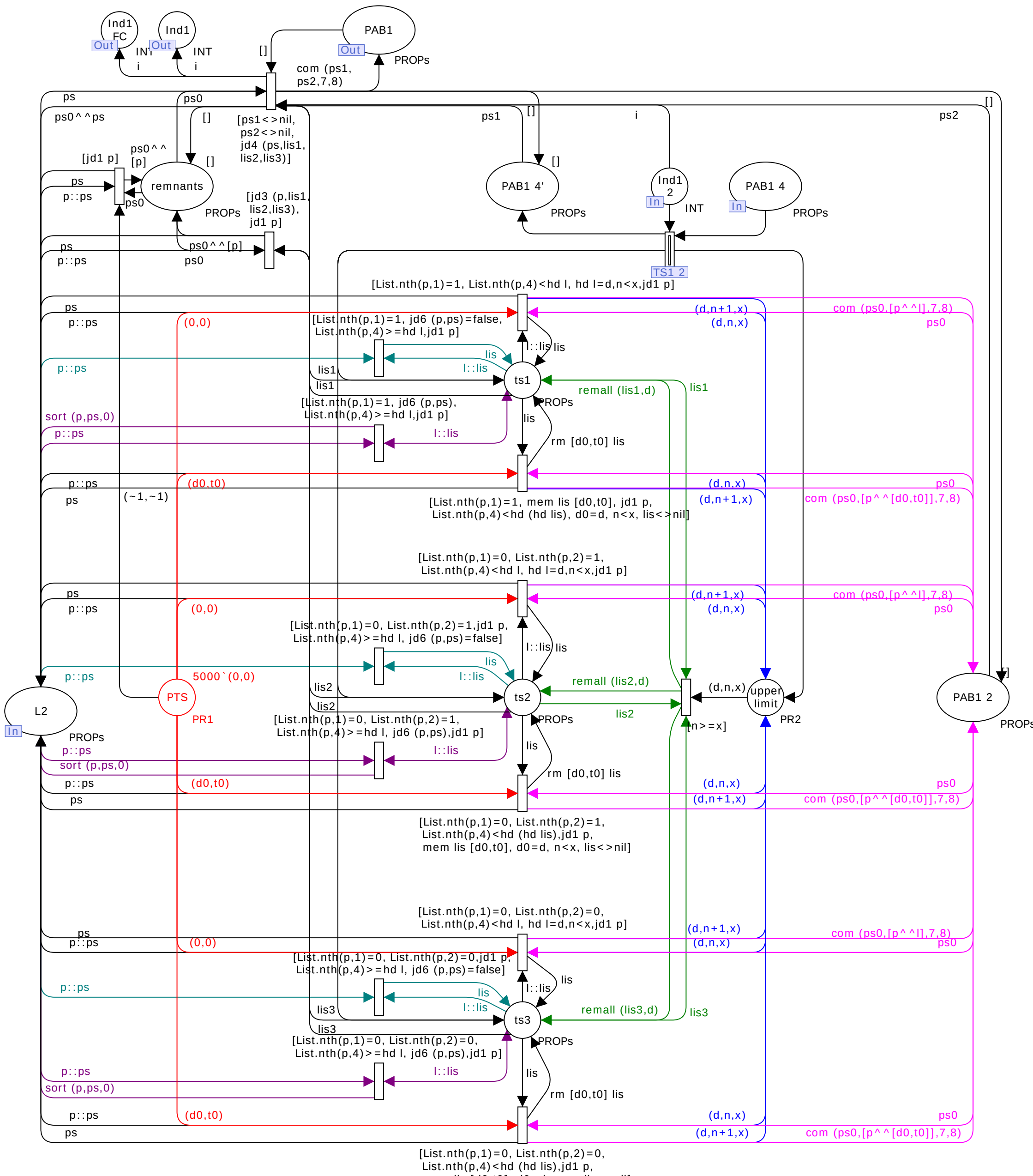

Figure A.10: The PN model of the booking of level 2 cases for MRI_1 in CPN Tools

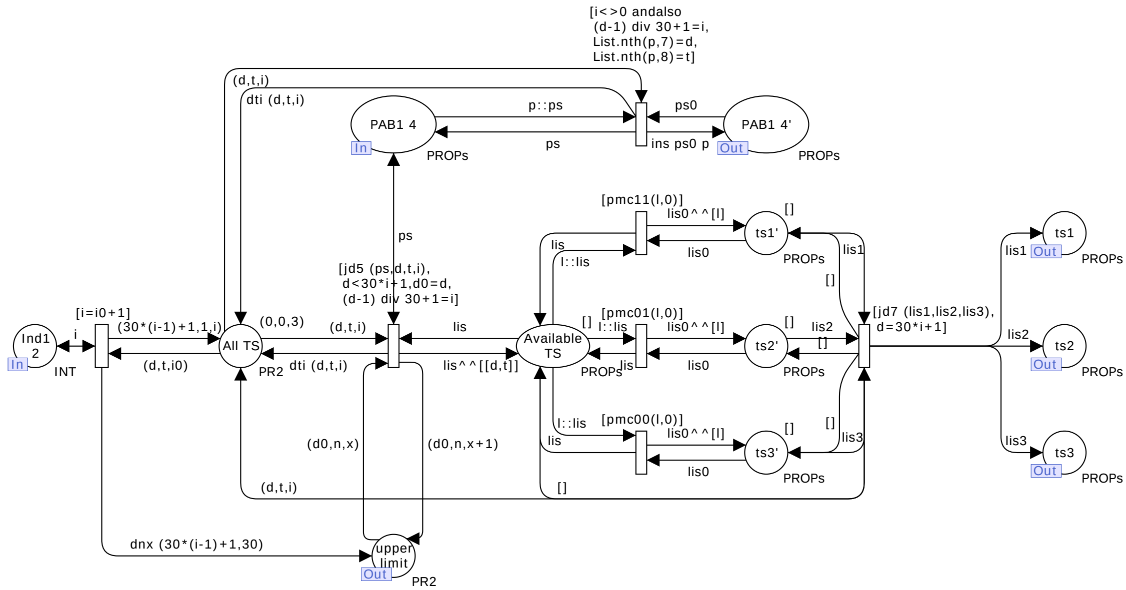

Figure A.11: The PN model of the time slot sorting of level 2 cases for MRI_1 in CPN Tools

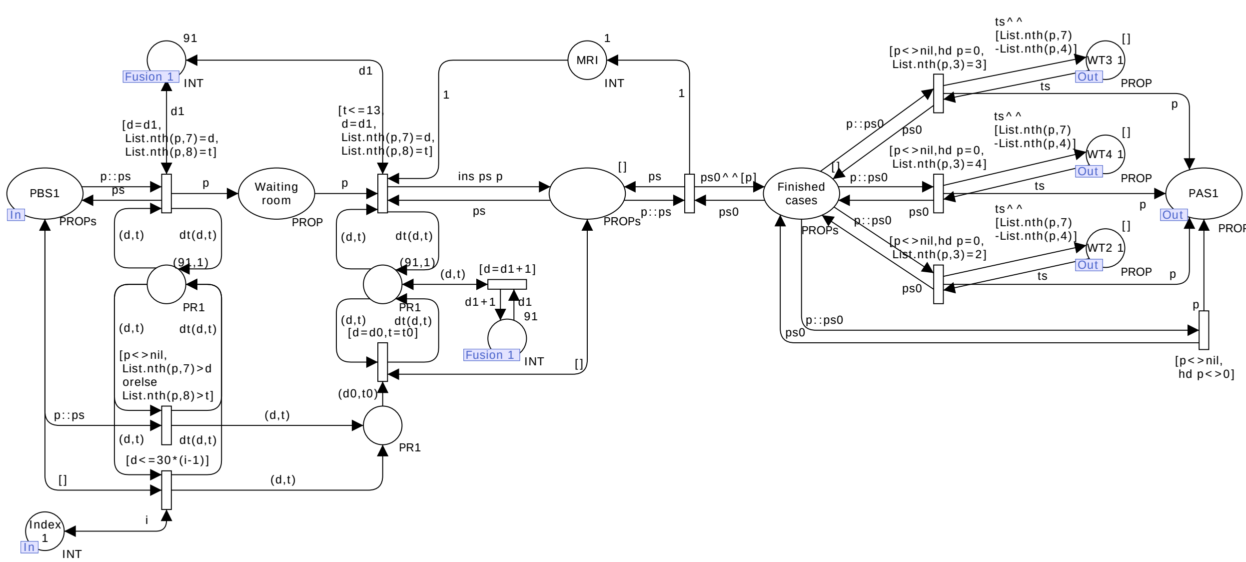

Figure A.12: The PN model of serving process for MRI_1 in CPN Tools

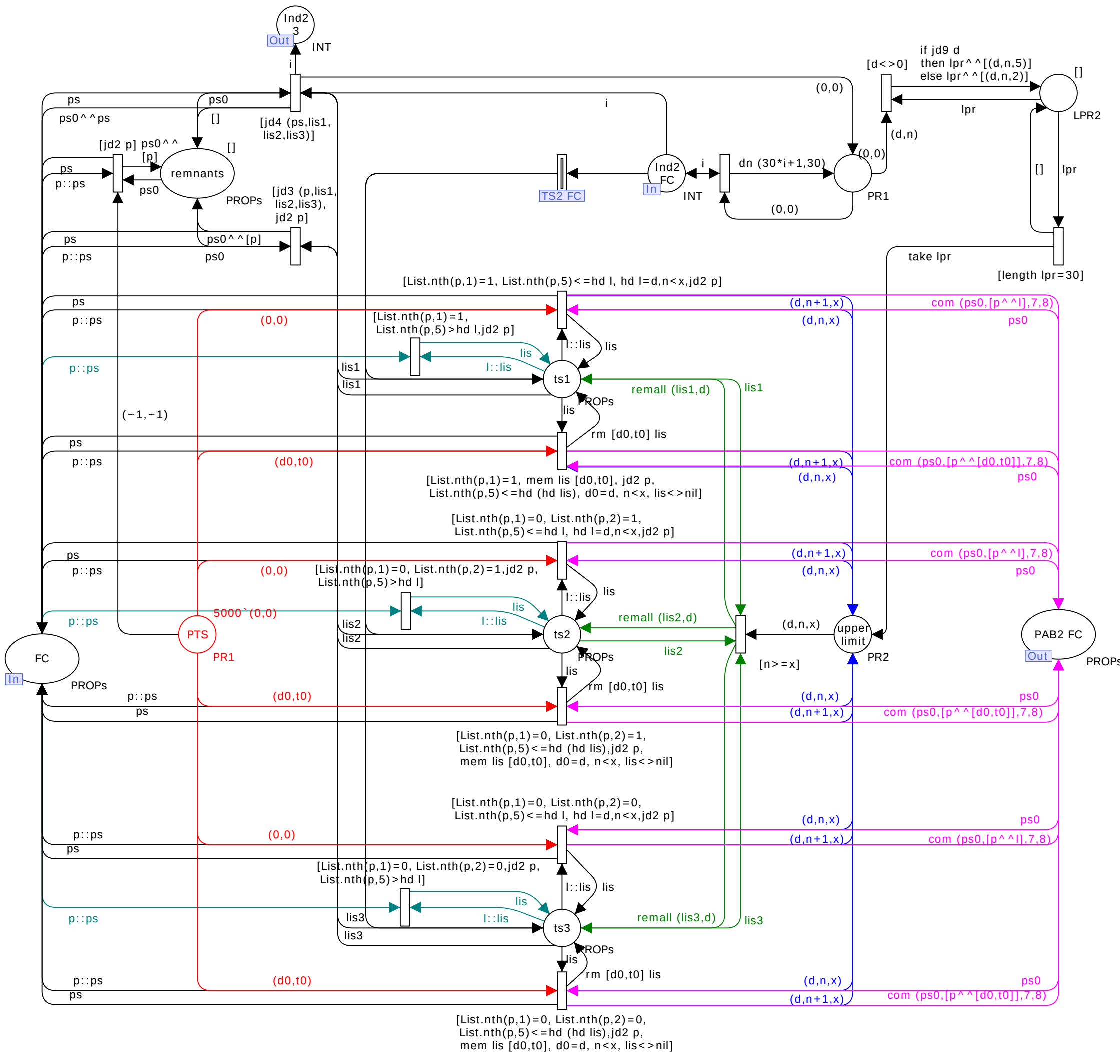

Figure A.13: The PN model of the booking of future cases for MRI_2 in CPN Tools

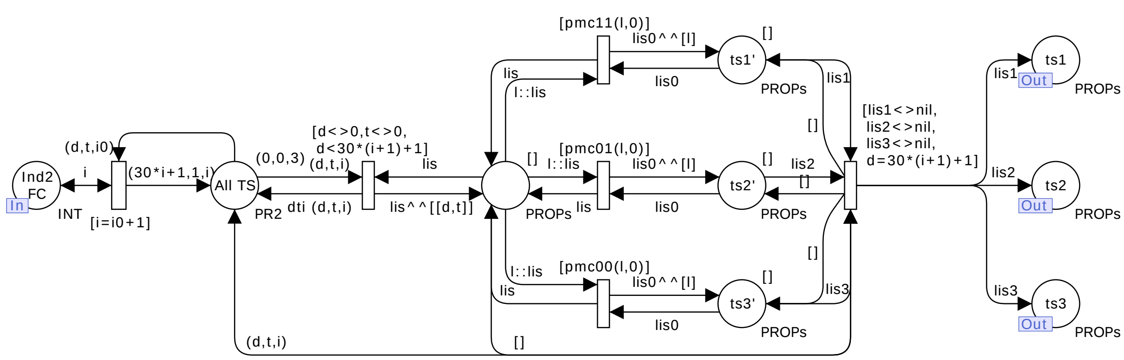

Figure A.14: The PN model of the time slot sorting of future cases for MRI_2 in CPN Tools

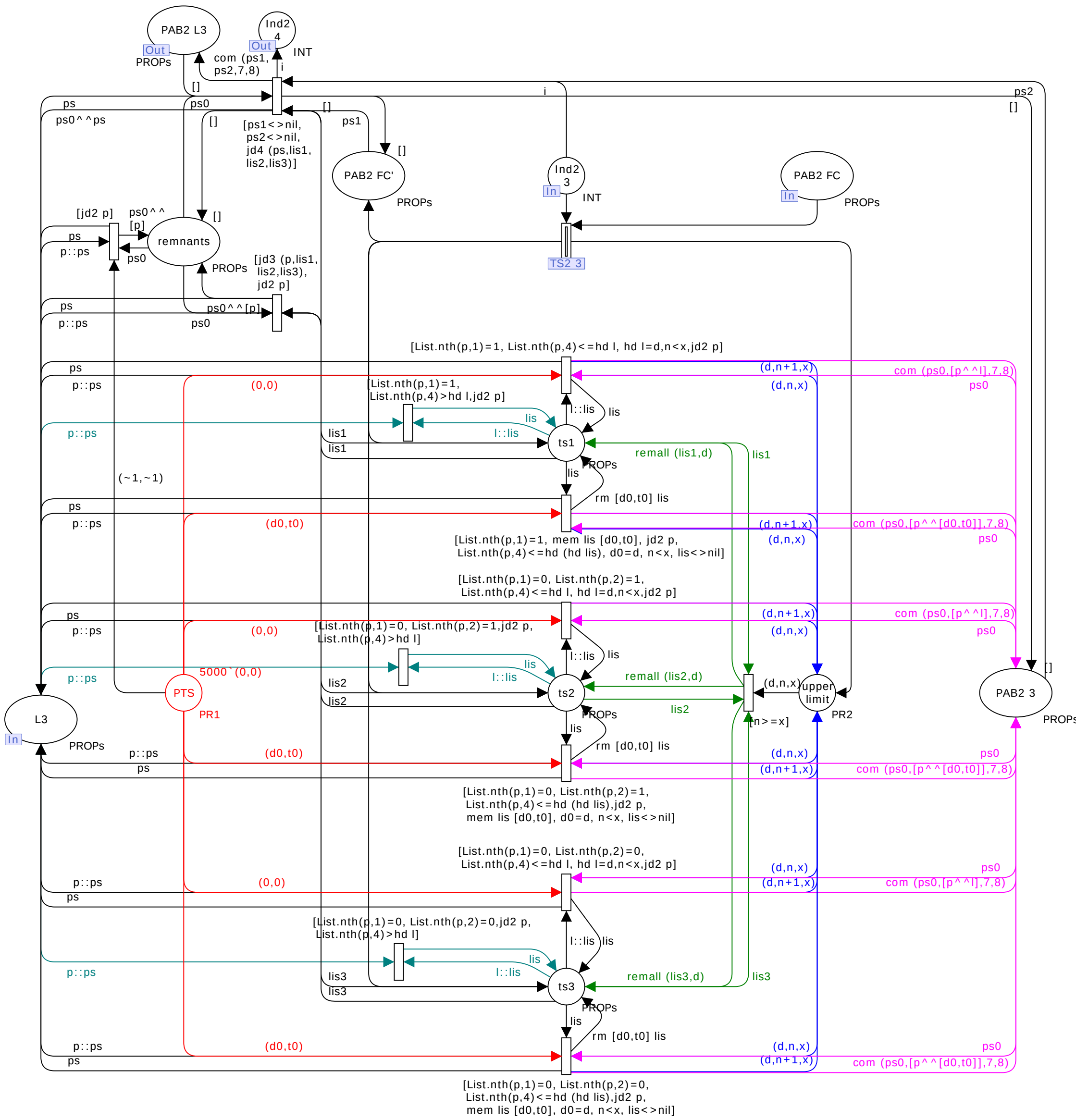

Figure A.15: The PN model of the booking of level 3 cases for MRI_2 in CPN Tools

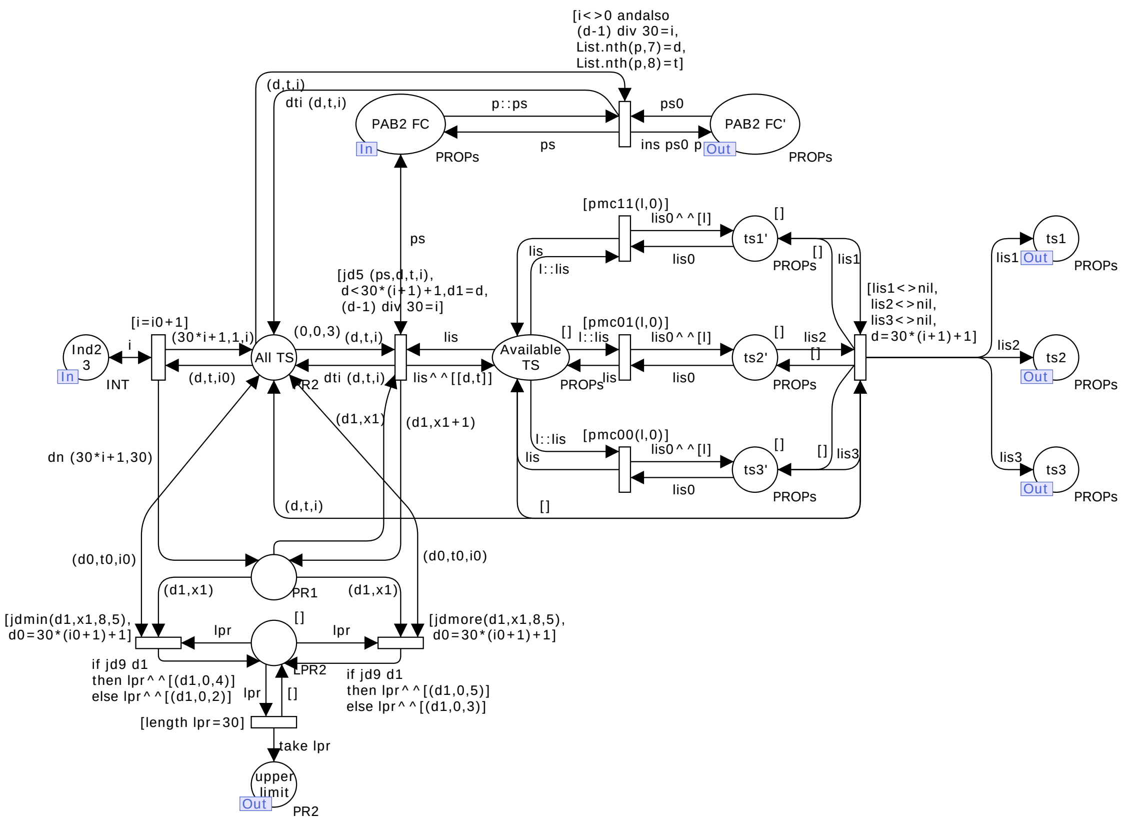

Figure A.16: The PN model of the time slot sorting of level 3 cases for MRI_2 in CPN Tools

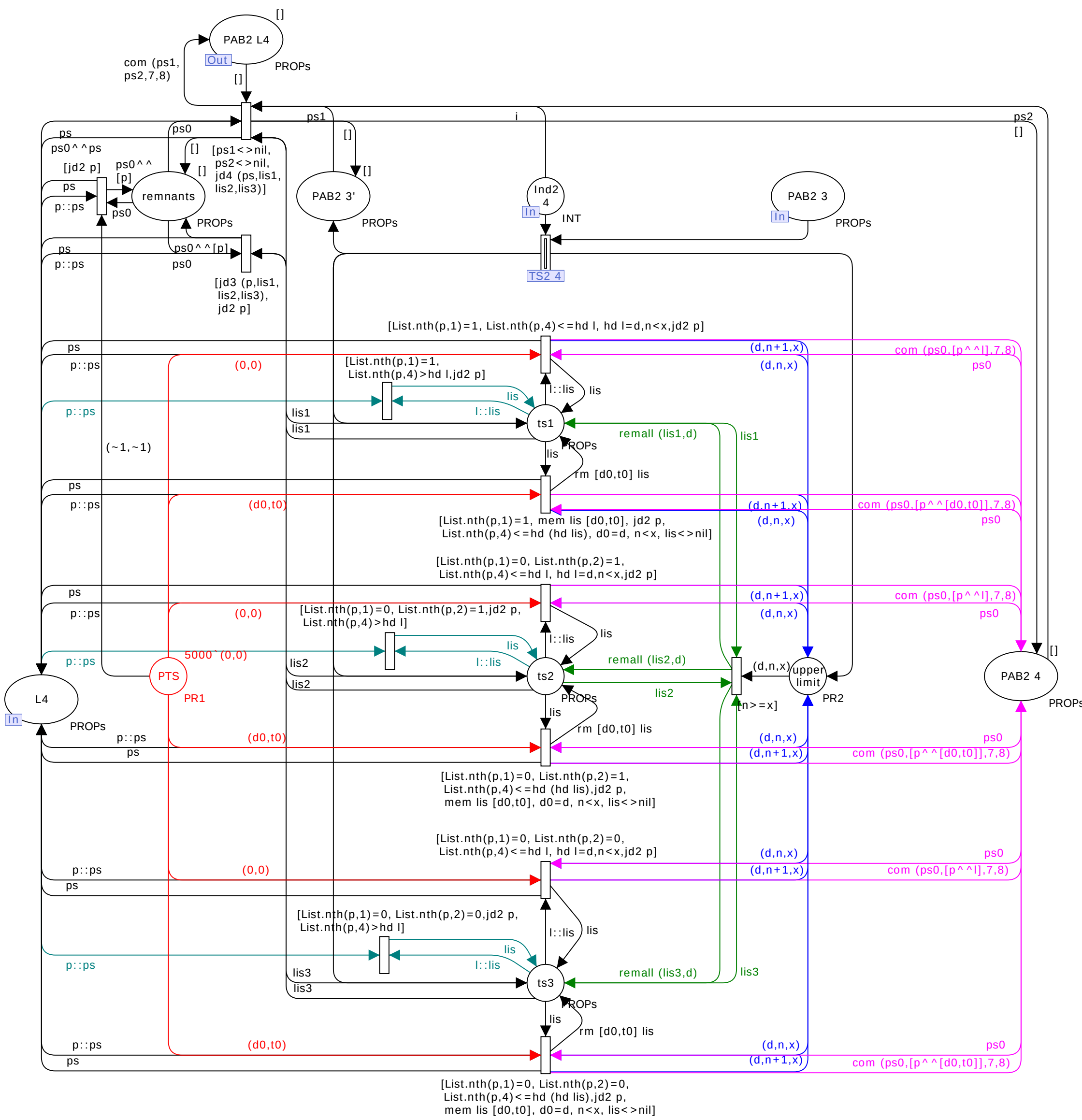

Figure A.17: The PN model of the booking of level 4 cases for MRI_2 in CPN Tools

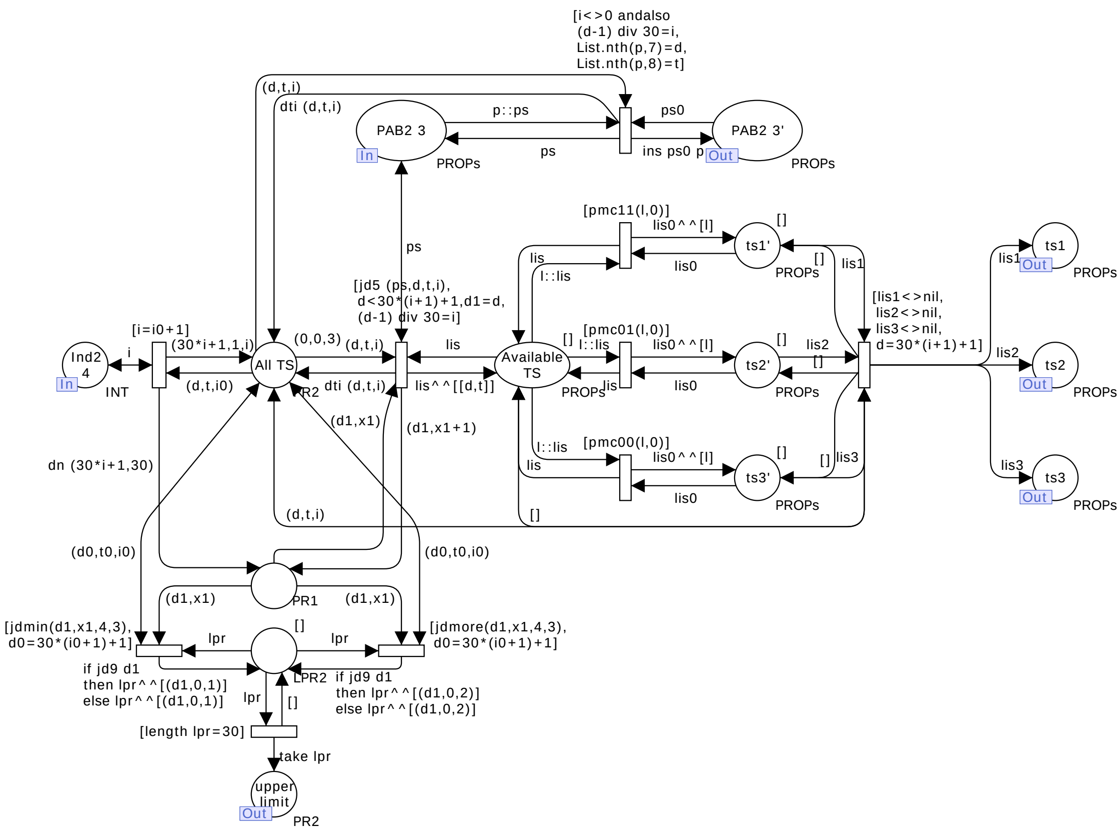

Figure A.18: The PN model of the time slot sorting of level 4 cases for MRI_2 in CPN Tools

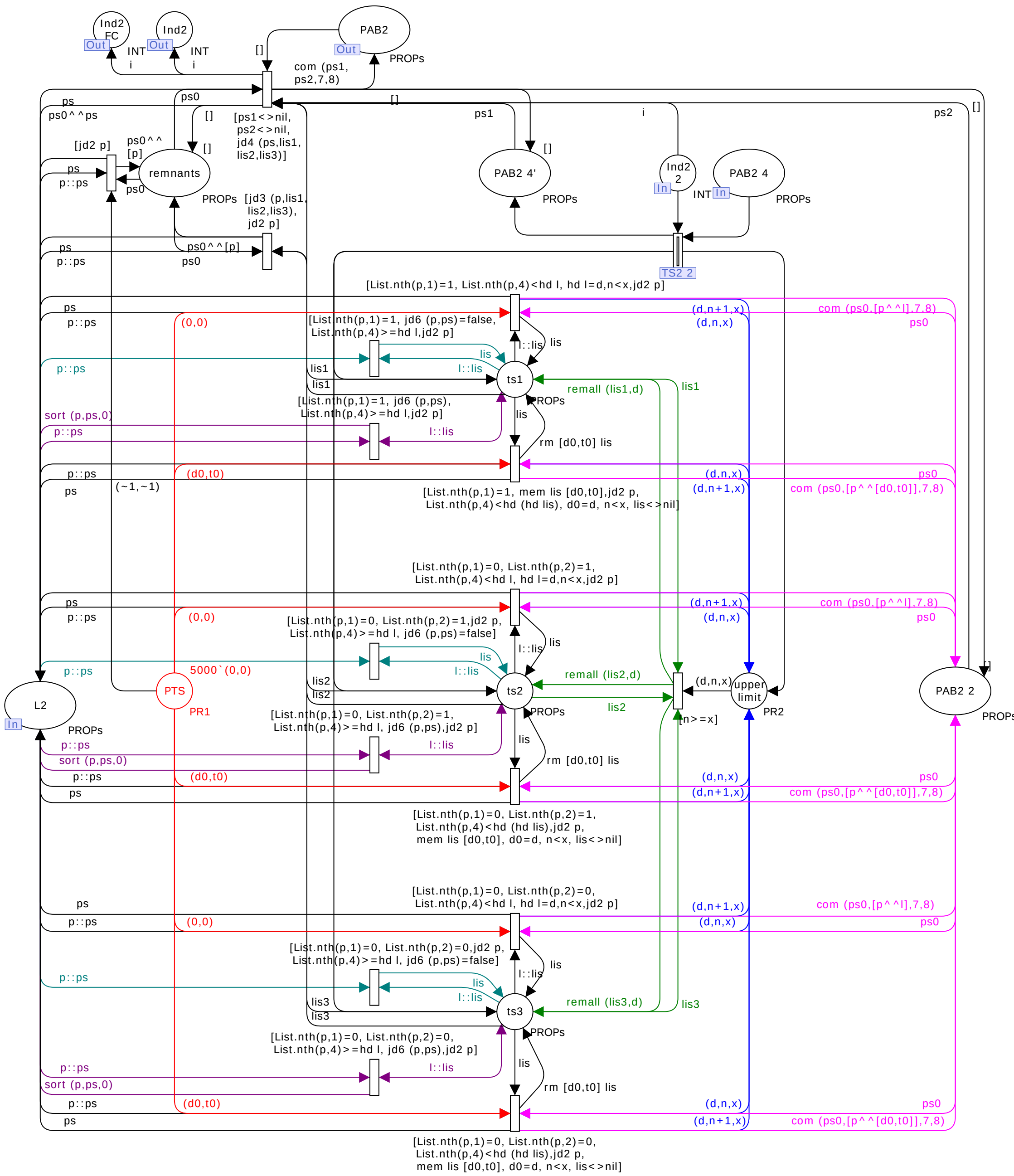

Figure A.19: The PN model of the booking of level 2 cases for MRI_2 in CPN Tools

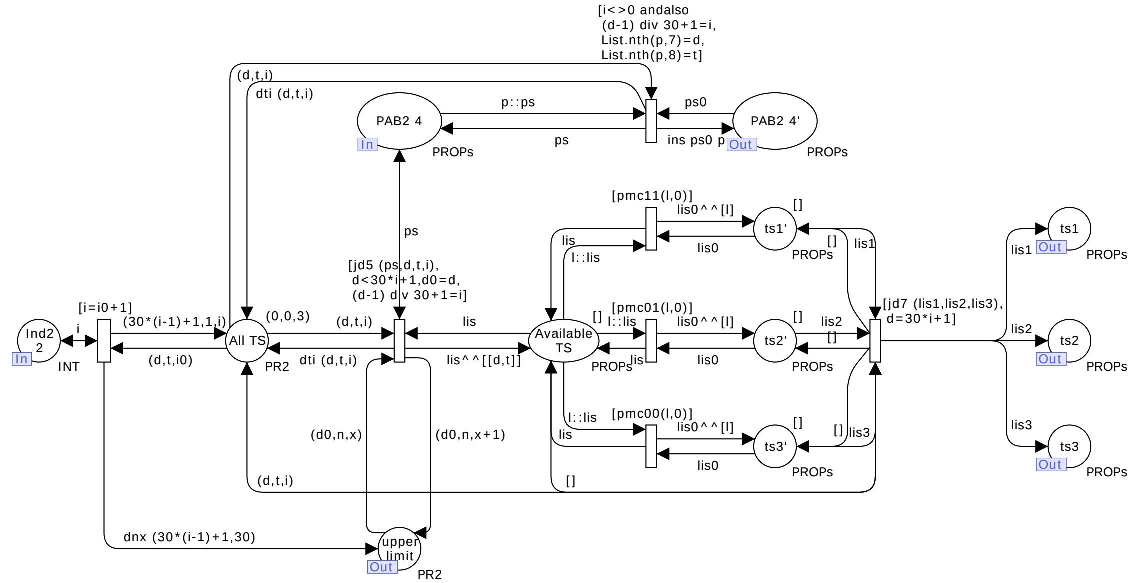

Figure A.20: The PN model of the time slot sorting of level 2 cases for MRI_2 in CPN Tools

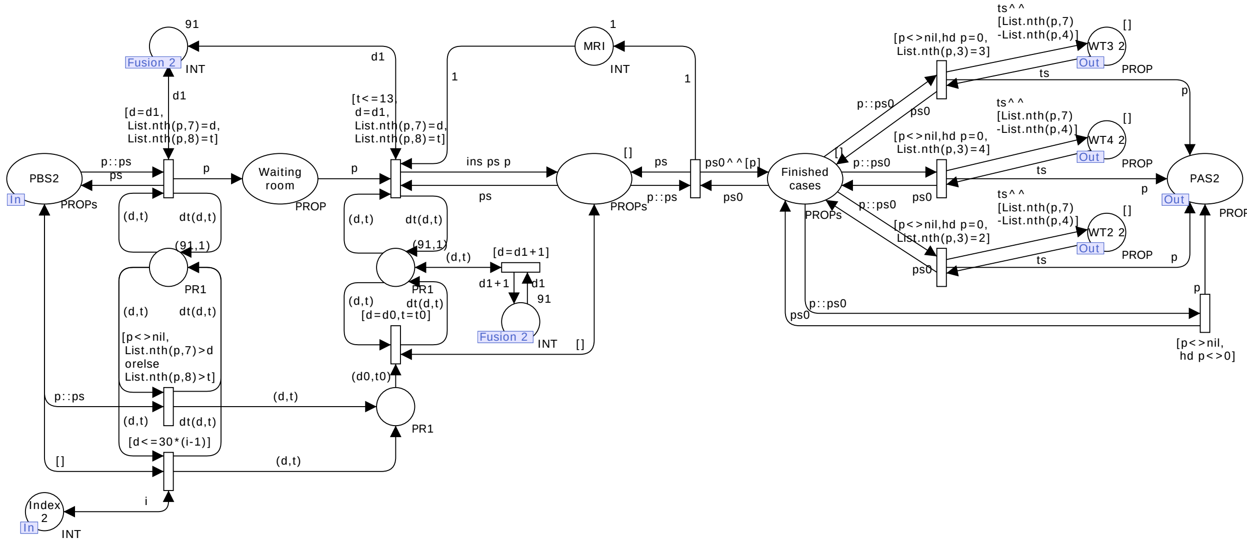

Figure A.21: The PN model of serving process for MRI_2 in CPN Tools

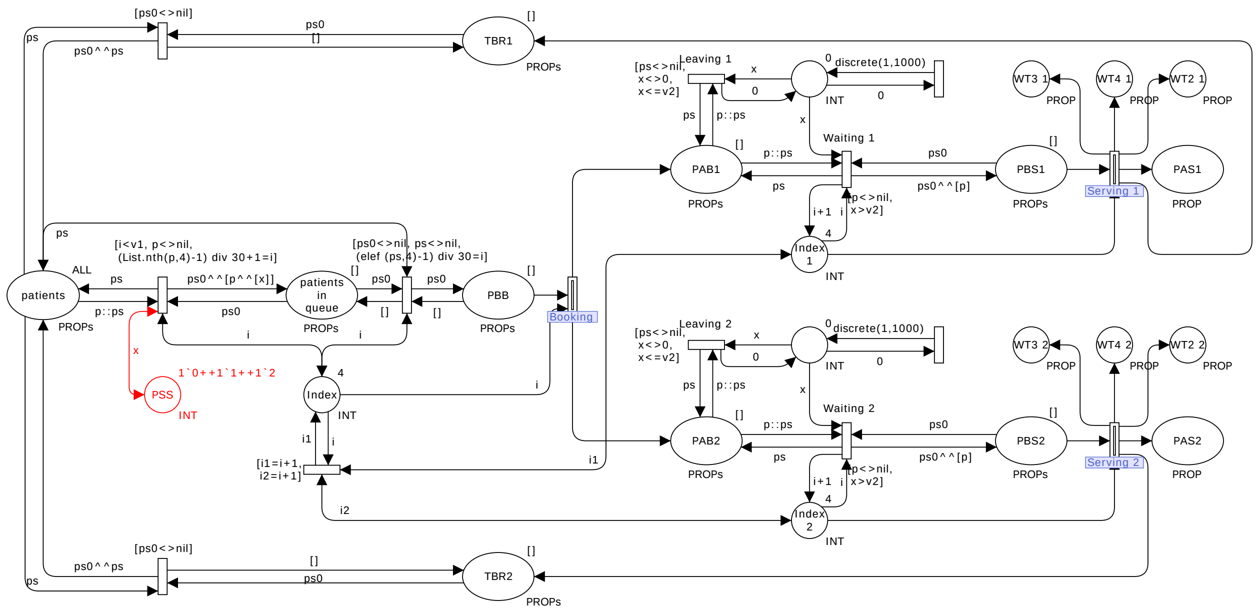

Figure A.22: The PN model of the whole process with service re-ordering in CPN Tools

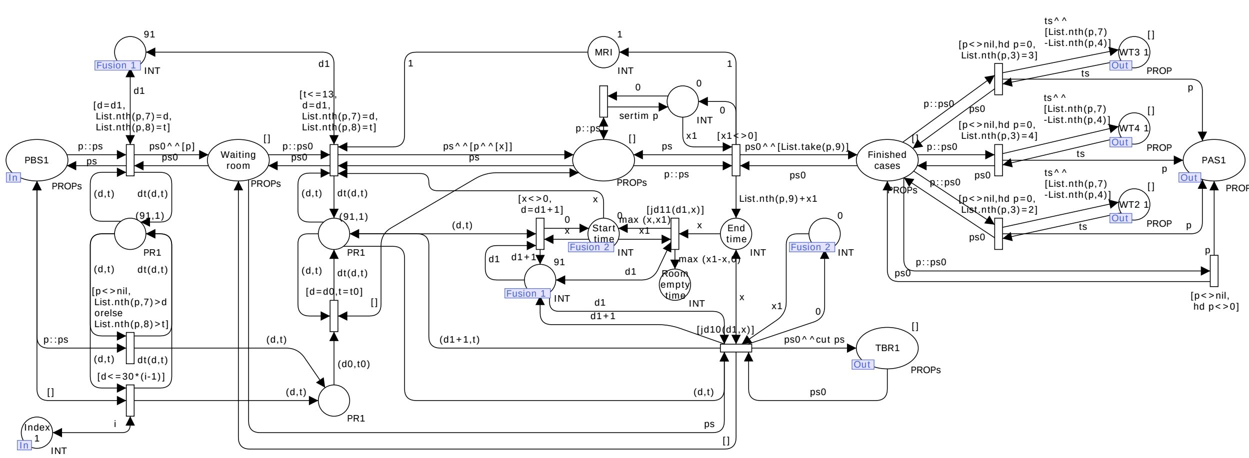

Figure A.23: The PN model of serving process for MRI_1 with service re-ordering in CPN Tools

Appendix B

EM Algorithm

Let $x_1, \ldots, x_n$ be the observed number of patients coming each day. We assume that they can be divided into two groups or situations. For each observation x_i, a random variable z_i is defined to indicate the group that x_i belongs to, i.e.,

$$x_i \in \text{group 1} \quad \text{if } z_i = 1 \quad \text{and} \quad x_i \in \text{group 2} \quad \text{if } z_i = 0.$$

We assume that z_i follows a Bernoulli distribution with the parameter of probability of success, γ, i.e. $z_i \sim \textit{Bernoulli}(\gamma)$ and conditioning on different values of z_i, x_i has a mixed Poisson distribution

$$x_i|z_i = 1 \sim \textit{Poission}(\lambda_1) \quad \text{and} \quad x_i|z_i = 0 \sim \textit{Poission}(\lambda_2).$$

In order to generate data from the distribution of x_i, the parameters $\theta = \{\gamma, \lambda_1, \lambda_2\}$ need to be estimated. As the observations of the latent variables z_i are unknown, the famous Expectation Maximization (EM) algorithm is used here.

EM algorithm is a technique used in point estimation and very suitable for the case containing latent variables or missing values. It consists of two major steps: the E-step and M-step. E-step computes the expectation of log likelihood of completed data with respect to the latent variables conditioning on the observations and current estimate of the parameters. M-step updates the estimate by maximizing the expectation function obtained in the E-step. Given initial values of those parameters, EM algorithm iterates these two steps to update the parameters such that the log likelihood function keeps increasing until it converges. In the following, details of the EM algorithm for our mixed Poisson distribution are derived.

Denote by $X = (x_1, \ldots, x_n)^T$ and $Z = (z_1, \ldots, z_n)^T$ the observed values and latent values, respectively. Based on the distribution assumptions, the log likelihood of the complete data is

$$\begin{aligned}
l(X, Z|\theta) &= \log \prod_{i=1}^{n} p(x_i|z_i)\, p(z_i|\theta) && \text{(B.1)}\\
&= \sum_{i=1}^{n} \log \left\{ z_i \left[\gamma p(x_i|\lambda_1)\right] + (1 - z_i)\left[(1-\gamma)\, p(x_i|\lambda_2)\right] \right\} && \text{(B.2)}\\
&= \sum_{i=1}^{n} z_i \left[\log \gamma - \lambda_1 + x_i \log \lambda_1 - \log x!\right] + (1 - z_i)\left[\log(1-\gamma) - \lambda_2 + x_i \log \lambda_2 - \log x!\right] && \text{(B.3)}
\end{aligned}$$

The third equality is because z_i can only take the value of 1 or 0. Suppose the current estimate of the parameters is $\theta^{(k)} = \left\{\gamma^{(k)}, \lambda_1^{(k)}, \lambda_2^{(k)}\right\}$. For the E-step, we need to take the expectation of $l(X, Z|\theta)$ with respect to $Z|X, \theta^{(k)}$. Note that $l(X, Z|\theta)$ is a linear combination of z_i and the computation of $E\left(z_i|X, \theta^{(k)}\right)$ suffices.

$$\begin{aligned}
E\left(z_i|X, \theta^{(k)}\right) &= p\left(z_i = 1|x_i, \theta^{(k)}\right) && \text{(B.4)}\\
&= \frac{p(x_i|z_i = 1)\, p\left(z_i = 1|\theta^{(k)}\right)}{p(x_i|z_i = 1)\, p\left(z_i = 1|\theta^{(k)}\right) + p(x_i|z_i = 0)\, p\left(z_i = 0|\theta^{(k)}\right)} && \text{(B.5)}\\
&= \frac{\gamma^{(k)} e^{-\lambda_1^{(k)}} {\lambda_1^{(k)}}^{x_i}}{\gamma^{(k)} e^{-\lambda_1^{(k)}} {\lambda_1^{(k)}}^{x_i} + \left(1 - \gamma^{(k)}\right) e^{-\lambda_2^{(k)}} {\lambda_2^{(k)}}^{x_i}} && \text{(B.6)}
\end{aligned}$$

For ease of notations, let $w_{1i}^{(k)} = p\left(z_i = 1|x_i, \theta^{(k)}\right)$ and $w_{2i}^{(k)} = 1 - w_1^{(k)}$. Therefore, the E-step is

$$Q\left(\theta|\theta^{(k)}\right) = E_{Z|X,\theta^{(k)}}\left(l\left(X,Z|\theta\right)\right) \tag{B.7}$$

$$= \sum_{i=1}^{n} w_{1i}^{(k)}\left[\log\gamma - \lambda_1 + x_i\log\lambda_1 - \log x_i!\right] + w_{2i}^{(k)}\left[\log\left(1-\gamma\right) - \lambda_2 + x_i\log\lambda_2 - \log x_i!\right] \tag{B.8}$$

For the M-step, taking derivatives of $Q\left(\theta|\theta^{(k)}\right)$ with respect to each parameter and setting them equal 0 yields

$$\frac{\partial Q\left(\theta|\theta^{(k)}\right)}{\partial\gamma} = 0 \quad \Rightarrow \quad \gamma^{(k+1)} = \frac{\sum_{i=1}^{n} w_{1i}^{(k)}}{\sum_{i=1}^{n}\left(w_{1i}^{(k)} + w_{2i}^{(k)}\right)},$$

$$\frac{\partial Q\left(\theta|\theta^{(k)}\right)}{\partial\lambda_1} = 0 \quad \Rightarrow \quad \lambda_1^{(k+1)} = \frac{\sum_{i=1}^{n} w_{1i}^{(k)} x_i}{\sum_{i=1}^{n} w_{1i}^{(k)}},$$

and

$$\frac{\partial Q\left(\theta|\theta^{(k)}\right)}{\partial\lambda_2} = 0 \quad \Rightarrow \quad \lambda_2^{(k+1)} = \frac{\sum_{i=1}^{n} w_{2i}^{(k)} x_i}{\sum_{i=1}^{n} w_{2i}^{(k)}}.$$

The procedures of the algorithm are described in the following.

- Step 1. Assign initial values to the parameters, $\theta^{(0)} = \left\{\gamma^{(0)}, \lambda_1^{(0)}, \lambda_2^{(0)}\right\}$ and set $k = 0$.
- Step 2. Compute $w_{1i}^{(k)}$ and $w_{2i}^{(k)}$, $i = 1, \ldots, n$, as follows,

$$w_{1i}^{(k)} = \frac{\gamma^{(k)} e^{-\lambda_1^{(k)}} {\lambda_1^{(k)}}^{x_i}}{\gamma^{(k)} e^{-\lambda_1^{(k)}} {\lambda_1^{(k)}}^{x_i} + \left(1-\gamma^{(k)}\right) e^{-\lambda_2^{(k)}} {\lambda_2^{(k)}}^{x_i}} \quad \text{and} \quad w_{2i}^{(k)} = 1 - w_{1i}^{(k)}$$

- Step 3. Update the parameters to $\theta^{(k+1)} = \left\{\gamma^{(k+1)}, \lambda_1^{(k+1)}, \lambda_2^{(k+1)}\right\}$, where

$$\gamma^{(k=1)} = \frac{\sum_{i=1}^{n} w_{1i}^{(k)}}{\sum_{i=1}^{n}\left(w_{1i}^{(k)} + w_{2i}^{(k)}\right)}, \quad \lambda_1^{(k+1)} = \frac{\sum_{i=1}^{n} w_{1i}^{(k)} x_i}{\sum_{i=1}^{n} w_{1i}^{(k)}} \quad \text{and} \quad \lambda_2^{(k+1)} = \frac{\sum_{i=1}^{n} w_{2i}^{(k)} x_i}{\sum_{i=1}^{n} w_{2i}^{(k)}}.$$

- Step 4. If $\left|\gamma^{(k+1)} - \gamma^{(k)}\right| + \left|\lambda_1^{(k+1)} - \lambda_1^{(k)}\right| + \left|\lambda_2^{(k+1)} - \lambda_2^{(k)}\right| < \varepsilon$, it converges and output $\theta^{(k+1)}$, where ε is a predefined threshold. Otherwise, set $k := k+1$ and goes to Step 2.