INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Modeling and Simulation of Healthcare Pharmaceutical Environments: A Petri Nets Approach

Filipe Gonçalves Pereira

Relatório de Estágio para Obtenção do Grau de Mestre em Engenharia Industrial

Ramo da Engenharia Electrotécnica

Orientador: José Gonçalves

Co-Orientador: Paulo Leitão

Supervisor Empresa: José Luis González Sánchez (Fundacion CARTIF)

Setembro 2010

“A great mind is one that is neither ancient nor modern; it is neither ashamed of the old nor afraid of the new. It thinks neither in terms of old traditions nor in terms of new fashions. It is only concerned with the true and the workable.” -N. Eldon Tanner

Dedico este trabalho acima de tudo e de todos aos meus pais, os verdadeiros “culpados” e responsáveis daquilo que sou e faço, quem me permitiu sempre tomar as minhas próprias decisões e ficou á minha espera para me apoiar caso corresse mal, e me felicitou com uma certa naturalidade caso corresse bem, como se não esperassem outra coisa:

Ao mestre André que ao longo de todos estes anos tem sido muito mais que um Pai, um professor, um patrão e um amigo, é um exemplo de vida e de homem que espero para sempre conseguir dignificar e orgulhar.

Á mãe mais preocupada, cuidadora, trabalhadora e dedicada que alguma vez podia pedir. Nem sei como retribuir o apoio incondicional, a fé que sempre depositaste em mim, a maneira como esperavas sempre mais e melhor. Tanto amor me deste que te devo tudo que sou e virei a ser… Elisinha… I love you too.

Joe, fomos como azeite e agua durante muito tempo, agora é mais uma cena ying-yang. Pões sempre as coisas numa prespectiva que acaba por me convencer…um bocadinho.

Abstract

This study examines the drug distribution procedure in hospital, based on a bibliographic research and an interview with the pharmacy director of “Hospital del Rio Ortega” in Valladolid, Spain. It was necessary to elaborate a theoretical flux-gram of the medication distribution procedure describing the various stages in the process, as well as the objectives of the system, and its critical issues. All these steps depend on various health care providers and health care assistants directly linked. This systemization allowed the observance of the critical knots, understood as the processes and elements with critical impact over the quality of the distribution system. The understanding of the possible causes of the system’s problems serves here as the basis for the formulation issues that make the evaluation possible.

There are two basic known types of drug distribution systems: collective and individual. The collective system is the most primitive, although there are hospitals worldwide that adopt this system, considered to be the simplest and lowest in cost of deployment. The individual distribution system is more complex, it needs more professional pharmacists and a hospital pharmacy working around the clock and has as main features the fact that the product or drug is dispensed per patient, and not for the sector in which one may find himself hospitalized. The research of medication errors due to the type of distribution system adopted by hospitals has created a debate on the advantages and disadvantages of the existing systems. The use of a system model for distribution of medicines will provide a more appropriate reduction in risk of errors and foremost a reduction in hospital costs. One objective of this study is to show the best method of distribution of medicines in a hospital pharmacy, that is, the method of distribution in which the chance to experience a medication error would be smaller, and distribution times as well as staff needs were reduced to a minimum.

The main objective of this work is to model the medication distribution process in Health-care environments, using Petri Net formalism, and extract

information from those models through simulation, to support the decision making relating human resource availability.

This work was done during a stage at the Cartif Foundation in Valladolid Spain, with the cooperation of the Valladolid University’s School of Industrial Engineering.

Resumo

Este estudo examina o procedimento para distribuição de medicamentos dentro de um hospital, baseado numa pesquisa bibliográfica e uma entrevista com o director da farmácia do hospital “Hospital del Rio Ortega” em Valladolid, Espanha. Foi necessário elaborar um fluxogram teórico do procedimento de distribuição de medicamentos descrevendo os vários passos do processo, e ainda os objectivos do sistema e os seus pontos críticos. Todos estes passos dependem de vários assistentes e auxiliares de saúde directamente ligadas. Esta sistematização permitiu observar os pontos críticos, compreendido como os elementos com impacto critico na qualidade do sistema de distribuição. A compreensão das possíveis causas dos problemas do sistema servem de base para formular questões que tornam possível a sua avaliação.

Existem dois sistemas básicos de distribuição de medicamentos: Colectivo e Individual. O sistema colectivo é o tipo de distribuição mais primitivo, mas apesar disso existem muitos hospitais do mundo que adoptam este sistema por ser considerado o mais simples e de implementação mais barata. O sistema individual é mais complexo e necessita de mais profissionais qualificados, nomeadamente farmacêutico e\ou técnico de farmácia, a trabalhar 24 horas por dia, tendo como principal característica o facto de que cada produto ou medicamento é dispensado por pessoa e não por sector ou andar em que um possa estar hospitalizado. A pesquisa de erros médicos devido ao tipo de distribuição de medicamentos adoptado pelos hospitais tem criado debate sobre as vantagens e desvantagens dos sistemas existentes. O uso de um modelo para a distribuição de medicamentos levará á redução de erros e acima de tudo pode contribuir para a redução de custos do hospital. Um dos objectivos deste estudo é mostrar o melhor método de distribuição de medicamentos de uma farmácia hospitalar, isto é, o método de distribuição em que se minimiza o risco de erros de medicação, tempos de distribuição assim como necessidade de recursos humanos.

O objectivo principal deste trabalho é modelar o processo de distribuição de medicamentos em ambiente hospitalar, usando o formalismo das Redes de Petri, e extrair informação desses modelos através de simulação para apoiar tomadas de decisão relativas á disponibilidade de recursos humanos.

Este trabalho foi desenvolvido durante uma estadia na Fundação Cartif de Valladolid em Espanha com a cooperação da Escola de Engenharias Industriais da Universidade de Valladolid.

Acknowledgements

I would like to begin thanking my coordinators José Gonçalves and Paulo Leitão from IPB not only for the guidance, support and information these past few months to help me overcome this challenge, but also for being great teachers and friends throughout my college years.

I wish to thank D. Juan Carlos Fraile and D. José Luis González for five months of work, information, fun, study travels and doing everything they could to make me feel comfortable in a different country and culture.

I must also thank my room-mates Anildio and João who always had an incentive to work or an idea to try and help out when things got “strange”.

Last but not least, I thank my friends and family that supported me and always encouraged me to do more and try harder all these years.

Acronyms

IPB	Polytechnic Institute of Bragança
ECTS	European Credit Transfer and Accumulation System
HdRO	Hospital “del Rio Ortega”
UDMDA	Unit Dose Medication Distribution System
KM	Kardex Megamat (Logistic Support Unit)
PSS	Pyxis Supply Station
WH	Warehouse
EDSPNs	Extended Deterministic Stochastic Petri Net’s
SCPNs	Stochastic Colored Petri Net’s
GUI	Graphical User Interface
GSPN	Generalized Stochastic Petri Net
CG	Care Giver
VASADU	Vertical Automated Storage and Automatic Dispensing Unit

Table of Contents

Table of Figures

1 - Introduction

This report was written in the context of the Final Project course of the Master degree in Industrial Engineering, specialization of Electrical Engineering, from the Polytechnic Institute of Bragança. The Final Project course is one year long and worth 42 ECTS, the first six months were focused on the preparation for the internship which consisted in the student preparing himself on the subject and tools that would be studied and used in the second phase which started on March 1st and lasted five months, until the 30th of July 2010. The project was done at the Cartif Foundation of Valladolid, Spain, with the supervision and cooperation of D. Juan Carlos Fraile and D. José Luis González, both professors at the University of Valladolid and representing the Cartif Foundation, which is a non-profit organization, dedicated to research and development in various areas including automation and control.

The internship activities were focused on modeling, extraction of information and knowledge of the Petri nets to support the monitoring and decision-making in the drug distribution procedure verified at the local hospital “Hospital del Rio Ortega” (HdRO).

The dispensing of medication is a technical and scientific activity directed towards the patient, of great importance for the success of the pharmaceutical and therapeutic treatment, which should only be carried out by qualified professionals, in this case a pharmacist. A rational distribution of medication consists in assuring consumers to receive the right product, in the right quantity and specifications, and in a reasonable time, obtaining the best cost efficiency [1].

The drug distribution procedure has various stages, starting with a prescription and ending with the administration of the drug to the patient. All the steps along the procedure depend on the various health care professionals, and are directly connected. Thus, the prescription is the Doctor’s responsibility, the dispensing is the pharmacist’s responsibility, the distribution or delivery to the floor or sector where the drug is needed is the responsibility of a care-giver, and the administration as well as patient monitoring are the responsibility of the nurse.

The type of drug distribution system chosen and applied in a hospital is one of the sensitive aspects that can cause grave and dangerous errors in medication, having as consequences harmful results in patients [2].

1.1 - The problem

The organization or the reorganization of the logistic chain constitutes a major stake for the public health establishments, because its improvement is one of the main reserves for saving money in this sector. In this direction, the optimization of the pharmaceutical logistic chain is very important: from the pharmaceutical product’s arrival, till the moment it reaches the patients bed [3].

The rational organization of the pharmacy [4] may be seen as a necessary evolution. However, major difficulties appear during this evolution, because the hospital pharmacy has such a large diversity of stored products. Within the same pharmacy, the system must be able to manage several areas dedicated to different kinds of storage: an area can be intended for the articles to store at low temperature (cold room) and another, for example, for dangerous products (safe room). This division in areas can also be relevant according to the rate of inventory turnover of certain products. Indeed, the frequency of restocking varies in an important way from one care unit to another.

The organization of the arrangement of the pharmaceutical products and their installation must satisfy a certain number of aims such as the accessibility to the stored articles, the improvement of the working conditions, the reduction in time and the preparation errors. These objectives involve the definition of criteria of internal organization of pharmacy [5] such as the products availability, the management policy, the storage cost, ergonomics, as well as the constraints to be respected such as the climatic conditions of the buildings and available surfaces. Although the advantages are easy to recognize (reduce in medication related errors, active participation from the pharmacy staff, and cost reduction related with minimizing medication spoil), it is necessary to highlight the fact that the system cannot disable the hospital pharmacy structure that allows itself to fulfill its objectives, which are the rationalization of medication costs, and enough safety that assures the process errors due to drug distribution are reduced. Obviously, the objective of this work is not to point out answers or final solutions for these problems, but to approach them in a way that can be very useful in the future understanding and developing of similar case situations.

In this work, the object of study is, only, the drug distribution process, considering of course, all the respective process participants. This leads to the study of the different types of prescriptions or possible drug requests from the doctor figure to the pharmacy, which can have different priorities or procedures, depending on the type of order.

1.2 - Objectives and contributions

The main objective of this work is to model the medication distribution process in Health-care environments, using the Petri Net formalism, and to extract information and knowledge from those models through simulation, to support the decision making aiming to improve distribution times and costs.

Petri Nets formalism [20] is a modeling tool with a well defined mathematical background and a good graphical environment that allows an easy understands. Combining these two qualities makes Petri Nets a powerful and suitable design tool to building mathematical models describing the system behavior of event-driven systems, as the medication distribution process is.

For this purpose, a case study based on the medication distribution process of the HdRO hospital will be used. The Petri nets models and posterior simulation will be performed by using the TimeNet software tool. It is a very powerful tool that allows users to create either deterministic or stochastic Petri nets, (in this particular work stochastic nets were used). The models will be simulated under different scenarios (e.g., number of people dispatched to an order), and the results will allow to verify eventual delays, buffer needs, staff re-location solutions, etc.

1.3 - Document organization

This document is divided in eight chapters, the present one being the first and containing an introduction to the approached theme, a brief explanation of the problem as well as the objectives of the study and the contributions that support the study.

Chapter two entitled “Concepts of Hospital Logistics” presents the hospital logistic main activities as well as the internal sub-systems considered to be crucial to the correct functioning of the distribution process.

Chapter three, entitled “Medication ordering and dispending”, presents and explains the medication distribution system supported by a flux graph to show the necessary steps for all types of prescription\request considered in this case study, and then describes all the steps taken by the medications along the process.

Chapter four, entitled “Stochastic Petri nets”, introduces the basic concepts of stochastic Petri nets as well as their time delays and marking conditions, and invariants in Petri Net models. It also introduces the Time Net software tool.

The fifth chapter, entitled “Modeling medication distribution processes”, explains how the Petri net model for the medication distribution process is achieved by using the TimeNet software and shows how the Petri net model evolves, using the token game function from the TimeNet software to exemplify how tokens move along the net.

The sixth chapter, entitled “Simulation of the Designed Models”, presents the variables used for simulation, both for the medicine cabinet request as well as the full Petri net model, defining the time delays as constants and the number of available resources as the exit variables that provide information in the form of a graph corresponding to each place or transition being watched. The resulting graphs are also analyzed in this chapter as they are presented.

Chapter seven contains the conclusions reached with the study and discusses some possibilities for the future work.

2 - Concept of hospital logistics

For years, public hospital managers and directors have been realizing that the services not directly connected with health-care (laundry, kitchen, maintenance and logistic activities), have excellent potential for improvement. With a revision and re-structuring of the way these operations are practiced, without reducing at all the quality of the clinical service. One of these services is the hospital logistics, aiming to effectively supply all health care products, as well as the necessary medication(s) for correct patient treatment.

A hospital's activities can be arranged in a group of central activities where all the internal processes of the hospital can be considered in groups. The patient support and treating is only the main activity, making the human resources, all the materials, the medical testes (x-ray's etc.) and all the clinical processes (diagnosing, medication etc.) support for the main activity. The correct balance of all these activities is what allows an institution to offer a good clinical service to the sick or chronically ill [5].

The basic activities developed in a hospital [13] are shown in Figure 1, and after, a short description of each of these activities is given.

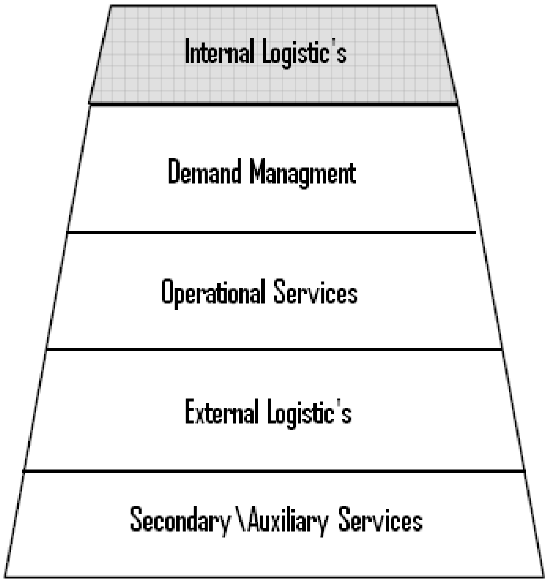


Figure 1 - Hospital main activities

Internal Logistics'- Is the group responsible for acquiring, receiving, storing and distributing the products necessary for the hospital's internal processes, which are referred to as *Hospital Logistics*.

Demand Management- Consists in analyzing, planning and assigning resources necessary to acquire goods and services.

Operational Services- Considers all the internal processes and activities that support the patients stay in the hospital.

External Logistics'- Is associated with the activities of clinical or medical follow up of the patient's condition.

Secondary\Auxiliary Services- Groups all the activities taken place inside the hospital and includes services like restaurants, stores, religious support etc.

Even though none of these can be disconnect from the others, internal logistics serve a more important role in controlling the processes that can lead the hospital to an improvement of its resources. The re-considering of activities in health care centre's or hospitals to improve the way these resources are applied, increasing efficiency, should allow better cost control, assuring global service quality to the patient.

2.1 - Internal logistic sub-systems

The internal logistics of a hospital can be considered to consist of four sub-systems, which are production, storage, transport and handling, and information gathering along the process. All these elements should interact to fulfill one objective: satisfy the internal demands with as much quality and as less costs as possible. In other words, assure management satisfaction. Each stage of the sub-process and the activities that should be developed to guarantee a good management are:

- **Provision**: this sub-system is in charge of activities related with shopping\buying functions, orders, inventories, storage, transport, product planning, and information management. This sub-system includes all the processes that provide the necessary material's that the productive sub-system needs to complete its functions.
- **Production**: is the sub-system that comprehends the intermediate storage of products, either in the central pharmacy or in storage spaces and medicine cabinets throughout the hospital, as well as the prescription preparation, maintenance and technical support. This sub-system serves to optimize the physical movement process of the materials in the hospital.
- **Distribution**: manages and distributes all products since they leave the general warehouse, until they reach their destiny, or point of consumption. The objective is to consider the internal transportation and distribution frequencies, to try to synchronize the demand from each service with the production, to try and reduce delivery times and warehouse stock.

From a logistic point of view, a hospital is no more than a production centre where numerous internal logistic processes occur (Figure 2). Starting with the planning and acquiring of goods, and including the design of the distribution system up to the point of consummation, all the sub-processes that intervene in the hospital logistics' process should be planned and executed as precisely as the logistic plan is designed. For example, in the provisions sub-system, the acquiring function includes search and demand from a supplier of the necessary materials for the production centre, which means all the raw materials the production sub-system will need. What to order, in what amount and when to order it should be based on a previous study of the consumed materials individually, to try and make an accurate prediction of the system needs. Nowadays, very few hospitals or health centers create an elaborated study of what in fact they need, and base their planning on consumption history and educated guesses about the future needs of the hospital. By using simulation tools that are not very expensive, it's much easier to comprehend the supply needs which in turn allow establishing a stock level closer to the hospital’s reality and needs, liberating funds to be spent in other sectors.

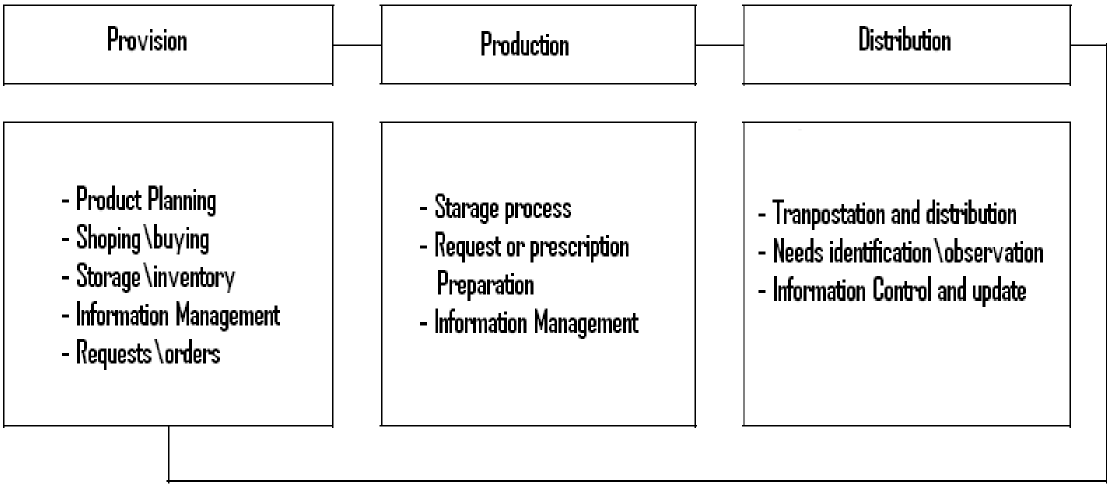


Figure 2 - Main logistic processes in each sub-system

The most important aspect within the production sub-system is the preparation of the request or prescription. This phase contemplates the selection and gather of all the products from their storage location, as well as the preparation for transport and distribution to consumers. The time spent in this sub-system affects the “response time” verified by the “costumer” (person waiting for medication). Aside from hospital, like HdRO that have installed automated dispensing units, most hospital warehouses do not have any type of automation in the request preparation. The most usual model found is one where the pharmacist or pharmacy technician move around the warehouse and gather the different medications as they go while considering the quantities available and take notes of the ones that need to be corrected to fulfill the minimum number of available doses. There are associated costs with this process, not only the personnel but also warehouse related costs (cleaning, maintaining, repairing), utility costs (water, electricity, gas etc.) and safety costs (insurance, surveillance). These and other costs that may seem insignificant, if reduced or even eliminated, may represent a considerable amount of money to one of the largest sectors in worldwide economy that moves millions and millions of euro’s and dollars every year.

3 - Medication ordering and dispending

This chapter describes the basis for the unit-dose medication distribution system, as well as give an inside view of how medication is ordered, dispensed and transported throughout the hospital case study used in this work, the HdRO. Four basic types of prescription-preparation-administration procedures were distinguished for modeling, to adequately describe how the global procedure for drug distribution takes place. These four procedures as well as the logistic support machinery that can be found in the pharmacy or anywhere else in the hospital are also described.

3.1 - Unit-dose medication

The dispensing of medication is a pharmaceutical action associated with the delivery of an informed prescription by a qualified professional (Doctor), for a specific patient. One of the dispensation types introduced recently in hospitals as an effective way to reduce errors in the prescription-preparation-administration process drugs go through in a hospital is the unit-dose medication distribution system (UDMDS). This system consists in delivering a prescribed drug in an individually prepared and packaged form, to be administrated along a certain period of a treatment [7].

The objectives of the UDMDS is essentially to ensure the maximum security and efficiency when dispending medication, reducing medication costs and allowing the pharmacist to integrate, in a substantial way, the health-care team, and finally to be able to track and detect any problem related with a drug since it was prescribed up to the moment it is delivered to a nurse for patient administration.

The service a pharmacy provides dispending medications when presented with a prescription is guaranteed to be safe and fast thanks to automatic dispensing machines, but also very important is the education and information given about the medication and it’s appropriate use. This results in better treatment or therapy results, and allows that a pharmaceutical follows through of the patients and their conditions and reactions to the treatment.

The central medication dispensing area is inside the HdRO pharmacy, and has the internal physical boundaries defined and with all the necessary resources to fulfill the medication dispensing in an easy opportune manner. The traditional system is one that provides a box or case with a certain number of medications to the nursing staff, whom then prepares each patients respective dose. This system has the disadvantages of requiring much more dedication by the nurses, not allowing an adequate control over which medications are stored in each section of the hospital. It leaves more possibilities for self-medication and can generate money loss, since no medications are returned even though they are not going to be consumed in that section of the hospital, despite they could be needed for other patients. The UDMDS offers the opportunity of a follow-through of the medication therapy effects on the patient which allows closer monitoring of the patient reaction to the drug from a pharmacy\therapeutic point of view. It has been shown, in numerous studies ([21], [22]) that this system is the safest for the patient, most effective from an economic point of view, and simultaneously it utilizes professional human resources much more effectively. Advantages of the UDMDS compared to other techniques include:

- More certainty that the right medication goes to the right patient, since each prescription is destined to each patient individually.
- The efficient and rational use of human resources involved in distribution, including nurses who assume a more direct role in the patient care.
- Cost reduction through the spoil control, reduction of self-medication and medication recovery after a patient treatment that doesn’t use the exact amount that would be supplied in a box.
- Error reduction in the prescription-preparation-administration procedure.
- More control by the pharmacist over the pharmacy personnel.
- Reducing the number of nurse-responsibility medicine cabinets.
- More control over medication costs per patient and per institution.

- Much easier to create a follow through profile of the patient response to the drug, facilitating information on adverse affects or identifying interaction between different medications.

The pharmacist upon receiving a delivery of *x* amount of different medications must then store them, either directly in a vertical automated storage and automatic dispensing unit (VASADU), or on the shelves destined for medications in boxes. The shelves serve as a buffer between the automatic unit and the pharmacy staff. In some occasions the shelves might allocate an excessive number of medications that didn't all fit in the automated unit but will be used before the ones that were stored in the system. In other occasions it may hold medications returned for some reason, and if there are more than a pre-determined number of those medications they will then be placed back in the automated system. The VASADU and the shelves used to hold medications in boxes are referred to as WH1 (Warehouse 1) and WH2 (Warehouse 2) respectively. Since all the medication is received in bulk quantities, the unit-dose packaging must be done by the pharmacist, who between orders packs a determined number of medications individually. They are then placed in another VASADU intended for unit-dose medications only, or on shelves that serve as buffers between the automated system and the pharmacy staff, as described for WH2. These means of storing (VASADU) and shelving unit-dose medications will further be referred as WH3 and WH4 respectively. Thus, upon receiving an order or prescription, depending on its type, the pharmacist takes either the medications in boxes or the unit-packaged medications, from the VASADU system or the storing shelves and fulfils the request or prescription.

3.2 - Logistic support -automated medication storage

To support the medication storage and facilitate the dispensation as well as make the process safer, the hospital pharmacy is equipped with automated machinery, although traditional shelves and cabinets are also used, either as a buffer or simply a faster easier way to access medications that are more commonly used (Figure 3). Storing medications on simple traditional shelves can however become confusing or even chaotic depending on how many different types being stored.

The storage support used in the central pharmacy of the HdRO, also found in many other logistic sectors (besides medication storage), is the ‘Kardex Megamat®‘ (Figure 4), which is a vertical automated storage and automatic dispensing unit that can be customized for the location where it is installed, in terms of height more than in terms of width since it can be up to 10m tall serving more than one floor and 4m long, but its maximum width is around 1.7 meters. The KM follows a “the product to the person” principle, which uses a computer to very precisely and in a matter of seconds make available the trays with the chosen medications. The shortest path for these trays is chosen, and since the motor has a frequency converter all the accelerating and braking done while rotating trays, is done very smoothly. All the products stored in the KM are protected by a sheet of steel, except for the access panel which is at a reasonable height for people to work comfortably. This panel is also where the control elements are, ordered in a very clear way. The access panel can be closed with an elevating door accessed manually.

Figure 3 - Difference between shelf storage and KM storage system

Figure 4 - The basic Kardex Megamat structure

The other medication storage and distribution support system present in the hospital case study is the ‘pyxis supplystation®’ system (PSS) (Figure 5).

Pyxis SupplyStation®
single-column unit
31″ W x 28″ D x 79.5″ H

Pyxis SupplyStation®
half-height unit
31.5″ W x 28″ D x 54″ H

Pyxis SupplyRoller® unit
31.5″ W x 28″ D x 56″ H

Figure 5 - Example of existing models of Pyxisss

The PSS is physically similar to regular medication transport cars or fixed cabinets with its drawers or compartments, with the distinct difference that the PSS is controlled by an on-board computer that monitors not only all the drawer access but also the number of available medications, and sends regular updates to the central pharmacy with the number of available doses of each drug. This means that the on-board computer (battery powered) only opens the drawer corresponding to the medication reference selected on the touch-screen, and it only opens it enough, so that the corresponding amount of doses that were selected can be reached. With this type of control the system can easily monitor the quantity of each drug at all times, and with that information, via wireless access the system sends hourly updates on the number of doses of each medication present in the system to the pharmacy. The pharmacists can then decide depending on the number of medications it contains, the best time to re-fill the system.

3.3 – Types of prescription

Four basic types of medication request were recognized in the drug distribution system of the HdRO, which are:

- Emergency/occasional prescription
- Regular/daily prescription
- Medicine cabinet request
- Pyxis automatic request

These types of distribution will be described in detail in the next sections.

3.3.1 - Emergency\occasional prescription

The first type of order or prescription considered is the emergency or occasional request. This type of order happens when in a sector of the hospital a determined medication is needed but it is not available in the pyxis supply station or in the closest medicine cabinet. Due to the high cost of the PSS (limiting the number of units in the hospital), the limited number of different drugs it can contain, and the unpredictability of medication needs per sector, often, be it an emergency or simply an adjustment in the patient’s treatment, a drug that wasn’t predicted in the daily medication order and happens not to be in the PSS or the closest medicine cabinet, the wanted drug must be ordered directly from the pharmacy and transported either by a care-giver or in case of emergency the transport suction tube. This order being of urgent nature can be performed over the phone or in case of non-urgent needs by faxing to the central pharmacy the respective prescription. This type of order happens in average once every hour or two, and takes approximately 5 minutes to prepare and only a minute to transport through the pneumatic suction tube or up to six minutes for a care-giver to transport depending on which sector it’s destined to. Administrating the drug in this case can be almost immediate since it’s an urgent nature request, or a few minutes if not and urgent situation. This type of prescription has the highest priority of all four types given its urgent nature.

3.3.2 - Regular\daily prescription

The regular or daily prescriptions are the ones which the nurses of each sector of the hospital order based on known treatments for the patients. Every day depending on the number of patients and the previously prescribed treatments, an order is made for all the medications needed in a sector of the hospital, since there are three distinct sectors on each floor and four floors in total (in the case study HdRO) every day 12 sets of orders are sent to the central pharmacy to be prepared. Each of these orders is then prepared and stored in a regular transport car with separate drawers with the respective unit-dose medications for each patient.

After the preparation, the transport is performed by one of the available care-givers who take the car to the respective sector, afterwards, the nurse or nurses in charge of administrating the medications to the patients take over the delivery process, which then becomes a matter of distribution. The preparation of the daily prescriptions takes around five hours for all twelve cars which represents about twenty-five minutes for each car, transport to the sector where it is needed can take anywhere between two and seven minutes depending on how far from the pharmacy the sector is. Administrating the entire car full of medications to all the patients along a sector can take up to sixty minutes. The daily prescription orders are second in order of priority since they absolutely have to go out every day at a pre-determined time according to the administrating conditions and time lines created by the nurses in each sector of the hospital.

3.3.3 - Pyxis automatic request

An automatic update is sent, every hour, from each pyxis supply station to the central pharmacy, containing information about the number of unit-doses of each drug that is in the system. Even though the PSS sends these hourly updates, the need to re-stock the system may not be critical until a certain number of a determined medication reaches a minimum; this is why the decision to re-stock the system is decided by the pharmacist depending on that minimum number of medications that should be present in the system. The update that leads to the decision of re-stoking the PSS is what will be considered a pyxis request throughout this report. The PSS contains unit-dose packed medications and can be re-stocked with as many units as desired or as many as can fit in the system. Despite the hourly updates each PSS is usually refilled three or four times a week, which represents an average of one order every forty-eight hours. Usually preparing the doses to refill the system takes around ten minutes and transport is also made in a large box all the way to the nurse responsible for inserting the medications in the PSS, since it is password protected and not just anybody can manipulate it.

3.3.4 - Medicine cabinet request

A medicine cabinet request is one that is destined to re-stock a specific medicine cabinet available in a certain sector of the hospital. Normally, medicine cabinets are located where there is no pyxisss available and vice-versa, where there is a PSS there are no traditional medicine cabinets. Unlike all the other orders or prescriptions, medicine cabinet requests are for boxes or cases of medications and not unit-dose medications. When a nurse thinks that a medicine cabinet is in need of more of a determined drug she will send an order to the pharmacy asking for the drug but instead of asking for *X* amount of unit-doses the order is for *Y* amount of boxes of the drug. Latter on depending on the sector’s patient needs the medications will then be individually packaged by the nurse for distribution.

This type of order only happens two or three time a week, which represents an average of one order every seventy-two hours, and takes around ten minutes to prepare (in the pharmacy). After the order is ready it is transported by a care-giver in a big box to the corresponding sector of the hospital which takes the same average time of transportation as other requests that is between two and seven minutes depending on what floor or section it has to be delivered to. The medications are then stored by a nurse in the medicine cabinet, job that can take up to ten minutes depending on the size of the order. The medicine cabinet request, similarly to the pyxis automatic request, is lower in priority since normally the orders are made before a product actually runs out, and if in fact there is an urgent need for a drug that is not in the medicine cabinet or in the PSS, an emergency order is made and the drug will be immediately delivered to the section of the hospital where it is needed.

3.4 - Distribution block diagram

The prescription types referred to in 3.3 can be described by a block diagram representing the basic steps in the distribution process. The distribution process happens in two distinct ways, depending on the type of order, since there can be unit-dose medication requests (3.3.1, 3.3.2, 3.3.3) or a request for boxes of medication to re-stock the medicine cabinets present in specific areas of the hospital (3.3.4). This means that three types of orders must share the resources where the medications are stored (WH3 & WH4). After analyzing the necessary steps for each type of order to take place, the following block diagram was made to illustrate in an easy and comprehensive way, not only the basic procedure for each type of order but also to compare them and see what makes them alike or distinct. The block diagram shown in Figure 6, considers the four different order\prescription types, and shows (describes) how each one dispenses and prepares medications for transport. It does not however specify the time involved in each preparation or transport, these are aspects so relative that can vary so much from hospital to hospital, or in the same hospital from one week to another that they will only be mentioned for simulation purposes in further chapters. What is in fact represented on the block diagram are the actors in the procedure (doctor, nurse, care giver and in some cases the PSS) as well as the pharmacist who has the main role in the preparation process. The bridge between the pharmacy and the section nurses is made by care-givers who manually transport the medications from the central pharmacy to the nurse station that requested them (doctors make prescriptions and nurses make orders for the medicine cabinet).

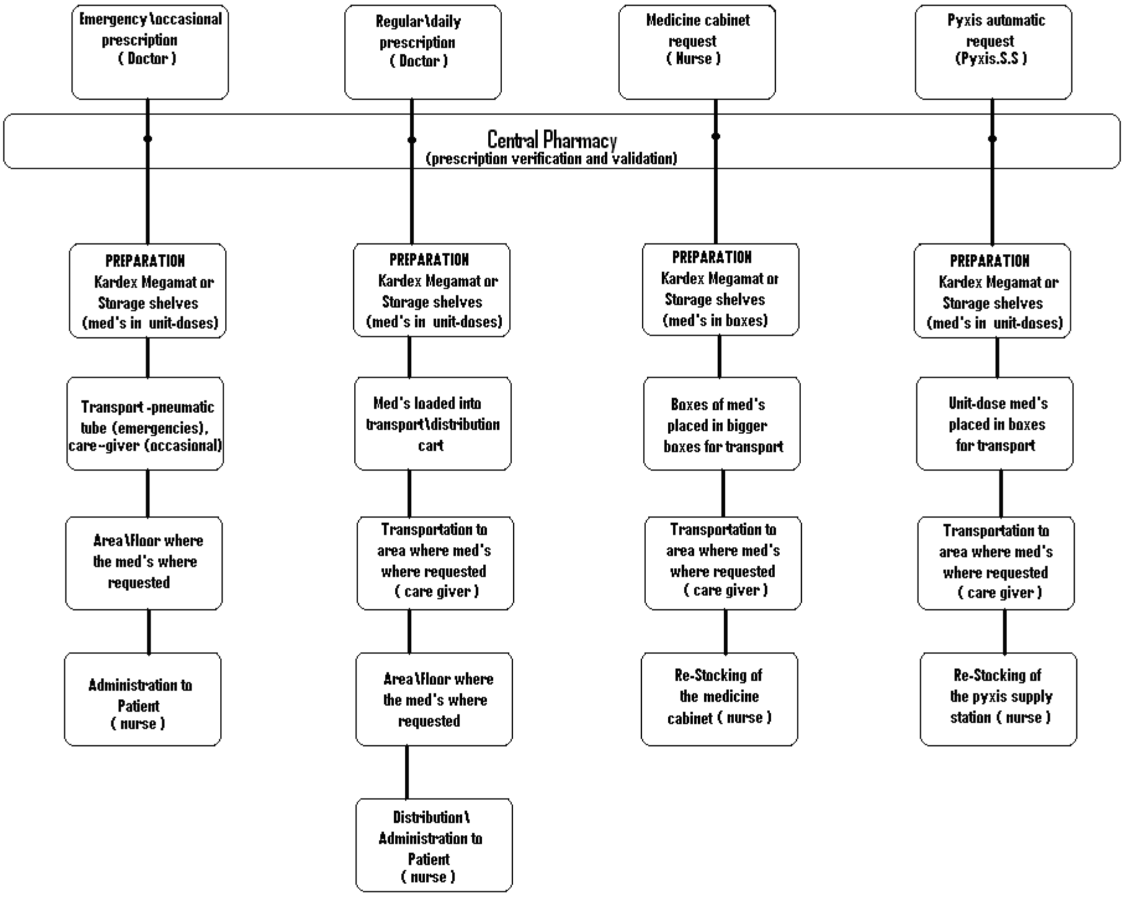


Figure 6 - Block diagram for the distribution process

3.5 - Distribution flux-graph

A flux-graph is a graphical representation of all the steps that constitute a process [8]. When monitoring a process it's important to have a plan or a direction to follow so that the object of the monitoring is known in terms of its goals, objectives and principles.

To make the proposed flux graph (Figure 7), initially a visit was made to the case study Hospital HdRO to see in firsthand how the pharmacy service interacted with the rest of the hospital using the unit-dose medication distribution system. This was a very important step in the study process because it made it easier to contrast the theoretical workflow that was researched with the reality that took place in a public hospital. The flux graph presented ahead shows all the steps of the medication distribution process, starting with the prescription, passing through the preparation and distribution from the pharmacy to distinct points of the hospital, and finishing with the administration to the patient or the storage in one of the logistic support units. According to Malik & Schiesari [8], in a flux graph, a process or action which is represented by a rectangle, usually only has one exit or flow direction, while the decision, represented by a diamond shape can have two exits or flow directions. What this means is that a decision may involve an evaluation or a judgment that can mean returning to a previous state of the graph. After elaborating the flow graph, a description was made for each step of the graph, with the intent of facilitating the general comprehension of the objectives of each stage of the distribution process and the points found to be critical to the process, as well as priority definitions between different types of orders.

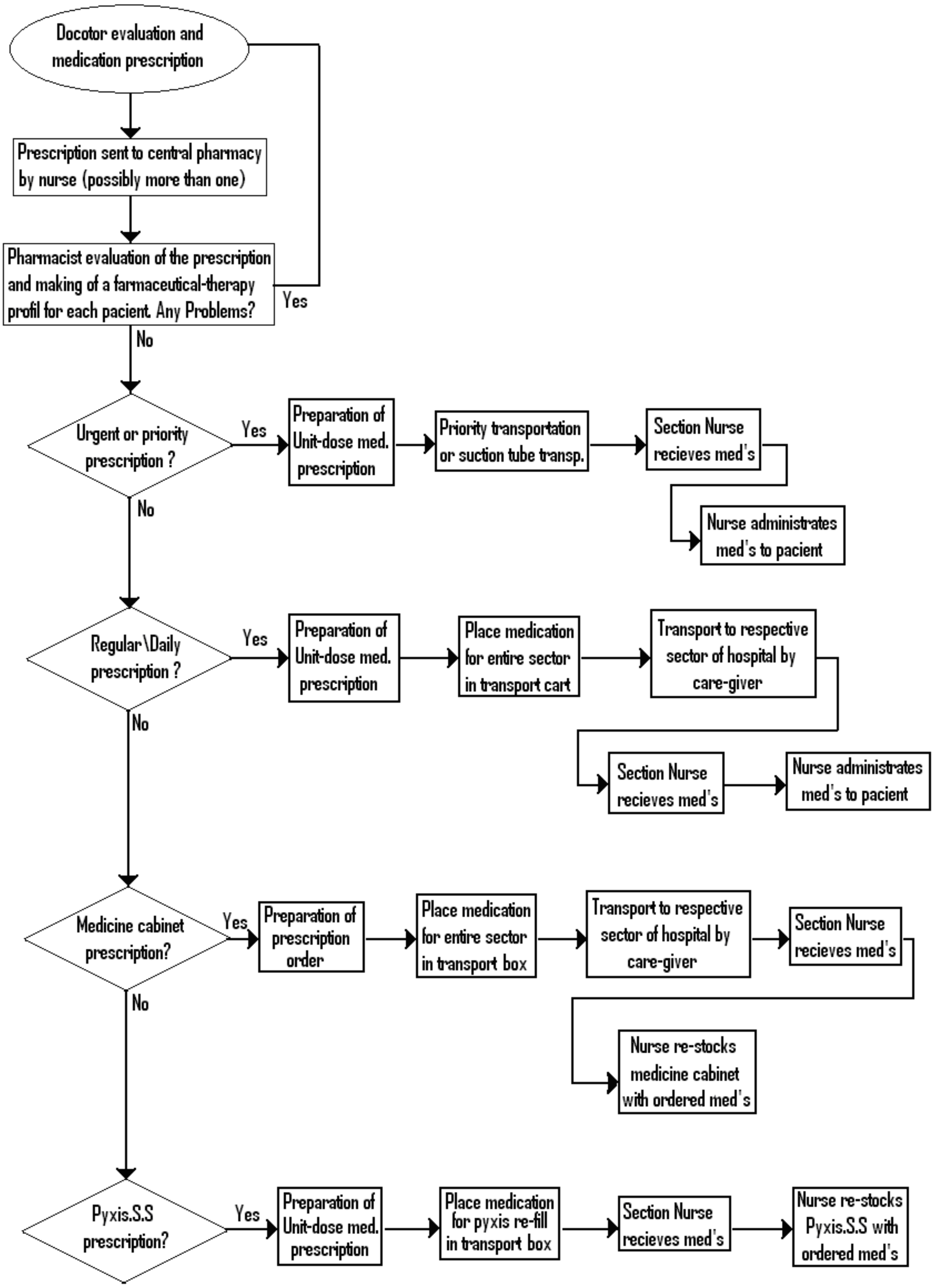


Figure 7 - Flux graph defining UDMDS steps

As referred previously and quoted in the "Spanish society of hospital pharmacy" by Napal [9], when discussing the UDMDS, it is more important to begin by reflecting on its principles than about the current procedures. It can be said that it is a distribution system not only involving a pharmaceutical work philosophy, but also mandates the involving of the entire health care team.

This philosophy or paradigm implicates the commitment of the pharmaceutical team in the health-care quality clinically as well as management-wise. If these principles are so important from the philosophical point of view, they should not be forgotten when describing de steps of the flow diagram, to approach them as much as possible to reality, increasing safety and reducing medication related errors. It is also necessary to accept the fact that the UDMDS is a system in which the pharmacist intervenes, previously checking and validating the doctor's prescription before continuing the distribution process.

The flow diagram's most significant aspects to be considered will now be described:

Prescription- This stage is executed by the doctor, who should complete the prescription process (manually or computerized), according to Lima [10], with enough information to create the patient's pharmaceutical-therapeutic profile (age, weight, diagnosis, admission date, room or bed number etc.), which is one of the systems global objectives. In hospital's like HdRO, where the patients profile and prescription are computerized, some of the stages of the evaluation process can be guaranteed to reduce human error like illegible entry data, abbreviations or initials among other things. A study made by Silva [11] showed that 75% of medication errors related with the prescription (in the hospital studied by Silva), where due to environmental issues like phone call interruptions, small spaces for allot of people, poor ventilation or work flow organization. The pharmaceutical profile is elaborated when starting the patient's medication history, using data collected with a prescription formulary or an interview with the patient by the pharmacist with a confidential information based relationship. Medication history information will allow the pharmacist to verify if the conditions related with the medication usage meet all demands, like clinical instructions for the medication usage, safety, effectiveness and treatment acknowledgement by the patient or a next of kin. The prescription should be made legible, it should use the generic medication name not use abbreviations or initials, and apply the international unit measure system.

Sending the prescription- The prescription is taken to the pharmacy for the preparation of the unit-doses and the evaluations found relevant to the process, by the pharmacist. It can be sent by fax, photocopy, carbon copy, or e-mail. When the sending is done in person, usually it will be by a pharmacy assistant, nursing auxiliary (care giver) or administration personnel.

Prescription evaluation- This is the stage where the pharmacist evaluates the prescription and checks for possible reactions or side-effects, confirms the dosage, makes sure there are no duplicate doses or pharmaceutical classes. This is also the stage where the pharmaceutical therapy profile is completed, where the medication being used by the patient will be evaluated, including the cross checking of the medication with possible side-effects, and also a follow up of the medication treatment verifying if the treatment objectives are fulfilled. This is a very important stage in the UDMDS since one of the main goals of the system is to rationalize the medication usage, guarantee process quality and reduce the medication errors or medication related problems. Medication related problems can be seen as any event experienced by the patient, and that involves or is suspected to involve the medication therapy, interfering with the fulfillment of the expected objectives [12]. The pharmacist's intervention in this evaluation stage can also be related with the medication substitution, either by a corresponding generic or the "therapeutic substitution", which switches one medication with another with a different active principle but with the same clinical indications. In this stage if a medication related problem is detected the pharmacist should contact the (prescribing) doctor to adjust the prescription. Some philosophic considerations exist in the difference between medication errors and medication related errors. Medication related errors are usually due to non compliance to one of the three bench-marks: clinical indication, effectiveness and safety. Not fulfilling one of these conditions can also be understood as flaws in the treatment. Medication errors are understood as all and any adverse event that can cause harm to the patient, related to the medication therapy, from gross errors in the prescription (units or quantities) to administrating errors or switched medications.

Medication distribution flow- Some medications can have a specific distribution flow due to pre-determined rules for their dispensing. Among these are antimicrobial's that have a growing problem with bacterial resistance and demand a great amount of concern relating the irrational use of these drugs. Other types of medications included in these specific distribution flows are drugs subject to a special type of control, or a set of sanitary rules (laws), and in some cases internal protocols depending on specific needs.

Preparation of unit doses- For the preparation, it is necessary to calculate the quantities and verify the existence of the medications in the pharmacy stock, usually done by a pharmacy assistant, but can also be done by the pharmacist. The unit-dose medications are in pre-determined quantities, dosage and concentration in the prescription are ready for patient administration are usually already packed in the pharmacy, being only necessary to separate the medications needed for 24 hours and transport them to the sector that made the order. If not already available, the doses can be prepared when separating the pharmaceuticals. It's important to remember that the most common drugs are separated and packed by the pharmacy technicians or assistants, then stocked until needed. In both cases it's fundamental to have an adequate preparation area. This phase of the distribution process is very important to rationalize the costs related with medication, also creating a positive impact in security and quality of the system due to the dose precision. This group of factors avoids the spoiling of doses and loss of medication or expired batches since the medications sent to the nurse sector are already packed in the quantity, dosage and concentration determined by the doctor in the prescription. This can be found to be a critical step in the process since the dividing and re-packing of the medication after being removed from its original package, requires a procedure that will conserve the integrity and characteristics of the medication.

Confirmation of prescription- The unit-dose medications, after being prepared are confirmed and double-checked by the pharmacist or by the pharmacy assistant, comparing the prescription with the doses gathered to fill the prescription. This is also a very important stage, since it is one that aims to guarantee the process quality, as well as reduce medication errors, mainly related with dispensing. The previously mentioned study by Silva [11], concluded that 20% of, medication errors related with the distribution, were attributed to flaws in the reviewing and confirmation plan. If there are any inconsistencies, the doses can be recovered, and in cases where they can't they are destroyed. If possible, a correction is made (label related errors for example), and the medication is then sent for transport to the nurse sector where it was ordered from.

Transport to respective sector- After the pharmacist or the pharmacy assistants are finished preparing and confirming all the ordered medication's, a care-giver is in charge of the transportation of the medications, either in the distribution cars (regular\daily prescriptions), or in a box, from the pharmacy to the sector of the hospital where they were ordered and delivers them to the nurse in charge of administrating the medications to the patients or re-stocking a medicine cabinet or the Pyxis support station.

Nurse reception and patient administration- The nurses upon receiving the ordered medications, once again compare the received drugs with the ordered drugs in the prescription. If any inconsistency is found that can be pharmacy process flaws or prescription mistakes, then the medications are returned to the pharmacy. The unit-dose medications are administrated to the patients very easily since they come ready, and save nurses preparation time. They are also identified and the time-line for administration should be discriminated on the label, which gives the patient security and process quality, reducing medication errors. In this stage it is necessary for the nursing team to be conscious and cooperative with the system so that the package integrity of the unit-doses is preserved up to the moment of administrating the medication to a patient, so that the UDMDS can reach its proposed objectives.

4 - Stochastic Petri nets

The classical Petri net is a directed bipartite graph introduced by Carl Adam Petri in 1962 as the result of his PhD work [23]. The two types of nodes are called places and transitions. In a Petri net, places and transitions are connected via arcs. While the places typically stand for the passive system elements which may be interpreted as conditions, the transitions represent the active system elements as events. Thus, every arc in the net connects an active and a passive element and since there are only direct arcs in Petri nets, they are represented by arrows. In graphical representation, places are depicted as circles and transitions as rectangles. According to the arc direction, every place has a set of pre-transitions and a set of post-transitions. In turn, transitions without replaces (*post places*) are called input (*output*) transition and are depicted as flat rectangles. The firing of an input transition is not restricted by the net, i.e., the event modeled by an input transition that may take place represents an event, which is only enabled to take place when all the preconditions of that event are fulfilled. The fulfillment of a condition is carried through the existing tokens in the places. If all pre-places of a transition are marked sufficiently with tokens (e.g., if the corresponding biochemical components are available), this transition may fire (e.g., the reaction takes place). If a transition fires, one token is removed from each of its places and added to each of its post-places. Thus, the tokens are the dynamic elements of the net. Altogether, arcs connect an event with its preconditions which must be fulfilled to trigger this event, and with its post-conditions which will be fulfilled when the event takes place. Principally, a place in a discrete net may carry any integer number of tokens, indicating different degrees of fulfillment, since the number of pre-places of a transition generally has not to be equal to the number of post-places of this transition; the number of tokens in the whole net is not conserved [6].

In Figure 8, a Petri net example is shown. Places are referenced with p- and a number, and transitions by t-and a number. In this example two types of transitions are used, the thin bars are instant transitions, and the boxes are timed transitions or delayed transitions (which will be introduced latter).

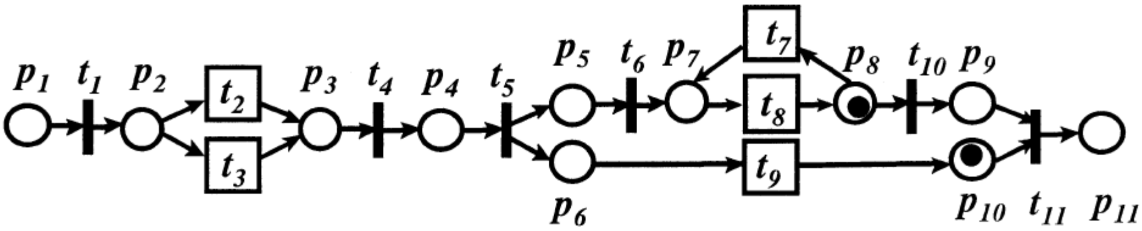


Figure 8 - A Petri net example

Note that both types of transitions respect the firing rule. In the Petri net depicted Figure 9, places p8 and p10 are marked. The transitions that are enabled are t7 and t10. In particular, transition t11 is not enabled. If transition t10 would fire on basis of the token in p8, then t11 becomes enabled since a token would be added to p9. As a final remark, transitions t7 and t10 are said to be in conflict, as they compete for the same token to fire, in cases like this priorities must be defined [15].

4.1 - Timed Petri nets

Petri nets were originally proposed as a causal model without any notion of time or probability. In fact, the concept of time was intentionally avoided in the original proposal of Petri nets. The addition of time restricts the behavior of the net, i.e., the dynamic behavior of the Petri net is only partially reflected by the network structure. However, for many practical applications, the addition of time is a necessity. Without an explicit notion of time it is not possible to analyze the performance (e.g., throughput and utilization) of the modeled system. Since the early seventies there has been a discussion within the Petri net community on the addition of time. More theory-oriented researchers oppose or simply ignore timing issues. More application-oriented researchers advocate and investigate different timing mechanisms and analysis techniques [15].

4.1.1 - Location of delay

There are many ways to introduce time into Petri net models. As explained in the previous section, a Petri net consists of places and transitions connected via arcs. This means, time can be associated with places, transitions, or arcs. In most timed Petri net models, transitions determine time delays. In only a few models, time delays are determined by places and/or arcs. It seems that it is more natural to associate time to transitions: Transitions represent activities and activities take time. However, authors like Sifakis [16] defend that it is more convenient to associate time to places since this leaves the original firing rule intact, i.e., enabling and firing are instantaneous.

4.1.2 - Type of delay

Independent of the decision on where to put the delay (i.e., transitions, places, or arcs), several types of delays can be distinguished. In this paper, we distinguish stochastic delays. Many of the older timed Petri net models use deterministic delays, i.e., the delay assigned by a transition, place, or arc is fixed. Deterministic delays allow a simple analysis method but have a limited range of applications. In real applications, delays correspond to the duration of activities which are typically variable. Therefore, fixed delays are often less appropriate. There are two ways to describe the inevitable variability. One is to assume constraints on delays (e.g. defining a minimum time for an event to occur), which are non-deterministic delays. Another way is to assume a probability distribution for each delay, in other words, a stochastic delay. Most of the timed Petri net models use stochastic delays. In these models each delay is described by a probability distribution. To make simplify analysis, typically only a restricted set of probability distributions are allowed. Florin & Natkin [17] introduced a SPN model were only exponential delays (i.e., delays described by a negative exponential probability density function) are allowed. Marsan [18] presented a widely used GSPN model that allows for both immediate transitions (i.e., transitions with no delay) and timed transitions (i.e., transitions with exponential delays). Models allowing for arbitrary probability distributions typically defy exact analytical analysis [14].

4.1.3 - Pre-selection versus race semantics

In a classical Petri net a transition is enabled if each of its input places contains enough tokens (typically 1). Only enabled transitions can fire, and firing is instantaneous (i.e., the moment tokens are consumed from the input places, tokens are added to the output places). Transitions are in conflict if they share input places. Firing one transition in conflict with another may disable some or all of these transitions. The choice between several enabled transitions in conflict with each other is solved in a non-deterministic manner. When adding time to a Petri net the enabling and firing rules need to be modified to specify how conflicts are resolved (i.e., the relation between enabling and firing) and whether firing is instantaneous or delayed. Clearly, these two issues are related if we assume that transitions determine the delays and that firing is instantaneous, then it is necessary to associate time to the enabling of a transition. If time is associated to the enabling, there is no need to explicitly define how conflicts are resolved, i.e., enabled transitions “race” against each other and the one that is scheduled to fire first will fire. This firing/enabling semantics are called the race semantics or pre-selection semantics [14].

For example, priorities or probabilities are used to resolve conflicts. In the pre-selection semantics there is no race between enabled transitions: The moment transitions become enabled one of the enabled transitions is selected. For pre-selection semantics the delays can be associated to the firing of a transition or the minimum time a token spends in a place [16]. For race semantics the delays are associated to the enabling time. Note that an enabled transition can be disabled by another transition in case of a conflict. Such a transition loses the race and will not fire. If the transition becomes enabled again, a new race starts. In this new race there are several possibilities for the new enabling time of this transition.

4.1.4 - Capacity, priority, and queuing policy

In a timed net, the capacity of a place or transition is relevant, since places can have a limited capacity restricting the number of tokens it can hold instantly and transitions can limit the number of concurrent firings\enabling of the same transition. Considering a transition with one input place containing three tokens, would this transition be enabled three times under race semantics? Can the transition fire concurrent with itself under pre-selection semantics?

There are three types of capacity related semantics: single server semantics, multiple server semantics, and infinite server semantics [15].

For single server semantics the capacity of a place/transition is 1, for multiple server semantics the capacity of a place/transition is an integer k, and with infinite server semantics there is no capacity restriction. Most timed Petri net models assume infinite server semantics. Several timed net models allow for a priority mechanism, which, if multiple transitions compete for the same token, fires the transition with the highest priority.

4.1.5 - Analysis of timed nets

As indicated in the previous section, there are many ways to introduce time in Petri nets. All timed Petri net models are executable, which means, it is possible to construct a trace of the modeled system by playing the token game. Therefore, simulation can be used to analyze the model. If all non-deterministic measures are replaced by stochastic measures (i.e., delays and conflict resolution), the simulation can be used to obtain confidence intervals for performance measures, such as utilization and throughput. Since simulation does not require difficult mathematical techniques, it is easy to understand for people with a non-technical background. Simulation is also a very flexible analysis technique, since it does not set additional constraints. However, sometimes simulation is expensive in terms of the computer time necessary to obtain reliable results. Another drawback is the fact that (in general) it is not possible to use simulation to prove that the modeled system has the desired set of properties [14].

4.2 - Stochastic timing

The majority of stochastic Petri net (SPN) models use a continuous time domain. In these models, each delay is described by a probability density function. For arbitrary probability density functions, only simulation or approximations are feasible analysis techniques. Therefore, many SPN models impose restrictions on the type of delay distribution.

The GSPN (Generalized Stochastic Petri Net) model extends the SPN model with immediate transitions. Immediate transitions fire without any enabling time and have priority over timed transitions (i.e., transitions with exponential enabling times under the race semantics). A marking is vanishing if an immediate transition is enabled. A marking is tangible if only timed transitions are enabled.

The GSPN model distinguishes between these two types of markings: only tangible markings consume time, in other words, the average waiting time of vanishing states is zero and the average waiting time of tangible states is positive. Note that only the tangible states consume time. Therefore, the vanishing markings are not relevant for most performance measures. GSPN also allows attributing a weight different than 1 to each arc. Each arcs weight defines its value compared to the other arcs and can represent a quantity or the number of times something must be verified.

Mathematically, a stochastic process is a family of random variables {x(t)} defined over the same probability space. Put differently, the values (also called states) that members of the family x(t) can take on, all belong to the same set called the state space of x(t). Examples of stochastic processes are the number of persons on the beach as a function of the time of the day or the number of processes executing on a computer as a function of time. You will come to suspect already that if we can describe the latter mathematically we have made great progress at predicting the behavior of the computer. The classification of stochastic processes (some people also call them random processes) depend on three things: the state space; the nature of the time parameter and the statistical dependencies among the random variables x(t) for different values of the time parameter. [15]

4.3 - Invariants in Petri nets

The reachability set of a Petri net is often drawn as a tree, where the nodes of the tree are markings of the Petri Net. Two nodes M and M0 are connected with a directed arc

iff M[t > M0 for some t Є T

This arc is also labeled with t Є T. The reachability tree can be generated starting with the initial marking of the Place-Transition net and adding directly reachable markings as leaves. Next we proceed with these new markings and determine their directly reachable markings. These markings now become the new leaves of the already generated part of the reachability tree etc. If we reach a marking previously explored we need not continue building the tree any further at that node. Reachability trees can be transformed directly into graphs by removing multiple nodes and connecting the nodes appropriately. Such a graph is called a reachability graph.

The corresponding reachability graph and the reachability tree corresponding to the Petri net example illustrated in Figure 9 are displayed in Figure 10.

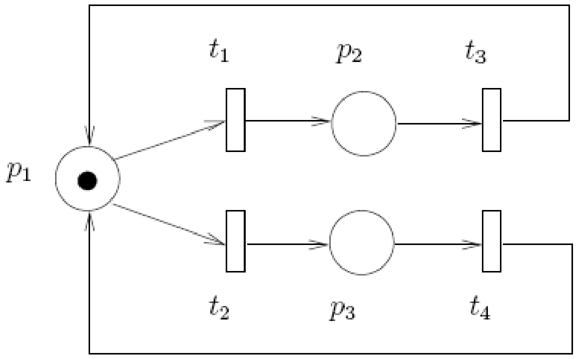


Figure 9 - Petri net example

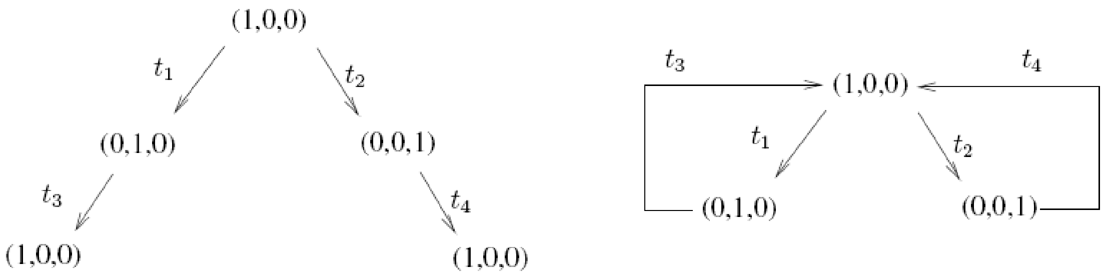


Figure 10 - Reachability tree and corresponding reachability graph

Unfortunately the method of creating the reachability tree by simulating the token game fails in the case of unbounded nets like the one depicted in Figure 11. In order to cope with infinite reachability sets a special symbol ω is introduced for unbounded nets to represent the marking of an unbounded place of the Petri nets model. ω can be referred to as infinity. The arithmetic rules for ω are: for any ‘a’ Є N0: ω + a = ω, ω − a = ω, a < ω, ω = ω [14].

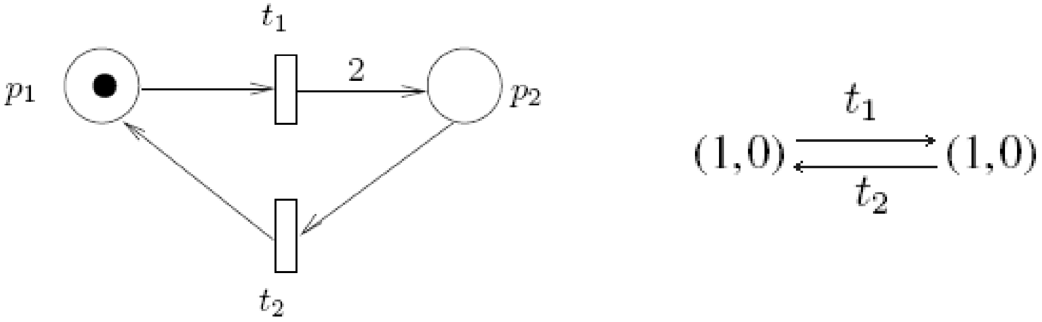


Figure 11 - Petri net model with infinite reachability set; Reachability tree

An alternative approach to the analysis and verification of Petri Nets models is based on the incidence matrix and the state equation [24], which are used to represent dynamic behavior of Petri Nets.

The incidence matrix is an integer (n)x(m) matrix, where n is the number of transitions and m is the number of places, that defines all possible interconnections between places and transitions in a Petri Net.

The state equation for a Petri Net represents a change in the distribution of tokens on places (marking) as a result of a transition firing [24]. This equation is defined as follows:

$$M_k=M_{k-1}+A^T.u_k,\quad K=1, 2\ldots$$

where M_k is a mx1 column vector representing a marking M_k immediately reachable from a marking M_{k-1} after firing the transition *ti*, and A represents the incidence matrix. The *k*-th firing vector u_k is a nx1 column vector that has only one nonzero entry; this entry, a 1 in the *i*-th position, represents a transition *ti* firing in the *k*-th firing of the net, firing sequence starting with an initial marking M_0.

Two concepts related to the incidence matrix are especially useful in studying properties of Petri Net models [24]: the T- and P-invariants. The analysis of P-invariants allows verifying mutual exclusion relationships among functions and resources, and the analysis of the T-invariants allows the identification of work cycles (i.e. the alternative paths to evolve). The quantitative analysis can be performed by means of the simulation of the timed Petri nets models, allowing the verification of the system compliance with specified performance indexes, such as throughput and resource utilization, and the development of optimization strategies.

4.4 – TimeNet software tool

TimeNet is a graphical and interactive toolkit for modeling with extended deterministic stochastic Petri nets (EDSPNs) and stochastic colored Petri nets (SCPNs). TimeNet has been developed at the Real-Time Systems and Robotics group of *Technische Universität* Berlin, Germany (http://pdv.cs.tu-berlin.de/). The project has been motivated by the need for powerful software for the efficient evaluation of timed Petri nets with arbitrary firing delays (TimeNet 4.0 user manual). It provides a user-friendly JAVA graphical interface and is especially tailored to the steady-state analysis of deterministic and stochastic Petri nets. For the class of generalized and stochastic Petri nets, steady-state and transient analysis components are available. Exponentially distributed firing delays are allowed for transitions. Different solution algorithms can be used, depending on the net class. If the transitions with non-exponentially distributed firing delays are mutually exclusive, TimeNet can compute the steady-state solution.

TimeNet uses a remote component based structure which is shown in Figure 12, when simulating SCPNs models. This structure allows a distributed execution of the simulation components consisting of the GUI (user PC), the simulation server, the result monitor, and the database. Each component may run on the local computer or on any other computer in the network.

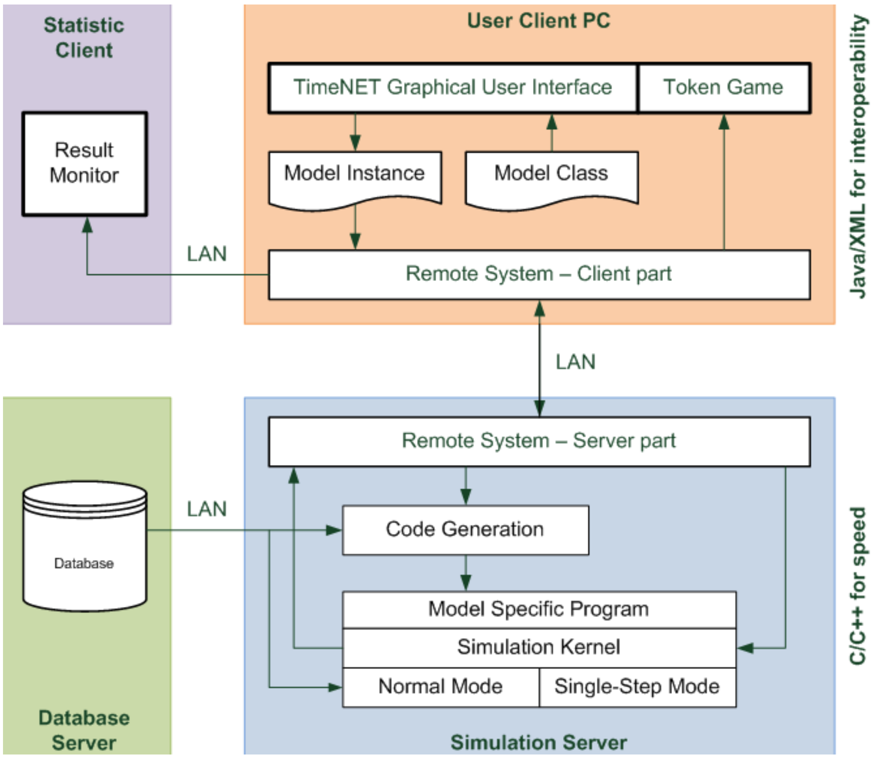
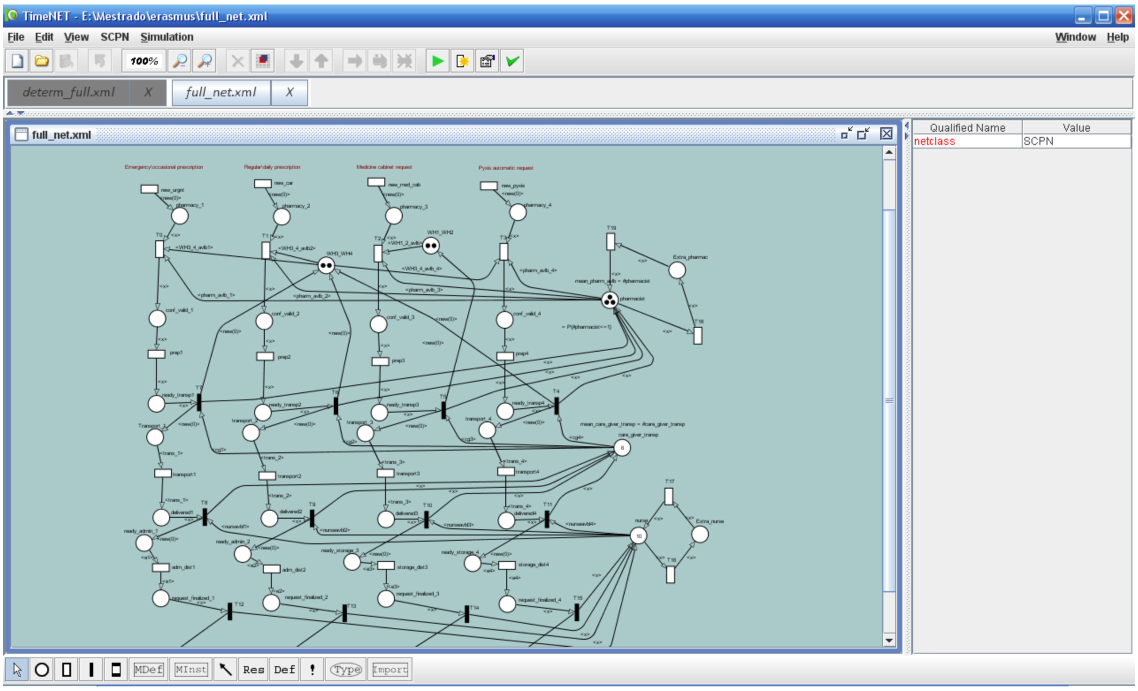


Figure 12 - TimeNet system architecture

The SCPN net class provides a database access to load initial markings and other parameters of a model. In general, the database is not required and should be only used for very complex models.

Most of what happens in the world is not deterministic, even though it might appear so. A computer for example, with a given input always returns the same output. However, while processing data, it is not possible to predict what input values will arrive next or what time sequence they will arrive in. Think of the node of a computer network to understand this. Although the set of messages which may arrive at the node is finite and known, we cannot tell for certain from instant to instant which messages will arrive from where. Moreover, the network software is likely to be using the same processor(s) at the node as the operating system. When the process executing the network software will be interrupted and by which process cannot be said for certain.

All of which makes it impossible to tell for certain what will happen next. The process just described is called stochastic. SCPNs are especially useful to describe complex stochastic discrete event systems and are thus appropriate, e.g., for logistic problems [14].



Qualified Name	Value
netclass	SCPN

Figure 13- Time Net screenshot

5 - Modeling medication distribution processes

In this chapter, the medication distribution processes will be modeled by using Petri nets and particularly by using the TimeNet software tool.

To model the prescription-preparation-administration process, first the different types of prescription procedure where analyzed separately, to make an easier perception of all the steps of the process, and afterwards determine what resources are common to all the process and must be shared. And it was mentioned before in this report (chapter 4), that a place in the net represents a state of the system or the availability of a resource, while transitions represent the occurrence of an event or the execution of an action. Being so, the transitions in the represented procedure models assume that all the conditions for the execution of that action have been verified. For example, when the transition “validation” is enabled, after it fires, it is assumed that the prescription was validated (preparation can begin). Cases when prescriptions are not found valid by the pharmacist are not considered in this study, first because of the very small amount of situations when this happens, and secondly, because it is the first step in the process and if not verified, the time spent on this action can be despised (the time spent on validation are mean values, and not so exact that a minute or two spent with a non-valid prescription should affect the global result of the model). Places represent the accomplished actions like “order ready”, “ready for transport”, or, “order completed”. The performance measures or “watches” are taken in specific points of the net like pharmacist, warehouse and car-giver availability, as well as the number of orders made and the number of orders finalized. By varying the number of available tokens in these critical places (changing resource availabilities), as well as the time delays in the different stages of the procedure, it is possible to verify how the net behaves in different conditions and what results to expect for each particular scenario.

5.1 - Single request type model

The Petri net shown in Figure 14 is a model of the medicine cabinet request procedure. The model contains the steps and conditions mentioned before for this type of order, as well as all the actors or interveners in the process as previously explained, and the respective warehouse or logistic support system needed in this process. Also, the mean values like the average number of pharmacists or care-givers available can be found as a result measure. Tokens in the places that represent resources like the medication warehouses or nurses represent the actual number of those resources available over time.

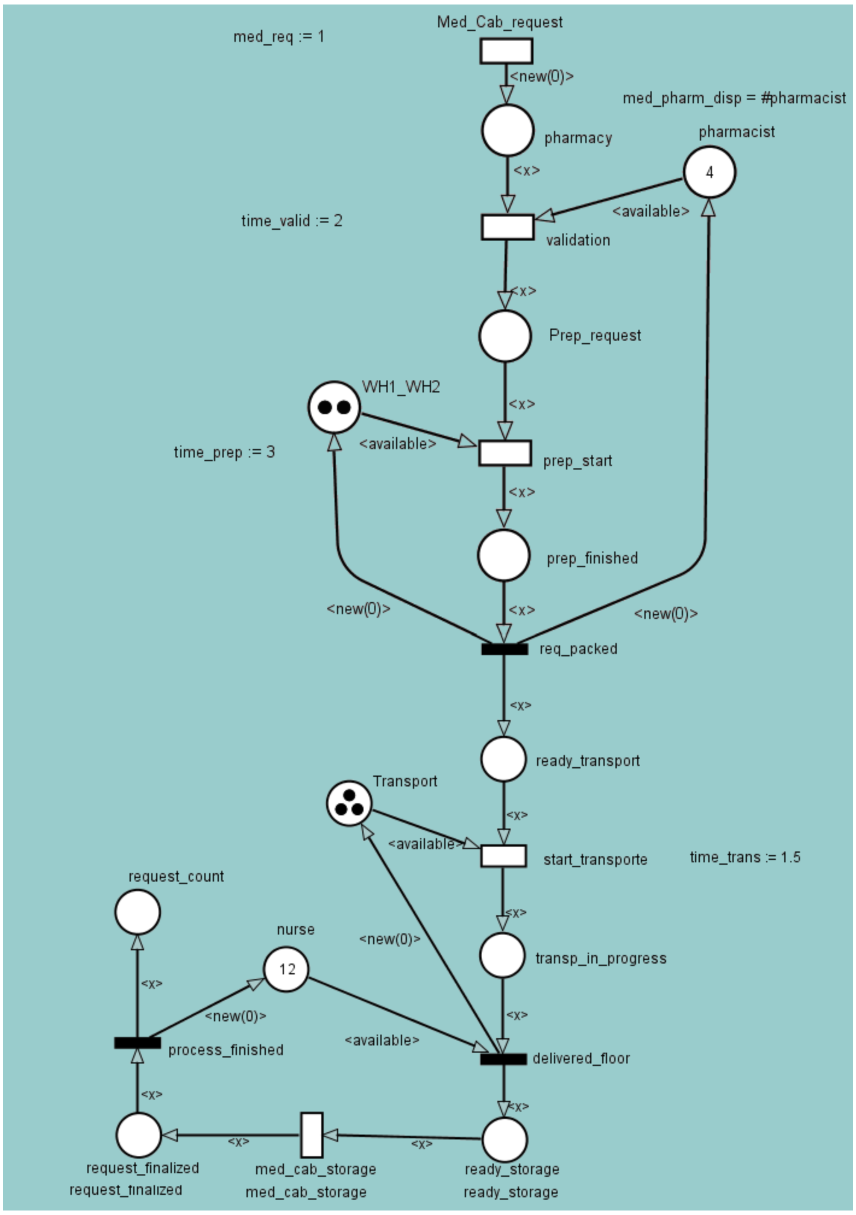
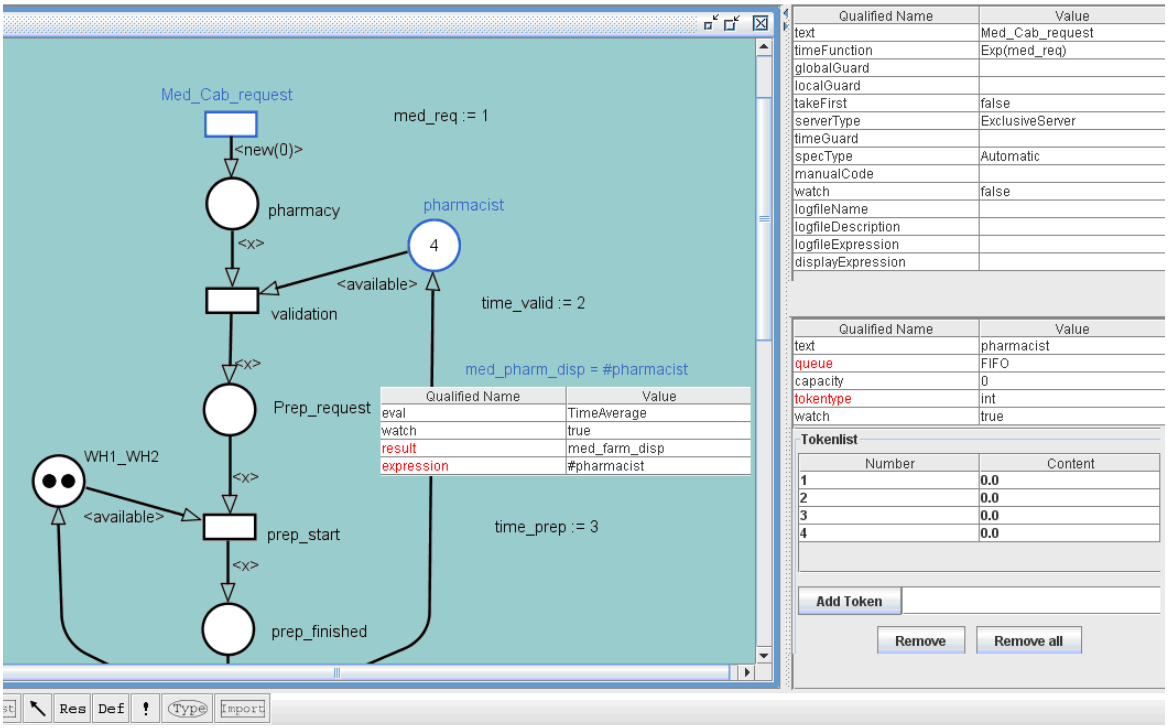


Figure 14 - Medicine cabinet request Petri net model

As shown in Figure 14, the pharmacist resource is released when the request is packed since the pharmacy is a restricted access area, and the medications can be left un-attended until the care-giver is available, but the same care-giver must wait for a nurse to be available to accept the order, since leaving a car or box full of medications with no supervision on any given floor or section is out of the question. The nurse becomes available after administrating the drug(s) to the respective patients or storing them in the medicine cabinet or PSS.

Note that the numbers of tokens in Figure 13 as well as measure definitions (delay times), do not necessarily represent a combination of values used for simulation, they are a mere example of how such time delays and number of tokens can be defined.

In Figure 15 the properties of some of the Petri net element definitions are shown.



Qualified Name	Value
eval	TimeAverage
watch	true
result	med_farm_disp
expression	#pharmacist

Qualified Name	Value
text	Med_Cab_request
timeFunction	Exp(med_req)
globalGuard	
localGuard	
takeFirst	false
serverType	ExclusiveServer
timeGuard	
specType	Automatic
manualCode	
watch	false
logfileName	
logfileDescription	
logfileExpression	
displayExpression	

Qualified Name	Value
text	pharmacist
queue	FIFO
capacity	0
tokentype	int
watch	true

Tokenlist

Number	Content
1	0.0
2	0.0
3	0.0
4	0.0

Figure 15 - Example definition of places, measures and tokens

First, the timed transition *Med_Cab_request* creates timed tokens simulating the request arrivals at the pharmacy, whose delay function is exponential. The expected value for the delay is defined by the *med_req* parameter, which makes easier to change this value from simulation to simulation if wanted.

The *‘pharmacist’* place definition, has the option to choose the queue type, the place capacity, the token type and the watch option, which if true returns the number of tokens over time in that place.

Tokens are attributed to a place by adding them to the token list. Also presented here is the performance measure *med_farm_disp*, which calculates the mean number of tokens; in this case, in the place ‘pharmacist’ over time. This type of measure, watch and definition is used to facilitate the changing of parameters for different simulations and to follow the state of the places (information over time) throughout the simulation.

The availability of a resource like a pharmacist or the medication automated dispensing machine and\or shelves (referred to as WH1 and WH2), is confirmed by the places that define them and the tokens present in that place. As an example illustrated in Figure 16, the transition *‘validation’* will become enabled if there is at least one token in the ‘pharmacist’ place, and after the delay time the transition will fire. After firing the transition consumes both tokens, and only “returns” the ‘pharmacist’ token after the order is packed, by creating a new token at the transition ‘*req_packed*’, making the pharmacist available to start a new request preparation, and putting a token in the *‘ready_transport’* place to continue the process. This same type of confirmation is made for the pharmacists, the care-givers or transporters, the nurses and the medication warehouses. This confirmation, since it’s present to guarantee resource availability can in some cases represent a waiting time, if the resource is not available, the transition will not be enabled and the process delay time does not pass (the timed delay only begins after the transition is enabled with tokens in all its input places).

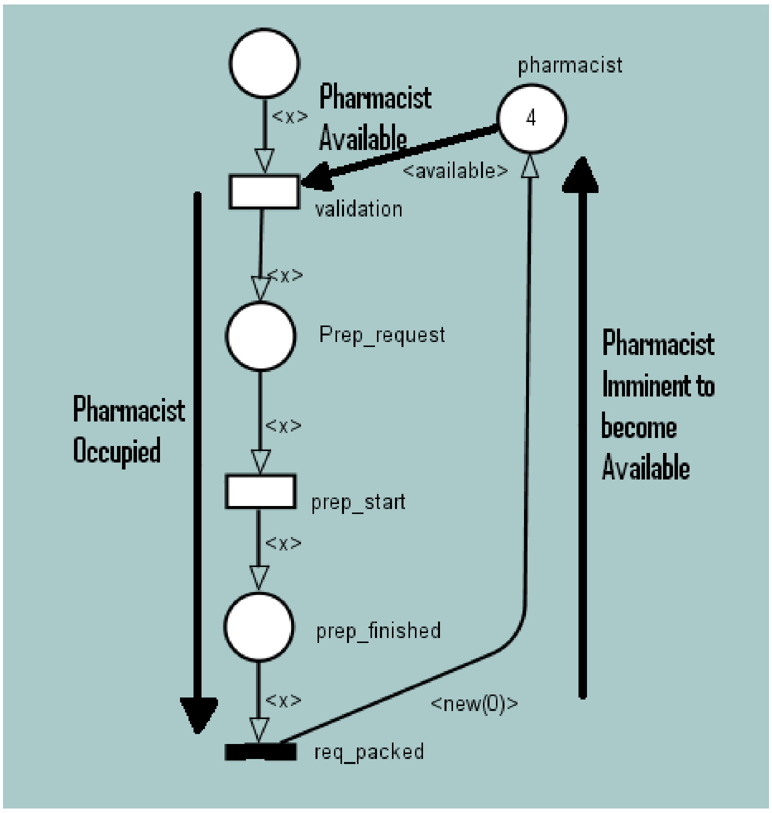


Figure 16 - Example of resource availability check

The models for the remaining three prescription\request procedures are very similar to this one, with the differences being in the time delays, in some cases like the emergency prescriptions, different transport procedures or in other cases, the type of preparation for transport. All the features described in this model are also used in the other request types in the same way, for obtaining token information over time and defining performance measures to facilitate the variation of these parameters.

Each of the sequences starts with a timed transition (with different delay times) that “injects” (creates) a token in that sequence, simulating a new order, and the respective process begins. After that, comes the confirmation of availability from the warehouses as well as the pharmacist availability, to then proceed to validate and confirm the prescription. After the order is prepared and packed (ready for transport), another wait (confirm availability) holds the development of the procedure until a care-giver is available to transport the order. After the transport time ends, supposedly the medications are delivered to the floor or area of the hospital where they were ordered, and wait (only if necessary, i.e. no tokens available) for a nurse to accept the delivery and begin the distribution and administration process, and making the nurse available again from a resource point of view by creating a new token for the nurse place (returning the one that enabled the transition that started distribution and administration). All four types of orders follow this basic process or procedure, although depending on the type, a specific form of transportation or package may be used.

5.2 – Multiple request model

The model that describes all four medication request types is presented in Figure 17. The resources that are common to all processes or at least more than one, must be shared, which means that the availability of these resources must be shared by the distinct processes and waiting times may be forced in some cases by the non-availability. Although, this wait is automatically created and resolved by itself (when the resource is not available the transition is not enabled. When it becomes available, the process is immediately resumed and the transition that was waiting for a token is enabled and its delay time begins counting).

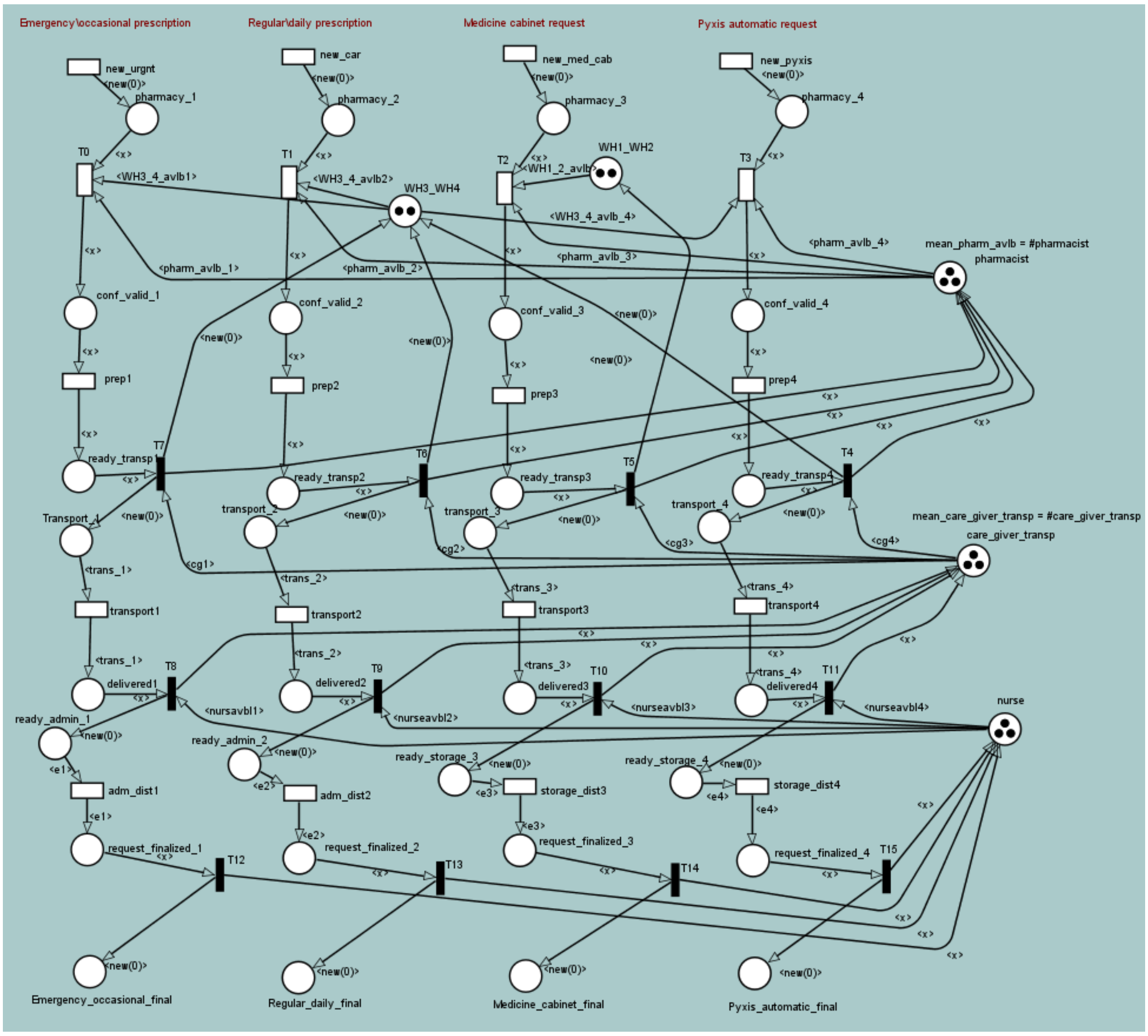


Figure 17 - Model describing full distribution system

The model illustrated in Figure 17 may appear to be confusing due to all the lines crossing through the middle to confirm resource availability, but in fact it is quite simple and similar to the medicine cabinet request model presented previously. The procedure is basically linear in terms of individual orders, but in terms of general resources (pharmacists, care-givers and nurses), there is an availability issue that must be solved with the corresponding priorities for the different process types. This translates into the non-relevance of the invariants, since the processes are unit-directional and once one begins there is only one possible path for it to follow. Therefore the knowledge extracted from the net will not be through its place or transition invariants, but from the evolution graphs returned by simulating different scenarios of resource availability and possibly the time consumed by the process.

Although the use of TimeNet software tool simplifies the simulation of different possibilities of process timing, the referenced values for the simulation are expected values and with exponential timing delays, the event distribution over time will be random but in the range of the "expected" value. These time delays are very relative not only concerning the prescription\request frequency but also the time the entire subsequent processes take like preparing or transporting the order, thus the exponential delay time distribution to simulate the reality of the unpredictability ruling in the entire system. Since the resources are common to all four types of procedure, the full net which converges all the partial distribution models into one big model, is really the main point of interest for simulating the prescription-preparation-distribution process.

5.3 - Token game

The TimeNet software also allows the user to follow the "token game" along the net, i.e. the path that tokens follow along the process depending on the availability and priorities. This allows the user to observe, in a visual manner, the behavior of the Petri net or a specific part of the process when tokens are in circulation.

The token game function basically shows the token evolution, or the net evolution step-by-step as the places and transitions are "token" or fired respectively. As transitions become enabled (with tokens in all its input places), they pass tokens along the net as defined by the output arc's, and it is possible to see the different elements that become enabled or disabled with the net evolution, the relation between elements that interact like availability confirmations (as described and shown previously in Figure 16).

The token game becomes helpful in situations when the user wants to observe close up how the net evolves and behaves in certain precisely defined situations, which cannot happen in the simulation process since the simulation is done as an internal process of TimeNet and only the end values over time are given, unlike in the token game that allows the user to analyze step-by-step or token-by-token a particular situation of the net.

The Figures 18, 19, 20, 21 and 22 illustrate some steps of a token game will be presented as an example of how this feature allows the user to understand the natural evolution of the net, in this case until the tested sequence repeats itself, or the net assumes the same conditions of the initial state.

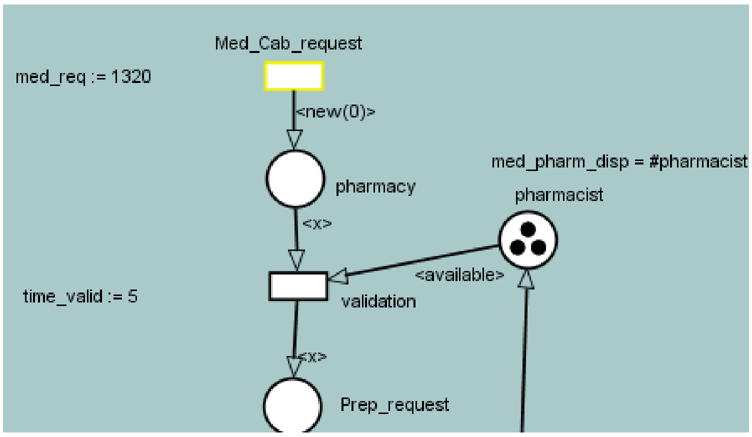


Figure 18 - Token game part 1

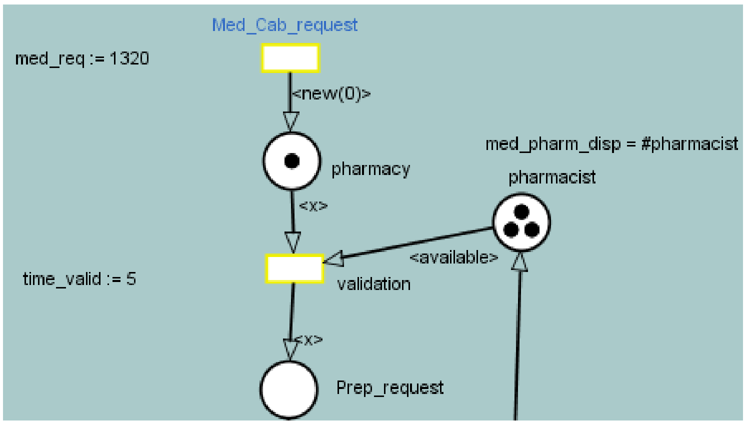


Figure 19 - Token game part 2

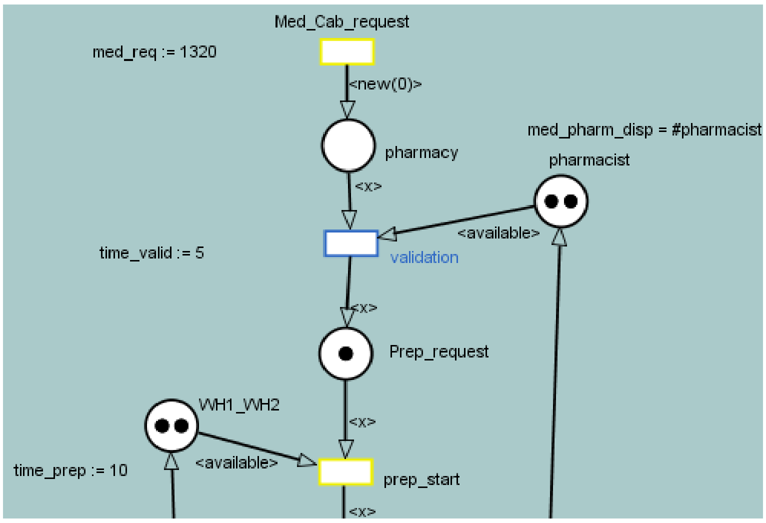


Figure 20 - Token game part 3

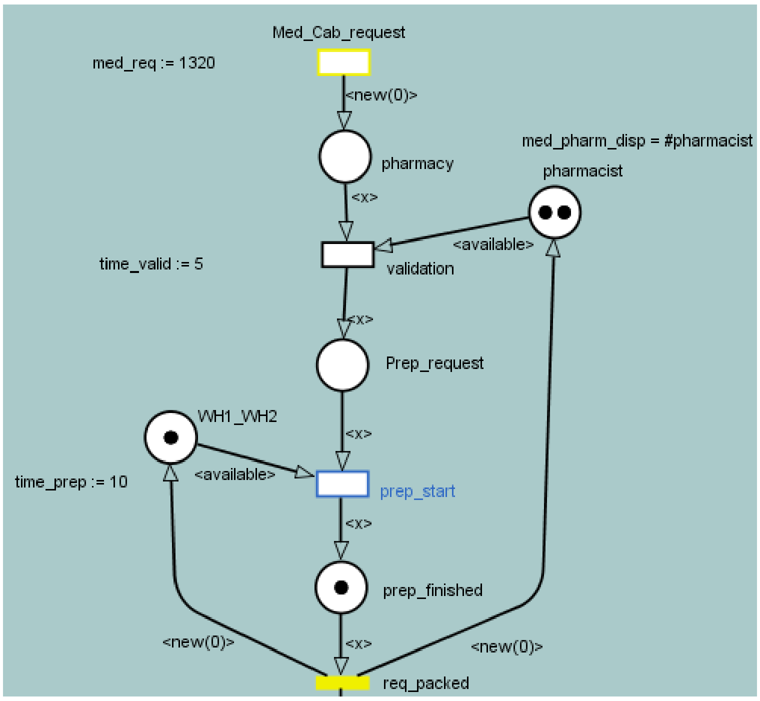


Figure 21 - Token game part 4

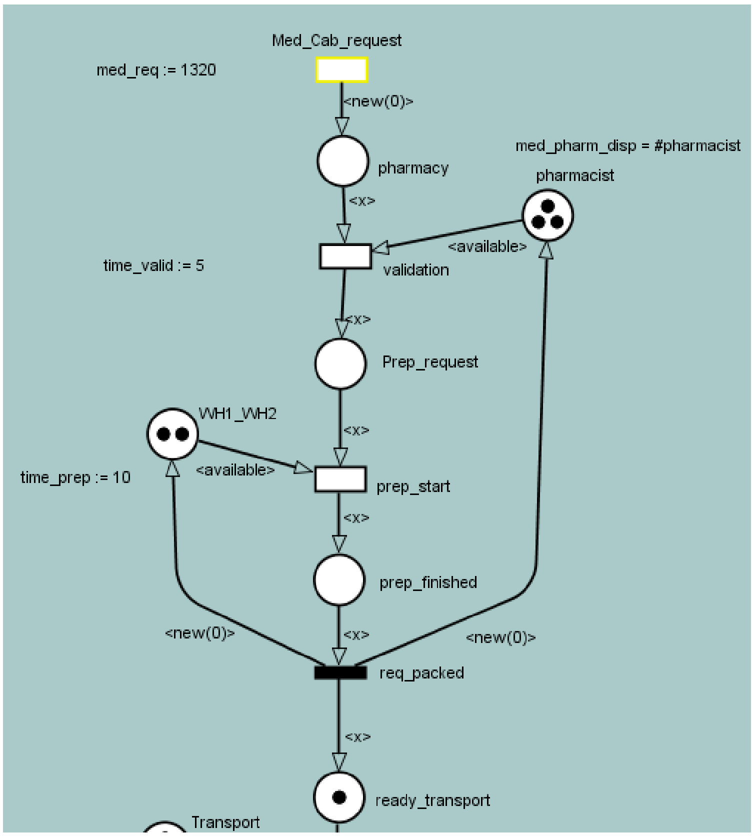


Figure 22 - Token game part 5

6- Simulation of the Designed Models

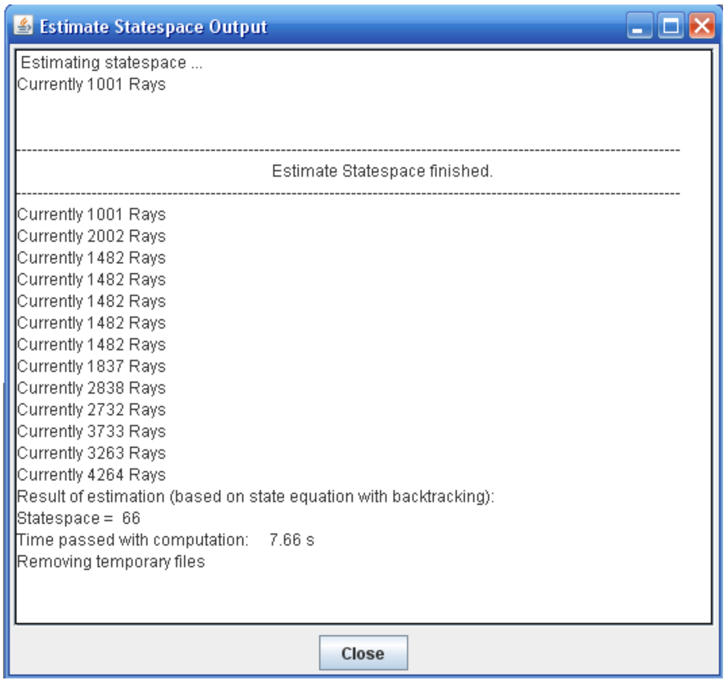
In this chapter, the designed Petri nets models for the medication distribution process will be simulated by using the TimeNet software tool. The simulation of Petri nets models in TimeNet returns data depending on functions or counting on numerous types of time delays, counting functions and watches on either transitions or places, which allow the user to gather information on the state of the net in a certain time interval of the simulation.

6.1- Deterministic simulation of the Petri net models

As mentioned before, the simulations that will return state graphs with information about the net over time are made with deterministic colored Petri net models using the TimeNet software. However, to validate the Petri net model and check its structure for traps and siphons as well as place invariants, it is necessary to use a deterministic net for the software to return this information.

The structure of the Petri net is the same as presented in chapter 5.2, though different information is given it is possible to simulate the same conditions with the number of resources varying. For example, it is possible to maintain all the net resources and conditions but simulate what happens for different values of pharmacists dedicated to the process. It is also possible to calculate for example the probability of the number of tokens in a place being larger or smaller than a given value.

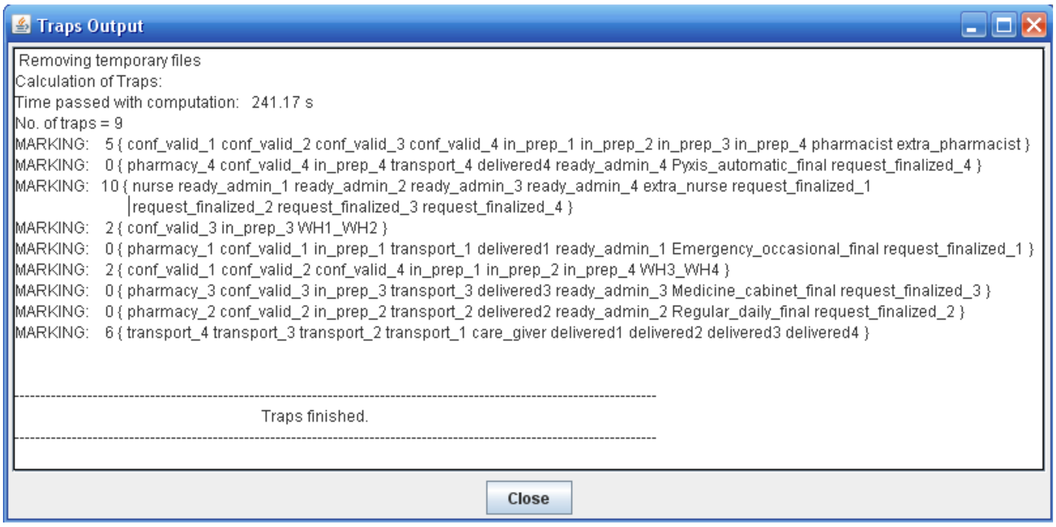
The *Estimate Statespace* function of the TimeNet tool computes an estimation of the number of reachable states that the current model has, based on the structure of the model [19]. Figure 23 shows an example of a monitor window with the result output for the full net model.



```
Estimating statespace ...
Currently 1001 Rays

--------------------------------------------------------------------------------------------------------------------------
                              Estimate Statespace finished.
--------------------------------------------------------------------------------------------------------------------------
Currently 1001 Rays
Currently 2002 Rays
Currently 1482 Rays
Currently 1482 Rays
Currently 1482 Rays
Currently 1482 Rays
Currently 1482 Rays
Currently 1837 Rays
Currently 2838 Rays
Currently 2732 Rays
Currently 3733 Rays
Currently 3263 Rays
Currently 4264 Rays
Result of estimation (based on state equation with backtracking):
Statespace =  66
Time passed with computation:     7.66 s
Removing temporary files
```

Figure 23 - Statespace result window

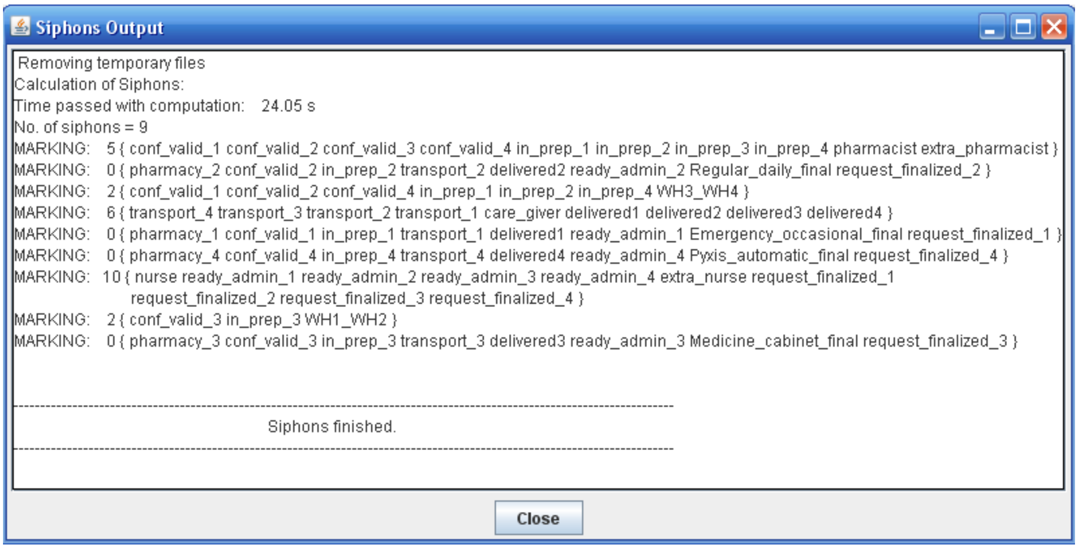
The *Traps* function computes the set of minimal traps (i.e. place sets that will never become unmarked in any subsequent marking after they are once marked) [19]. Figure 24 shows an example. Every trap is described by the corresponding places and their initial marking.



```
Removing temporary files
Calculation of Traps:
Time passed with computation:   241.17 s
No. of traps = 9
MARKING:   5 { conf_valid_1 conf_valid_2 conf_valid_3 conf_valid_4 in_prep_1 in_prep_2 in_prep_3 in_prep_4 pharmacist extra_pharmacist }
MARKING:   0 { pharmacy_4 conf_valid_4 in_prep_4 transport_4 delivered4 ready_admin_4 Pyxis_automatic_final request_finalized_4 }
MARKING:  10 { nurse ready_admin_1 ready_admin_2 ready_admin_3 ready_admin_4 extra_nurse request_finalized_1
               request_finalized_2 request_finalized_3 request_finalized_4 }
MARKING:   2 { conf_valid_3 in_prep_3 WH1_WH2 }
MARKING:   0 { pharmacy_1 conf_valid_1 in_prep_1 transport_1 delivered1 ready_admin_1 Emergency_occasional_final request_finalized_1 }
MARKING:   2 { conf_valid_1 conf_valid_2 conf_valid_4 in_prep_1 in_prep_2 in_prep_4 WH3_WH4 }
MARKING:   0 { pharmacy_3 conf_valid_3 in_prep_3 transport_3 delivered3 ready_admin_3 Medicine_cabinet_final request_finalized_3 }
MARKING:   0 { pharmacy_2 conf_valid_2 in_prep_2 transport_2 delivered2 ready_admin_2 Regular_daily_final request_finalized_2 }
MARKING:   6 { transport_4 transport_3 transport_2 transport_1 care_giver delivered1 delivered2 delivered3 delivered4 }


--------------------------------------------------------------------------------------------------------
                              Traps finished.
--------------------------------------------------------------------------------------------------------
```

Figure 24 - Traps result window

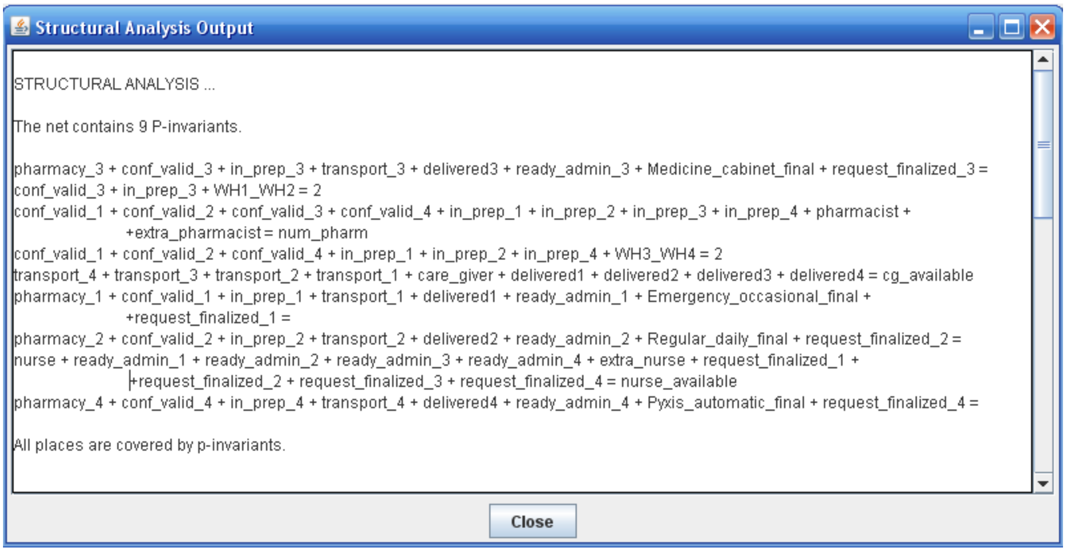
The *Siphons* function computes the set of minimal siphons (i.e. place sets that will never become marked again in any successive marking after they become unmarked). The output of this command (Figure 25) is similar to the one of Traps [19].



```
Removing temporary files
Calculation of Siphons:
Time passed with computation:   24.05 s
No. of siphons = 9
MARKING:   5 { conf_valid_1 conf_valid_2 conf_valid_3 conf_valid_4 in_prep_1 in_prep_2 in_prep_3 in_prep_4 pharmacist extra_pharmacist }
MARKING:   0 { pharmacy_2 conf_valid_2 in_prep_2 transport_2 delivered2 ready_admin_2 Regular_daily_final request_finalized_2 }
MARKING:   2 { conf_valid_1 conf_valid_2 conf_valid_4 in_prep_1 in_prep_2 in_prep_4 WH3_WH4 }
MARKING:   6 { transport_4 transport_3 transport_2 transport_1 care_giver delivered1 delivered2 delivered3 delivered4 }
MARKING:   0 { pharmacy_1 conf_valid_1 in_prep_1 transport_1 delivered1 ready_admin_1 Emergency_occasional_final request_finalized_1 }
MARKING:   0 { pharmacy_4 conf_valid_4 in_prep_4 transport_4 delivered4 ready_admin_4 Pyxis_automatic_final request_finalized_4 }
MARKING:  10 { nurse ready_admin_1 ready_admin_2 ready_admin_3 ready_admin_4 extra_nurse request_finalized_1
               request_finalized_2 request_finalized_3 request_finalized_4 }
MARKING:   2 { conf_valid_3 in_prep_3 WH1_WH2 }
MARKING:   0 { pharmacy_3 conf_valid_3 in_prep_3 transport_3 delivered3 ready_admin_3 Medicine_cabinet_final request_finalized_3 }



--------------------------------------------------------------------------------------------------------------------------
                                    Siphons finished.
--------------------------------------------------------------------------------------------------------------------------
```

Figure 25 - Siphons result window

The *Structural Analysis* function obtains the minimal place invariants of the model and extended conflict sets of immediate transitions, showing them in two windows. A place invariant (or semi flow) is informally a set of places for which a weighted sum of tokens remains the same for any reachable marking of the Petri net [19] as shown in Figure 26.



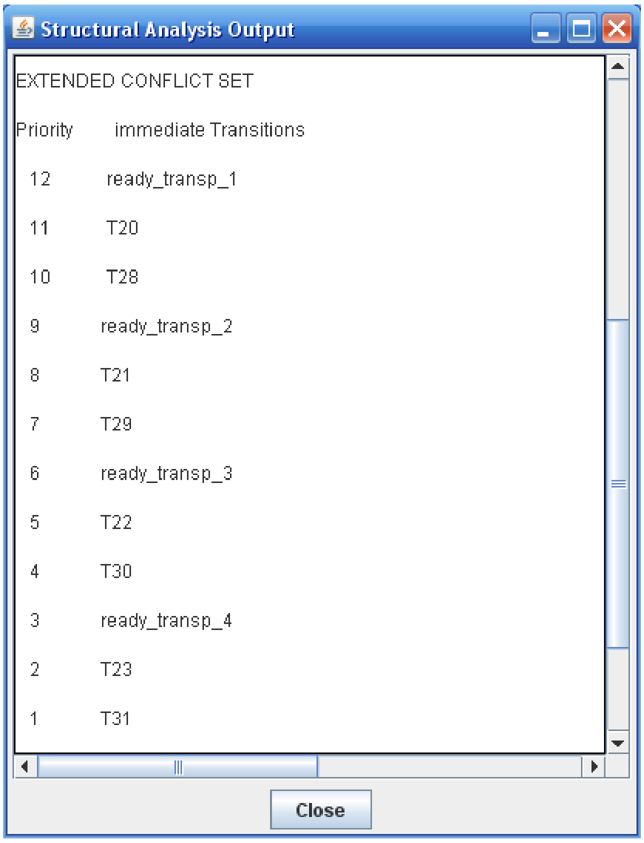
```

STRUCTURAL ANALYSIS ...

The net contains 9 P-invariants.

pharmacy_3 + conf_valid_3 + in_prep_3 + transport_3 + delivered3 + ready_admin_3 + Medicine_cabinet_final + request_finalized_3 =
conf_valid_3 + in_prep_3 + WH1_WH2 = 2
conf_valid_1 + conf_valid_2 + conf_valid_3 + conf_valid_4 + in_prep_1 + in_prep_2 + in_prep_3 + in_prep_4 + pharmacist +
          +extra_pharmacist = num_pharm
conf_valid_1 + conf_valid_2 + conf_valid_4 + in_prep_1 + in_prep_2 + in_prep_4 + WH3_WH4 = 2
transport_4 + transport_3 + transport_2 + transport_1 + care_giver + delivered1 + delivered2 + delivered3 + delivered4 = cg_available
pharmacy_1 + conf_valid_1 + in_prep_1 + transport_1 + delivered1 + ready_admin_1 + Emergency_occasional_final +
          +request_finalized_1 =
pharmacy_2 + conf_valid_2 + in_prep_2 + transport_2 + delivered2 + ready_admin_2 + Regular_daily_final + request_finalized_2 =
nurse + ready_admin_1 + ready_admin_2 + ready_admin_3 + ready_admin_4 + extra_nurse + request_finalized_1 +
          +request_finalized_2 + request_finalized_3 + request_finalized_4 = nurse_available
pharmacy_4 + conf_valid_4 + in_prep_4 + transport_4 + delivered4 + ready_admin_4 + Pyxis_automatic_final + request_finalized_4 =

All places are covered by p-invariants.
```

Figure 26 - Structural analysis result window P-invariants

In this case, the traps, the siphons and even the place invariants are all groups of places that correspond to the nine different internal loops present in the net. It is understandable and easy to see that each type of prescription\request creates a type of infinite “loop” where an order starts the process and when it ends another one will eventually restart the process and repeat all the steps. This repetition happens not only for all four request types, but also for the other internal processes like the warehouse availability (two warehouses), and the human resource availability for the pharmacists, the care givers and the nurses (three more “loops”), hence the nine traps, invariants and p-invariants.

One other conclusion reached with the help of Figure 26 is that since all the places in the net are covered by p-invariants, the net is limited, since all the places present have a limited number of tokens (value of the p-invariant). The traps and siphons in a very similar way represent the same cycles within the net and also the possible number of tokens that can be in the cycle or loop.

Figure 27 shows the extended conflict set (ECS), the second output of the structure check. An ECS is a set of immediate transitions, obtained by the transitive closure of transitions that are in structural conflict. This is important for the specification of firing probabilities, because they are relative to the other transitions in the same ECS.

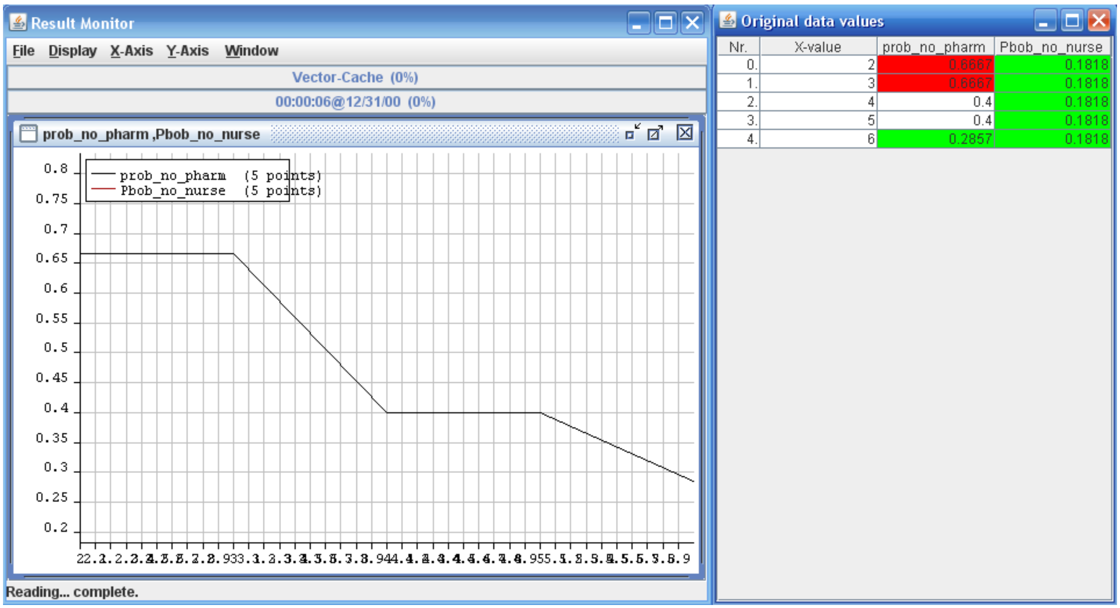


```
EXTENDED CONFLICT SET

Priority      immediate Transitions

 12          ready_transp_1

 11          T20

 10          T28

 9          ready_transp_2

 8          T21

 7          T29

 6          ready_transp_3

 5          T22

 4          T30

 3          ready_transp_4

 2          T23

 1          T31
```

Figure 27 - Extended Conflict Set result window

Priorities of immediate transitions must be adjusted to put transitions into different ECS, because transitions with differing priorities cannot be in conflict with each other and will therefore not be in the same ECS. It may be necessary to check the ECS and adjust the priorities also in the case of confusions, which are detected and notified by the structural analysis prior to the performance analysis algorithms [19].

Figure 28 shows the graph that represents the probability of there being only one or no pharmacist at all available, for different values of initial pharmacist availability. As the simulated number of pharmacists gets larger, the probability of there not being one available gets smaller. In this case (varying the number of pharmacists), the probability of having no nurses available does not change, and is so small that it is not represented in the scaled graph. The data values however are given in a table with the corresponding simulation values for each variable.

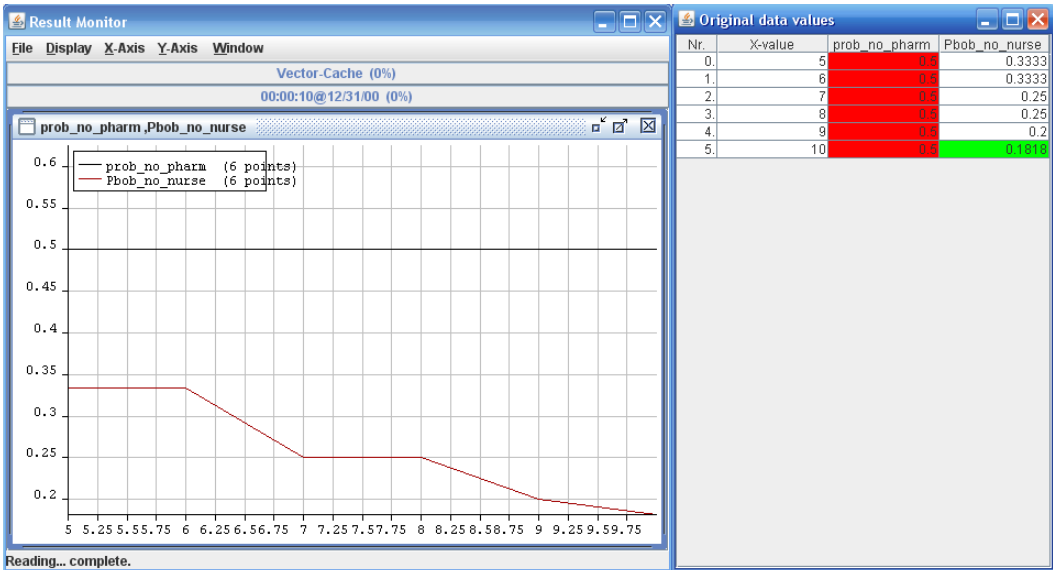


Nr.	X-value	prob_no_pharm	Pbob_no_nurse
0.	2	0.6667	0.1818
1.	3	0.6667	0.1818
2.	4	0.4	0.1818
3.	5	0.4	0.1818
4.	6	0.2857	0.1818

Reading... complete.

Figure 28 - Probability graph varying the number of pharmacists

The number of nurses available and the probability of there being no nurses depending on how many are dedicated to the process can also be simulated. What the software does is a series of simulations with different values for the nurse available resource or place, and for each simulation, calculates the probability of there being X amount of nurses available and represents them (the probabilities) in a graph. In the case of Figure 29, the simulated value for available nurses was between five and ten.



Nr.	X-value	prob_no_pharm	Pbob_no_nurse
0.	5	0.5	0.3333
1.	6	0.5	0.3333
2.	7	0.5	0.25
3.	8	0.5	0.25
4.	9	0.5	0.2
5.	10	0.5	0.1818

Reading... complete.

Figure 29 - Probability graph varying the number of nurses

6.2 - Simulation of the designed models

This section presents the variables used for the simulation process as well as the respective token information and state-graphs returned by the simulation process for the different simulation variables and for each of the models. The main objective of simulation is to be able to find the border values for the resources that allow the completion of the processes without causing waiting periods or unwanted delays. Therefore in this study different situations were considered for exemplifying the simulation process and the different variables that can be adjusted to recreate distinct scenarios and obtain information in the form of state graphs from each place, transition or measure defined in the net. Note however that all the performance measures as well as the watches on transitions or places have the possibility of an enormous number of different combinations, and being so only a few will be presented.

To begin the simulation, it was necessary to consider the expected time between orders as well as the delay in each stage of the process. These times are an average estimate time made by the director of the pharmacy at HdRO. Since the type of delay being used is exponential, the actual number of times a process (transition) is enabled as well as the delay in the process preparation can (and will) vary from simulation to simulation, recreating the unpredictability of the process.

6.3 - Medicine cabinet model and result graphs

For the first pair of simulations the expected values represented in Table 1 were used to define the exponential time delays, as well as the initial values for the resource's availability represented in Table 2. Simulating twice with the same values allows the user to observe the difference between two simulations with the same variables. Afterwards another simulation can be made with parameters adjusted to the results obtained, i.e. looking to minimize the needed resources and make the process more efficient.

Med.Cab.Req	Validation	Prep_start	Start_transporte	Med_Cab_Storage
1320	5	10	8	15

Table 1

The initial values used for simulation concerning the available resources where:

Pharmacist	Warehouse	Transport	Nurse
3	2	3	5

Table 2

The results returned by simulating the model with the presented values are shown in Figure 30, and even though as mentioned, all elements of the net can be "watched" for simulation, only the graph results found relative will be presented and explained. As referred previously, two simulations where made with the same resource and delay values, not only to have two possible situations with the same amount of resources and delays but also to verify how the exponential time delay varies around the respective expected value defined for the transition.

The evolution of the transition referred to as "med_cab_request" represents the number of requests made over time (total time simulation=10000, considering 1 time unit=1 real minute) for approximately one week. Despite the differences between the simulations, it is easily perceptive that the most significant time distribution is the request at the beginning of the model, making all the other distributions over time graphically very similar. It is easy to understand that the initial time delay sets a "pace" for the model, defining when evolutions in the system occur. Apart from a small advance in time we can see that following transitions have almost identical distributions over time. The place references tell the user how many resources are available over time in each place. For both simulations displayed in Figure 30 the resources where clearly always available, being so, a new simulation will be made, using the same delay times but a smaller number of resources available, to observe what happens to the net conditions in a different scenario.

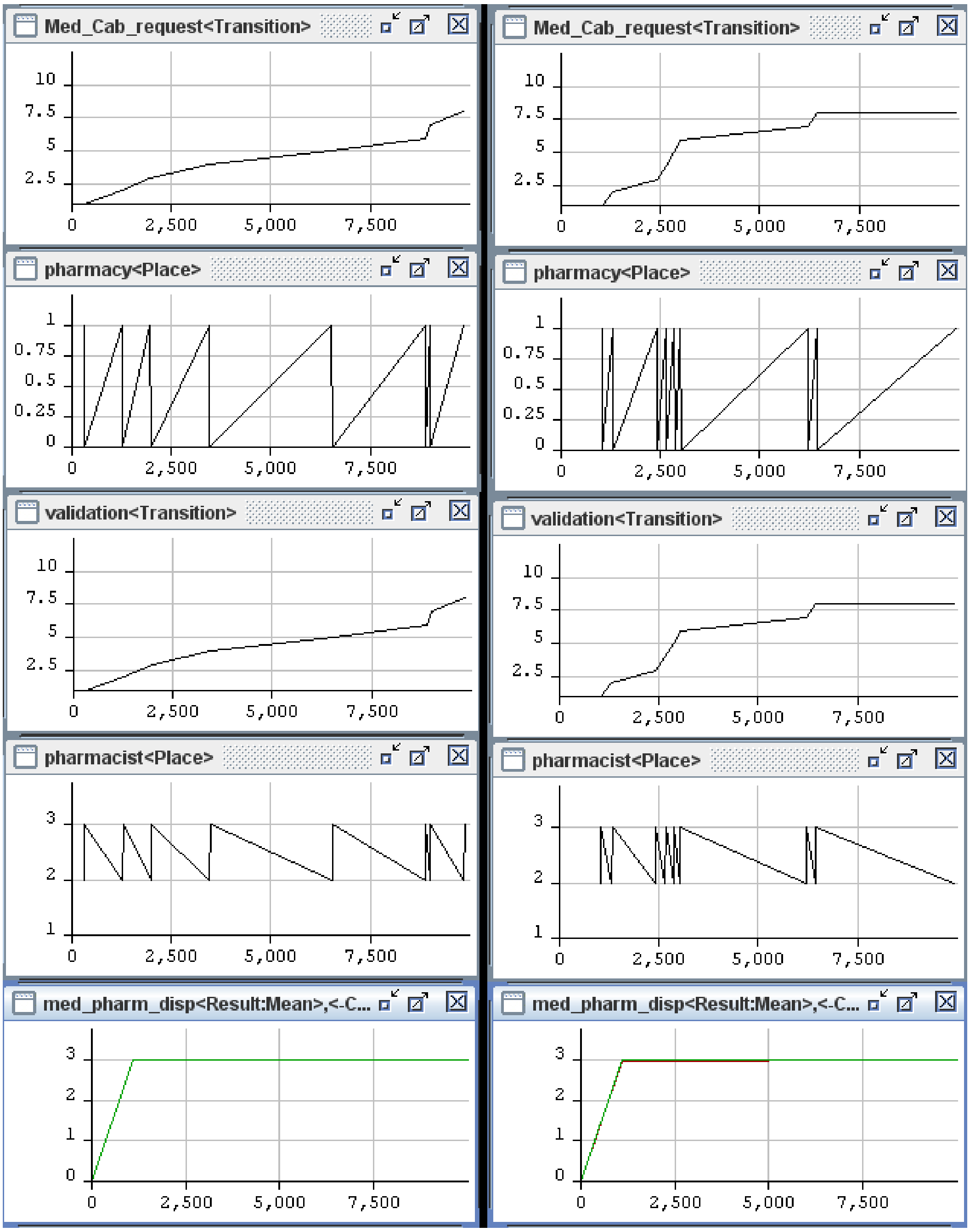


Figure 30 - Graphs from first pair of simulations

In both simulations, the mean token value in the pharmacist place was 2.9, which means that the three pharmacists initially defined for the simulation, where practically always available. From viewing the pharmacist place graph, it is obvious that only one was used, since the state graph is always between three and two (available) pharmacists.

So the main conclusion reachable with these two simulations is that for just the medicine cabinet request, the amount of resources deployed was more than enough. Being so, another simulation was made with minimum values for these resources to observe what happens to the model. The results are illustrated in Figure 31.

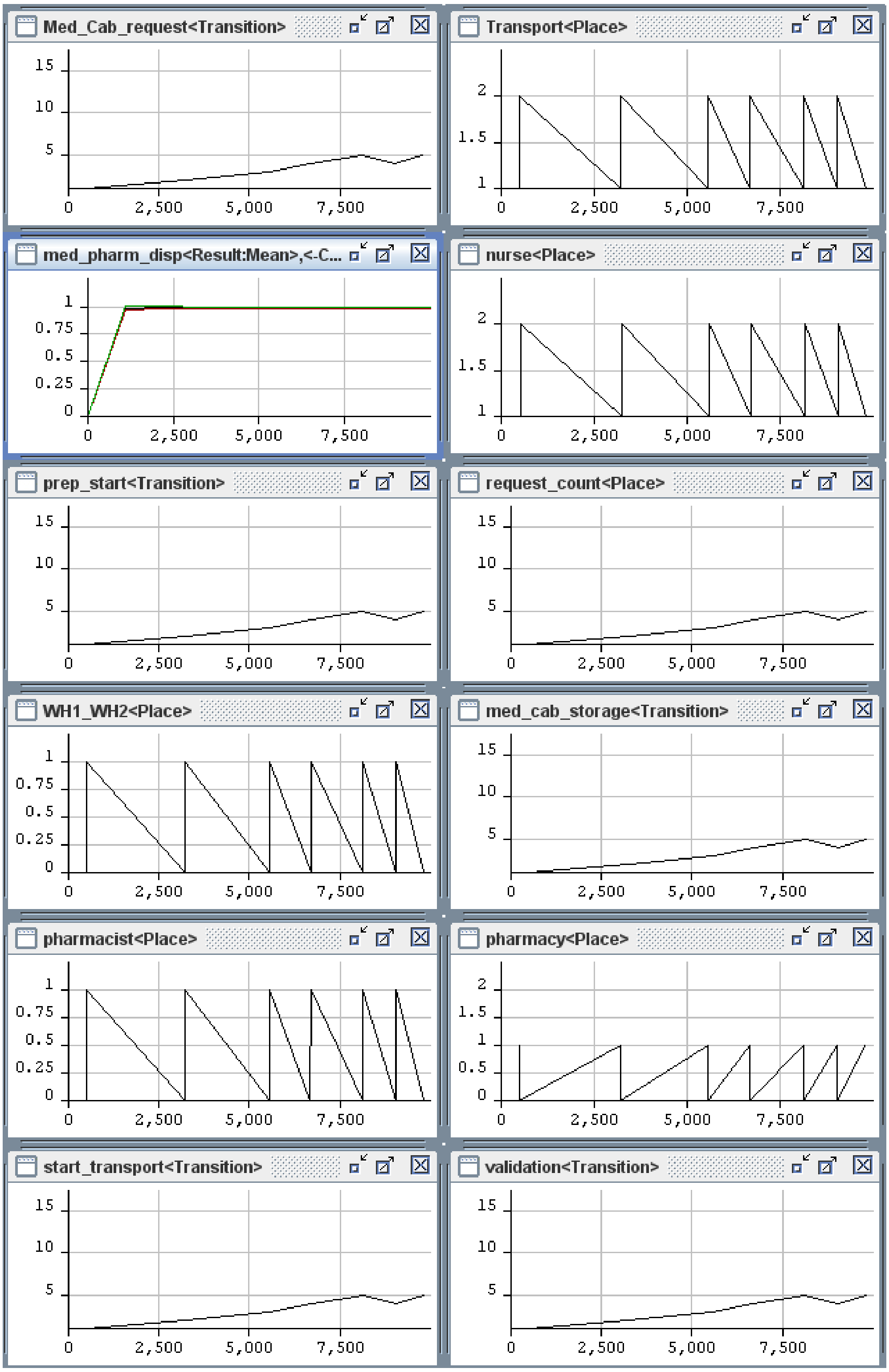


Figure 31 - Simulation with new resource availability

The global information given by these graphs can be very useful do determine whether or not there are enough people working in a station. For instance, in this simulation, fewer requests were made, but it doesn´t necessarily mean that the model was less effective since the exponential time distribution can make this difference happen. However, observing the place for the pharmacist we can see that as soon as it (pharmacist as a resource) becomes available it is needed in a new process. This could mean that there is a waiting period for the resource, although when looking at the pharmacy resource it is observable that when a token arrives (new request), the pharmacist becomes occupied. What can be concluded from the fact that when the pharmacy has a token the pharmacist (place) looses one, is that every time a request is made, there is a pharmacist available to attend the request. The nurses and care givers (transport resource) also vary between two and one, indicating that in this particular situation one of each resource would suffice to complete the procedure. Another interesting piece of information that can be extracted from these graphic results is the time between the first token starting the process and finishing it that will also describe the time used by the simulation for one request from beginning to end.

Figure 32 shows the first time event verified by the initial transition starting the process and by the last place in the model, making it possible to calculate how long the process took.

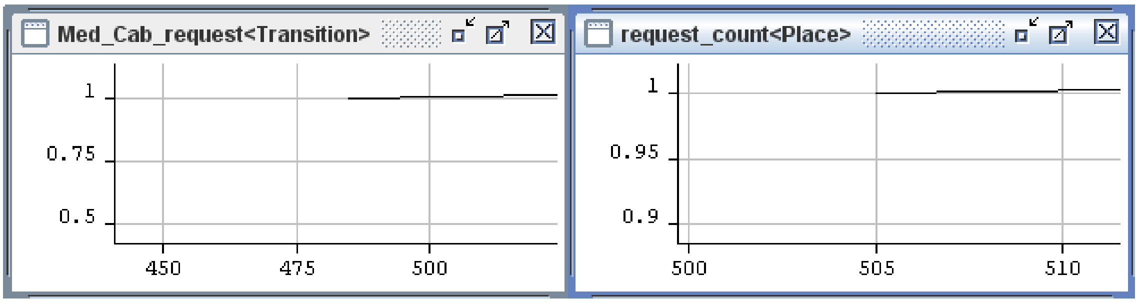


Figure 32 - Time between the beginning & end of the request

In Figure 32, it is possible to verify that the first token took approximately 20 time units from the beginning to the end of the request (first request started at 485 and finished at 505). The last simulations presented refer to only one part of a larger process that is the entire drug distribution system from the pharmacy to the patient. Since resources are common and must be shared, single model simulations will not represent the reality of the entire prescription-preparation-distribution process. For this, the entire system must be modeled and simulated as a whole and with the resource sharing as described in section 5.1, Figure 14.

6.4 - Full distribution system model and result graphs

This section will present the results obtained through the simulation of the full medication distribution model presented in chapter 5, Figure 17. Once again, all the transitions and places present in the net can be monitored by using a “watch” that graphically shows the behavior of the place or transition over time. The performance measures defined as well as the watches on places and transitions return information on the state of the net over time through graphs representing the number of tokens or enabling of transitions over time. Due to the large number of elements present in the model, the only graphs that will be presented are the ones found relevant to the subject being approached and explained. Much like the medicine cabinet request model net, the time distribution will be exponentially distributed around an expected value for each delay.

In this case and unlike the single request model made to describe the medicine cabinet request procedure, all the “actors” involved in the process as well as all four types of medication prescription\request processes are present in the net and “share” or wait for necessary resources to become available, as shown in Figure 33. This allows the study of the net and the combination of different resources with all the staff personnel, making it possible to simulate different scenarios of availability and time delays. The delay times used for the simulation are presented in the next tables. Note that for different types of request or prescription, delay times for preparation, transportation and administration may vary, and so, the initial values will be shown for each type separately, and then the available resources will be described. Since the time distribution is exponential (making the distribution times random), the changes made from simulation to simulation only relate to the resources common to all request types, i.e. the number of people available to the process. Being so, the relevant information about the process and what happens along its execution, is in the information about how many employees are available at any given moment.

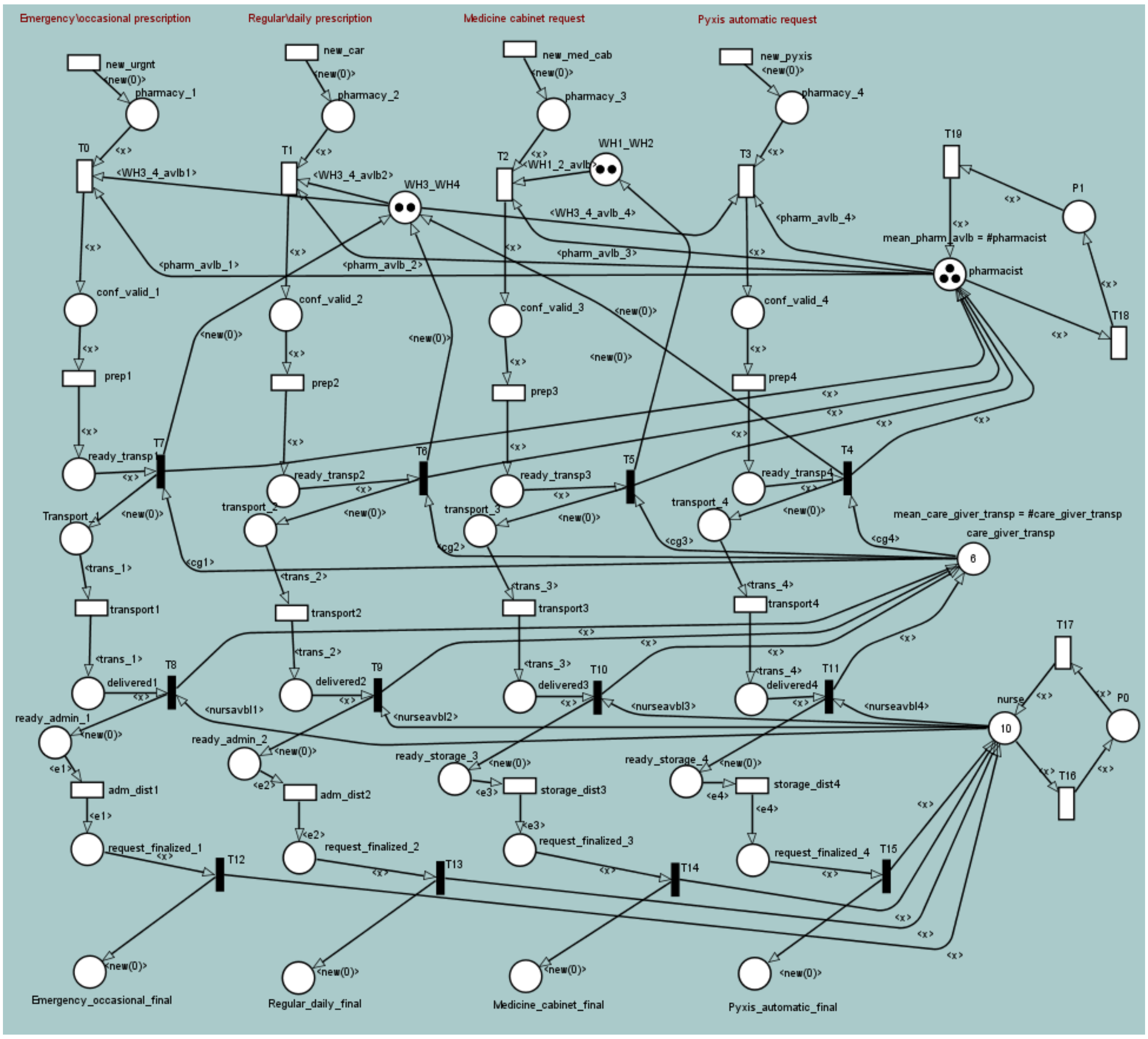


Figure 33 - Full net model for simulation

What this means is that for each group of resources shared between the distinct request types (pharmacists, care givers, nurses), the amount of usage of each resource, or, the number of them available (not occupied in the medication process) over time for a given demand or request rate. In addition to the net presented in figure 13 of chapter 5.1 the net used for simulation also has an alternate workload situation for both the pharmacists and the nurses. What this does is, when one resource is available for more than ten or fifteen minutes (exponential time distribution), it is accepted that the resource can participate in other activities that are not medication related. The medication distribution has maximum priority so that resources are only allocated when they are not needed in the medication preparation and distribution process.

6.4.1 - Simulation parameters

In order to extract knowledge from the Petri Net models, in cases like this where the invariants do not give allot of relative information about the net over time and for different situations of resource availability, a series of simulations with different availability definitions is more convenient. The delay times (Tables 3, 4, 5 and 6), on the various steps of the request's will be maintained, since they are exponentially distributed which makes them significantly random for simulation. What in fact can describe what happens to the net model over time, is the state of the model for each place or transition in the net throughout the simulation time.

Med.Cab.Req	Validation	Prep_start	Start_transporte	Med_Cab_Storage
2880	5	10	7	15

Table 3

Daily P. Req.	Validation	Prep_start	Start_transporte	Administration
60	5	25	7	60

Table 4

Urgent Req.	Validation	Prep_start	Start_transporte	Administration
120	3	5	4	2

Table 5

Pyxis Req.	Validation	Prep_start	Start_transporte	Pyxis_Storage
4320	5	10	7	10

Table 6

The first model resource values used for simulation are presented in the Table 7 and Table 8.

Pharmacist	W_H 1_2	W_H 3_4	Care-Giver	Nurse
4	2	2	6	10

Table 7

Pharmacist	W_H 1_2	W_H 3_4	Care-Giver	Nurse
3	2	2	3	5

Table 8

These resource values are initial values, since these are variables that will change over time (i.e. the number of tokens in the resource place is constantly changing) and give information on the state of the net due to each one's value and its relation with the other resources and their availability. One of the most important aspects is to know whether or not the available human resources are insufficient for the correct evolution of the net, or if they can be involved in another activity. This possible involvement in parallel activities will not only be verified by the resource place but also by the 'alternate work load' that displays when resources where available for more than the defined waiting time (five or ten minutes), and were used for another process. The priority between processes will be imposed by the global guard of the transitions that use these resources, i.e. by imposing a global guard that guarantees the resources won't be deviated from the medication process if there is a token in any of the prescription\request procedures.

6.4.2 - Simulation results

The first set of graphs returned by simulating the medication distribution model with the parameters and values described previously, are presented ahead and will be compared to results from the second simulation to observe what happens when the resource parameters are altered. Besides the watches on transitions and places of the net, two performance measures calculate and return the mean number of pharmacists and of care-givers available throughout the simulation (the same could be done for any place in the net), having returned the results illustrated in the graphs of Figure 34.

The difference between the two simulations is the fact that in the second case, the available human resources were reduced to verify what could happen in the eventuality of a staff reduction. It is observable in the figure that in the second case, there is significantly less availability from the care-givers and nurses, and yet, the pharmacist resource seems to still be enough in this case (one or two pharmacists appear to suffice for the greater part of the simulation).

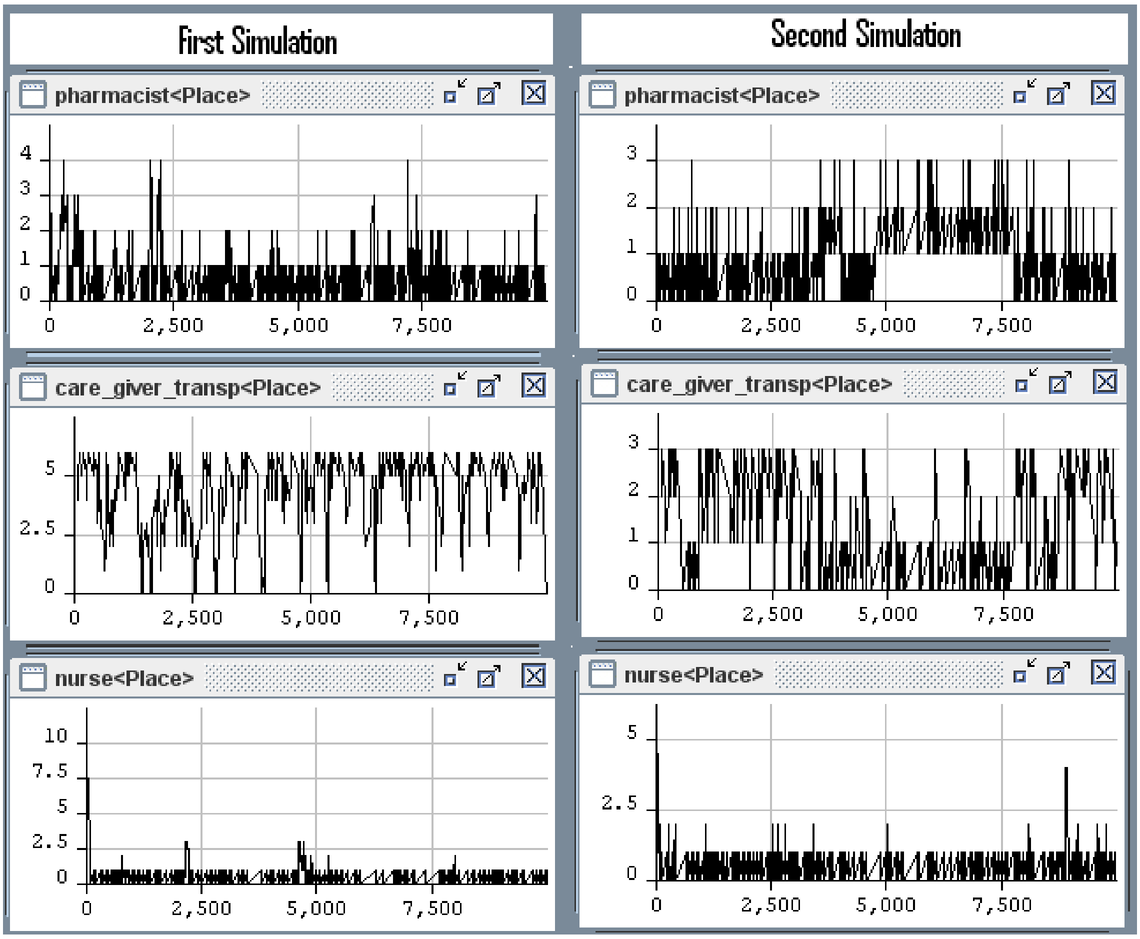


Figure 34 - Human resource availability simulation

The care-giver resource much like the nurse resource in the second case are mostly reduced to one or to no resource at all available, which can lead to the conclusion that for this volume of work load two pharmacist will practically always be enough, although nurses and care-givers are in need of more availability since there are repeatedly none of these resources available throughout the simulation. These conclusions can be understood better with the mean care-giver available graph (comparison between the first simulation graph and the second) presented in Figure 35.

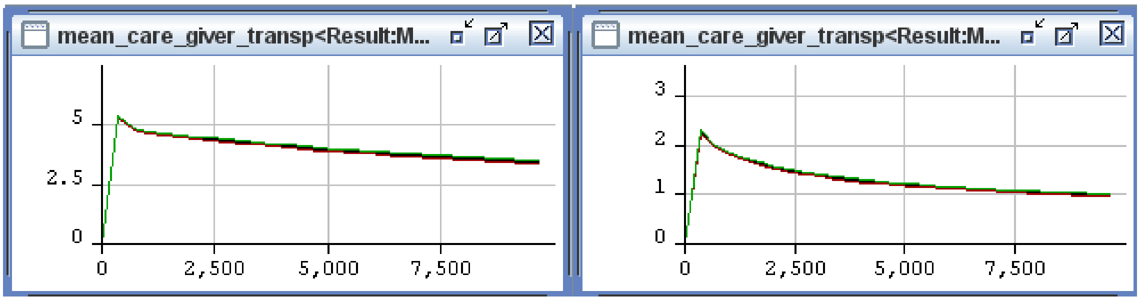


Figure 35 - Mean number of care-givers graph

Since the mean value considers the cumulative number o tokens divided by the time instant, the first value for t=0 is obviously zero, and then as the time variable rises the mean number of care-givers reaches its true value (the time step, or the number of sampling points can be defined by the user).

In the first simulation there were six care-givers dedicated to the process, and the mean number of them available over time was more than half of them, as the graph shows, while in the second case when three care-givers are defined, little over one is available (mean value over time (real number)). This means that in the first simulation, in average 50% of the resource was unused, and in the second case only one-third or about 30%. This does not necessarily represent a bad resource distribution; it merely describes the average number over time, of the care-givers not-occupied with the process. However, as shown in Figure 34 for the care-giver resource, in both cases (6 cg defined in the first and 3 cg in the second), the graph reaches the point where no care-givers are available (all of them are involved in a transport process). What happens is that in the first case, only a few times was it necessary to use the extra three care-givers, but when it happened, they were used and rapidly made available again. In the second case, when there is a need for more care-givers the graph shows that as soon as one is available, it becomes un-available immediately by going back into the process. This means that in the case where six cg were defined, for most of the time there where one or two available, but when only three are defined there is sometimes a waiting period due to un-availability that may or not be acceptable.

In general what can be perceived by comparing the graphs in Figures 34 and 35 is that the number of pharmacist in the second simulation are enough, but the amount of care-givers and nurses dedicated to the system must be closer to the ones simulated in the first case. This type of simulation can be helpful for predicting possible shortages of personnel or estimating the necessary amount of manpower needed to fulfill a determined work load. We can also observe from the “alternate work load” how many times it was possible to use the human resources in another procedure when they were available for a certain amount of time (simulated with 5 minutes). What this means is that if a nurse or a pharmacist is available for more than five minutes, they are accepted to be in another process and only return to the medication distribution system when the process is finished. This availability reduces the calculated mean number of resources available, but helps to see how many times a resource is not being used.

Figure 36 shows the state graphs for the places that correspond to the nurses and pharmacists that are committed to another process, and shows that despite some of the waiting periods, many times there is a resource that can perform another service and then return to the medication process, depending on the urgency and\or priority of each situation.

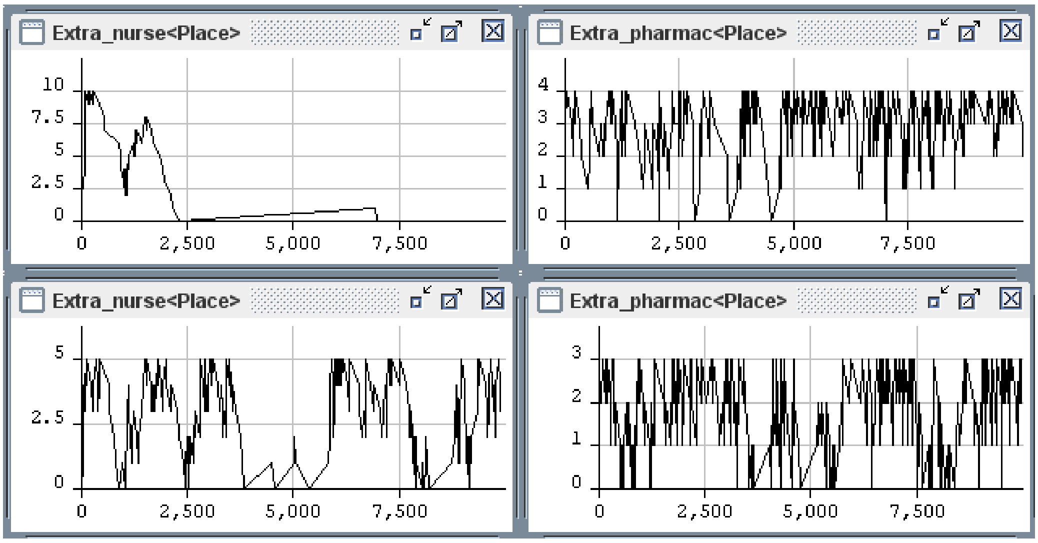


Figure 36 - Pharmacist & Nurse Alternative Work Loads

In the second case or simulation (bottom graphs) there appears to be more situations where a nurse can integrate a different task, even though in the nurse availability graphs in Figure 36 gives the idea that they are always occupied. This can be due to a matter of timing, i.e. the request for a nurse comes after she has been accepted in another process, having to wait for the same one or another one to become available and continue the distribution process.

The watches on transitions or places are returned with the evolution or growth of the number of tokens that pass through the element during the simulation. Each timed transition that begins a request starts depending on the expected time value defined for the delay, and sub-sequent delayed transitions graphically follow the grow rate of the initial transition. Figure 36 shows how similar the first and the last element behave graphically, due to the time condition imposed at the beginning of the process.

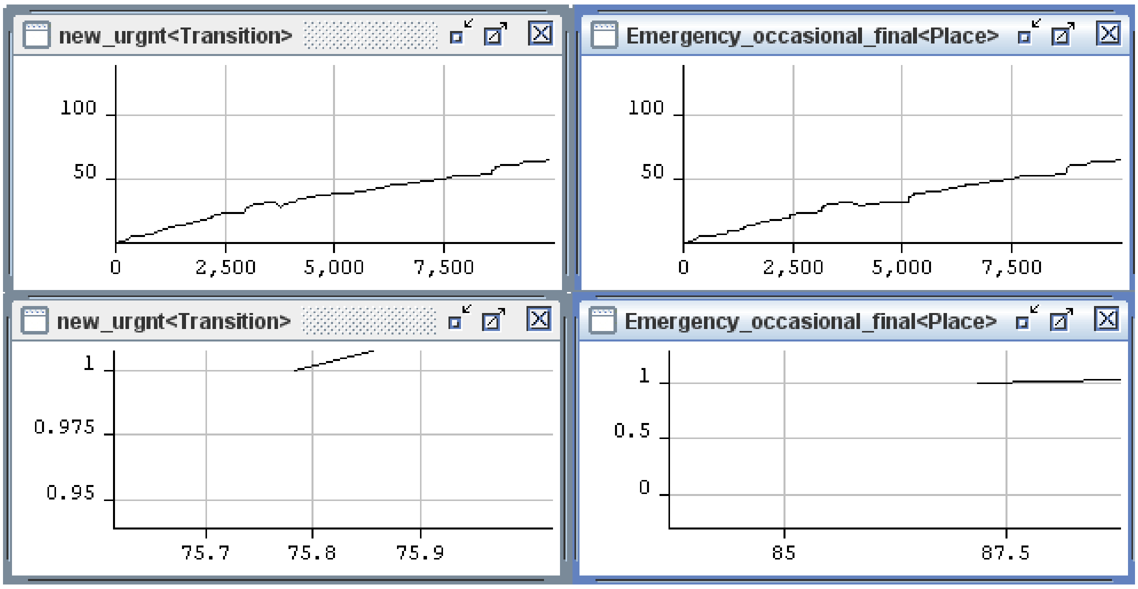


Figure 37 - Beginning and End graphs - Urgent request

Comparing these graphs, we see how the finalized request graph is extremely similar to the beginning transition graph, but also verify the total process time. This is because the first time delay is has a lot of influence in the rest of the process since it sets the pace or rhythm at which the order type happens, making the final graph almost identical to the initial one delayed the total time of the process. It is also observable in the graphs that are zoomed in, that the delay between the beginning of the process and the final place (last place of the net model for that type of request\prescription) is around twelve minutes. The expected values used for simulation were (in order of appearance in simulation model) 3, 4, 5, and 2, which sums up to fourteen time units, coming very close to the average expected time for this type of order. As mentioned before, all the transitions and places can be evaluated or watched over time so that this type of information by comparison can be made between procedures and within the same procedure, to study what happens over time to the elements of the model.

Although the exponential time delays defined along the process make the system unpredictable and random, which is good for simulation realism, a scenario can be simulated where more requests happen (more occurrences per period of time), meaning the expected value for each request type's time delay is smaller. Considering this, another simulation that contemplates a larger amount or frequency of orders will show how much of a difference in resource availability there would be for such a situation. The variables used for this simulation (number of human resources available as well as time delays for the request types) are presented in Tables 9 and 10.

Pharmacist	W_H 1_2	W_H 3_4	Care-Giver	Nurse
3	2	2	6	10

Table 9

Emergency Req.	Daily Request	Med. Cab. Req.	Pyxis. request
80	40	2000	3000

Table 10

The next figure (i.e. Figure 38) represents the graphs obtained for the previously mentioned scenario. As for the human resources, in this scenario, the pharmacist's seem to be enough, while the care givers have a large amount of waiting periods although small and in numerous occasions there are several of them available as can be observed on the graph. The nurse resource however seems to be in constant demand and practically never has more than one, if any at all, resource available.

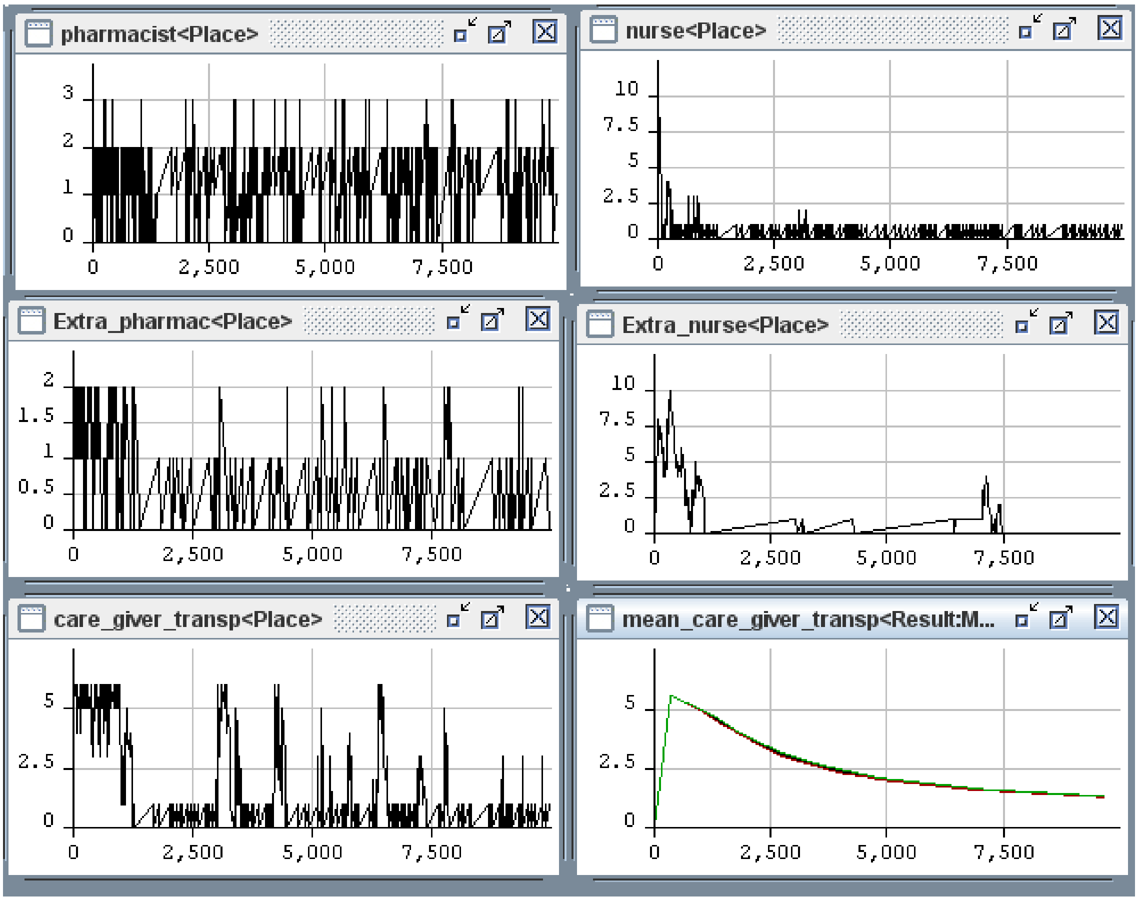
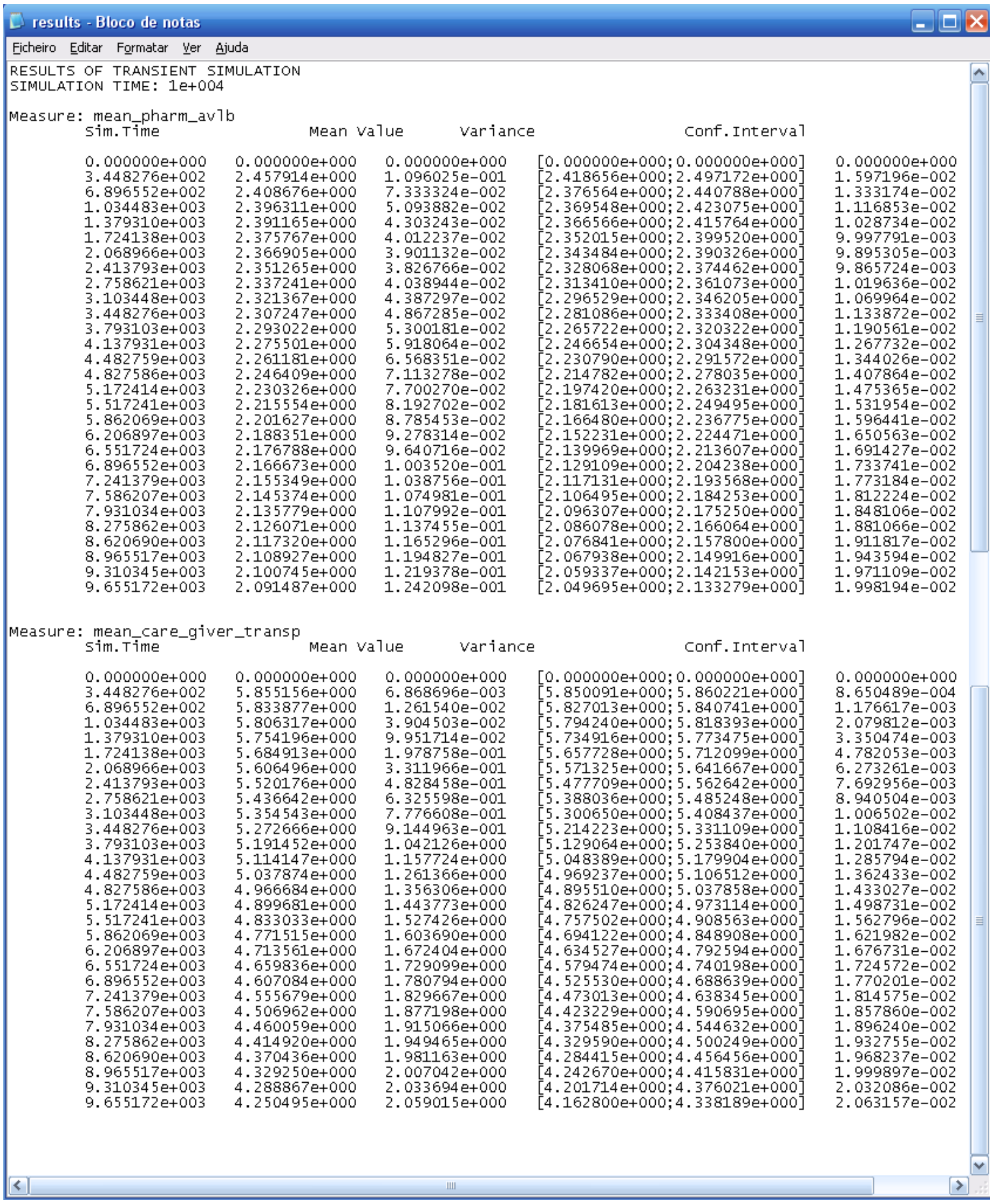


Figure 38 - Resource availability- simulation 2

In Figure 38 there is a graph that shows the evolution of the mean number of care-givers available over the simulation time (bottom right corner). However, this information is also returned by the 'simulationtrace' in the user-defined time intervals (i.e. the user chooses how many time intervals or the number of sampling points to consider in the simulation), or in an automatically software created folder as a notepad file, with the values from all defined performance measures as shown in Figure 39. Figure 39 shows a screen shot of the simulationtrace, which returns the mean number of resources (in this case care giver) for each simulation time interval, as well as the variance for the same interval.



```
RESULTS OF TRANSIENT SIMULATION
SIMULATION TIME: 1e+004

Measure: mean_pharm_avlb
        Sim.Time                Mean Value      Variance                Conf.Interval

        0.000000e+000   0.000000e+000   0.000000e+000   [0.000000e+000;0.000000e+000]   0.000000e+000
        3.448276e+002   2.457914e+000   1.096025e-001   [2.418656e+000;2.497172e+000]   1.597196e-002
        6.896552e+002   2.408676e+000   7.333324e-002   [2.376564e+000;2.440788e+000]   1.333174e-002
        1.034483e+003   2.396311e+000   5.093882e-002   [2.369548e+000;2.423075e+000]   1.116853e-002
        1.379310e+003   2.391165e+000   4.303243e-002   [2.366566e+000;2.415764e+000]   1.028734e-002
        1.724138e+003   2.375767e+000   4.012237e-002   [2.352015e+000;2.399520e+000]   9.997791e-003
        2.068966e+003   2.366905e+000   3.901132e-002   [2.343484e+000;2.390326e+000]   9.895305e-003
        2.413793e+003   2.351265e+000   3.826766e-002   [2.328068e+000;2.374462e+000]   9.865724e-003
        2.758621e+003   2.337241e+000   4.038944e-002   [2.313410e+000;2.361073e+000]   1.019636e-002
        3.103448e+003   2.321367e+000   4.387297e-002   [2.296529e+000;2.346205e+000]   1.069964e-002
        3.448276e+003   2.307247e+000   4.867285e-002   [2.281086e+000;2.333408e+000]   1.133872e-002
        3.793103e+003   2.293022e+000   5.300181e-002   [2.265722e+000;2.320322e+000]   1.190561e-002
        4.137931e+003   2.275501e+000   5.918064e-002   [2.246654e+000;2.304348e+000]   1.267732e-002
        4.482759e+003   2.261181e+000   6.568351e-002   [2.230790e+000;2.291572e+000]   1.344026e-002
        4.827586e+003   2.246409e+000   7.113278e-002   [2.214782e+000;2.278035e+000]   1.407864e-002
        5.172414e+003   2.230326e+000   7.700270e-002   [2.197420e+000;2.263231e+000]   1.475365e-002
        5.517241e+003   2.215554e+000   8.192702e-002   [2.181613e+000;2.249495e+000]   1.531954e-002
        5.862069e+003   2.201627e+000   8.785453e-002   [2.166480e+000;2.236775e+000]   1.596441e-002
        6.206897e+003   2.188351e+000   9.278314e-002   [2.152231e+000;2.224471e+000]   1.650563e-002
        6.551724e+003   2.176788e+000   9.640716e-002   [2.139969e+000;2.213607e+000]   1.691427e-002
        6.896552e+003   2.166673e+000   1.003520e-001   [2.129109e+000;2.204238e+000]   1.733741e-002
        7.241379e+003   2.155349e+000   1.038756e-001   [2.117131e+000;2.193568e+000]   1.773184e-002
        7.586207e+003   2.145374e+000   1.074981e-001   [2.106495e+000;2.184253e+000]   1.812224e-002
        7.931034e+003   2.135779e+000   1.107992e-001   [2.096307e+000;2.175250e+000]   1.848106e-002
        8.275862e+003   2.126071e+000   1.137455e-001   [2.086078e+000;2.166064e+000]   1.881066e-002
        8.620690e+003   2.117320e+000   1.165296e-001   [2.076841e+000;2.157800e+000]   1.911817e-002
        8.965517e+003   2.108927e+000   1.194827e-001   [2.067938e+000;2.149916e+000]   1.943594e-002
        9.310345e+003   2.100745e+000   1.219378e-001   [2.059337e+000;2.142153e+000]   1.971109e-002
        9.655172e+003   2.091487e+000   1.242098e-001   [2.049695e+000;2.133279e+000]   1.998194e-002


Measure: mean_care_giver_transp
        Sim.Time                Mean Value      Variance                Conf.Interval

        0.000000e+000   0.000000e+000   0.000000e+000   [0.000000e+000;0.000000e+000]   0.000000e+000
        3.448276e+002   5.855156e+000   6.868696e-003   [5.850091e+000;5.860221e+000]   8.650489e-004
        6.896552e+002   5.833877e+000   1.261540e-002   [5.827013e+000;5.840741e+000]   1.176617e-003
        1.034483e+003   5.806317e+000   3.904503e-002   [5.794240e+000;5.818393e+000]   2.079812e-003
        1.379310e+003   5.754196e+000   9.951714e-002   [5.734916e+000;5.773475e+000]   3.350474e-003
        1.724138e+003   5.684913e+000   1.978758e-001   [5.657728e+000;5.712099e+000]   4.782053e-003
        2.068966e+003   5.606496e+000   3.311966e-001   [5.571325e+000;5.641667e+000]   6.273261e-003
        2.413793e+003   5.520176e+000   4.828458e-001   [5.477709e+000;5.562642e+000]   7.692956e-003
        2.758621e+003   5.436642e+000   6.325598e-001   [5.388036e+000;5.485248e+000]   8.940504e-003
        3.103448e+003   5.354543e+000   7.776608e-001   [5.300650e+000;5.408437e+000]   1.006502e-002
        3.448276e+003   5.272666e+000   9.144963e-001   [5.214223e+000;5.331109e+000]   1.108416e-002
        3.793103e+003   5.191452e+000   1.042126e+000   [5.129064e+000;5.253840e+000]   1.201747e-002
        4.137931e+003   5.114147e+000   1.157724e+000   [5.048389e+000;5.179904e+000]   1.285794e-002
        4.482759e+003   5.037874e+000   1.261366e+000   [4.969237e+000;5.106512e+000]   1.362433e-002
        4.827586e+003   4.966684e+000   1.356306e+000   [4.895510e+000;5.037858e+000]   1.433027e-002
        5.172414e+003   4.899681e+000   1.443773e+000   [4.826247e+000;4.973114e+000]   1.498731e-002
        5.517241e+003   4.833033e+000   1.527426e+000   [4.757502e+000;4.908563e+000]   1.562796e-002
        5.862069e+003   4.771515e+000   1.603690e+000   [4.694122e+000;4.848908e+000]   1.621982e-002
        6.206897e+003   4.713561e+000   1.672404e+000   [4.634527e+000;4.792594e+000]   1.676731e-002
        6.551724e+003   4.659836e+000   1.729099e+000   [4.579474e+000;4.740198e+000]   1.724572e-002
        6.896552e+003   4.607084e+000   1.780794e+000   [4.525530e+000;4.688639e+000]   1.770201e-002
        7.241379e+003   4.555679e+000   1.829667e+000   [4.473013e+000;4.638345e+000]   1.814575e-002
        7.586207e+003   4.506962e+000   1.877198e+000   [4.423229e+000;4.590695e+000]   1.857860e-002
        7.931034e+003   4.460059e+000   1.915066e+000   [4.375485e+000;4.544632e+000]   1.896240e-002
        8.275862e+003   4.414920e+000   1.949465e+000   [4.329590e+000;4.500249e+000]   1.932755e-002
        8.620690e+003   4.370436e+000   1.981163e+000   [4.284415e+000;4.456456e+000]   1.968237e-002
        8.965517e+003   4.329250e+000   2.007042e+000   [4.242670e+000;4.415831e+000]   1.999897e-002
        9.310345e+003   4.288867e+000   2.033694e+000   [4.201714e+000;4.376021e+000]   2.032086e-002
        9.655172e+003   4.250495e+000   2.059015e+000   [4.162800e+000;4.338189e+000]   2.063157e-002
```

Figure 39 - Results File

All these simulations are simple examples of what kind of information can be extracted directly from the Petri net model as well as by comparing results from distinct scenarios of availability and execution times or delays.

7 - Conclusions and future work

The main objective of this work was to model the medication distribution process in Healthcare environments, using Petri Nets. These simulations can be used to extract knowledge about the net and its resources, to support decision making and resource attributions.

During the research performed for this study, many other works from various authors were found, where the logistics within a hospital were found to be one of the greater issues to improve in the attempt to, not only reduce general costs, but also, and found to be very important, improve the medication distribution process. The idea that a hospital must be seen as an industry in order to be able to quantify and attempt to reduce errors and avoidable expenses, is surging and creating a large debate. Not only from the medication point of view, but the entire logistic support system that keeps a hospital functioning.

In this work, the medication distribution process in a hospital, based on the case of HdRO, was modeled through the Petri nets formalism and using the TimeNet software tool. The designed models was simulated and knowledge about the system functioning extracted from the models. This information can be very useful in the understanding of the Petri net model and how it behaves or how it can be expected to behave in certain pre-determined conditions. The scenarios simulated by using the designed Petri net model, are possible events that are recreated with random time delays (exponential distribution based on an expected value) to simulate the unpredictability of the hospital logistic needs.

The net models presented only contemplate the medication distribution system from the point of a prescription or request reaches the pharmacy, up to the point it’s administered to the respective patient. Future developments could consider other logistic support systems present in the hospital that interact directly or indirectly with the medication distribution process or the human resources available and involved in the same support system. A net model that describes all the processes developed by the hospital warehouse could be created, to describe the entire logistic process developed by all the “actors” involved in such processes. Due to the unpredictability of (what is considered to be) an industrial process, simulations can be made for different scenarios, but it will always be impossible to know the hospitals needs for the next day or the next week, although it is possible to know for distinct scenarios, how the system might behave in those defined conditions.

This report attempts to achieve several “things”, to serve several purposes. The report is deliberately over-simplified with an eye toward the “bridging” function. This report may be unsatisfactory for many experts and practitioners but the amalgam of ideas may spark new directions of thinking. This is an exploration intended to encourage convergence and innovation. The author has freely used several sources of information to ‘connect the dots’ and show how distant disciplines, if coalesced, may offer new directions. For experts, there may be nothing ‘new’ in this report and for practitioners the vision may be too vague for implementation, but it is the synthesis of ideas from a variety of sources, when presented in confluence, as suggested in the report, that may be crucial in the transformation of decision support systems to adapt or perhaps, over time, to develop predictive intelligence.

In conclusion, this work can be considered a success. It not only embraces a point of view needed in the health-care industries that is the efficiency and safety of the medication distribution, but also focuses on the potential for extracting knowledge from a Petri net model, in this case, a net that describes a practically indescribable and unpredictable process. The time delays and resource distributions are the main aspect to consider and are essentially what define the process, and since they are either user defined or found by simulation, the net conditions can be totally known and help in assigning resources, finding buffer needs or unwanted time delays.

Bibliography

[1] BONFIM, JRA; MERCUCCI, VL., 1997 A construção da política de medicamentos. São Paulo, Ed. Hucitec. p.381

[2] ROSA, M.B.; PERINI, E., 2003 Erros de medicação: quem foi? Rev. Assoc. Med. Bras., São Paulo, v.49, n.3. p.335-341

[3] BABOLI A.; A. GUINET; T. HASSAN; M.T. BRANDON; B. CHARPIAT and G. LEBOUCHER., 2003. “Identification des flux de la logistique pharmaceutique d’un hôpital public”. p.392

[4] AARONS L.; I. NESTOROV; G. GRAHAM; S. DUFFULL; L. AARONS; E. FUSEAU and P. COATES., 2001. “Modeling and simulation for clinical trial design involving a categorical response. p.1210-1219

[5] CHOW G. and T.D. HEAVER., 1994. “Logistics in the Canadian Health Care Industry”. p.29-73

[6] B. BOLLABAS, 1998, Modern graph theory. New York: Spring-Verlag. p.87-97

[7] SARAH ATKINSON, AMÉLIA COHN, MARIA ELENA DUCCI and JASMINE GIDEON, 1999. Implementation of promotion and prevention activities in decentralized health systems: comparative case studies from Chile and Brazil. p.167-174

[8] MALIK AM, SCHIESARI, LMC, 2002. Qualidade na gestão local de serviços e ações de saúde. São Paulo: Editora Fundação Petrópolis. p.49-66

[9] NAPAL V, GONZALÉZ M, FERRÁNDIZ JR, 2002. Dispensación com intervención previa del Farmacêutico: dosis unitárias. In: FALGAS BJ, DOMINGUEZ A, HURLÉ G, PLANTAS MCG, LECUMBERRI VN, MOLINA EV. Farmacia Hospitalaria España. p.389-414

[10] LIMA CR, SILVA MDG, REIS VLS, 2000. Sistemas de Distribuição de Medicamentos em Farmácia Hospitalar. In: GOMES MJVM, REIS AMM. Ciências Farmacêuticas: uma abordagem em Farmácia Hospitalar. São Paulo: Editora Atheneu. p.25-32

[11] SILVA AEBC, 2003. Análise do sistema de medicação de um hospital universitário do estado de Goiás [Dissertação de Mestrado]. Ribeirão Preto: Universidade de São Paulo/Escola de Enfermagem de Ribeirão Preto. p.80-91

[12] CIPOLLE RJ, STRAND LM, MORLEY PC, FRAKES MJ, 2004 The Impact of Pharmaceutical Care Practice on the Practitioner and the Patient in the Ambulatory Practice Setting: Twenty-five Years of Experience. Bentham Science Publishers, The Netherlands Spring. p. 2-14

[13] BORJA AZORES MASSÒ, 2007. Logistica Hospitalaria Claves y tendencias de las operaciones logísticas en el sector hospitalario: calidad en la atención sanitaria y reducción de costes. p.45-58

[14] BAUSE and KRITZINGER, 2002. Stochastic Petri Nets- An Introduction to the Theory. Data Network Architectures Laboratory Department of Computer Science University of Cape Town Private Bag, RONDEBOSCH 7700 South Africa. p.80-98

[15] W. M. P. VAN DER AALST, K. M. VAN HEE and H. A. REIJERS, 2000, Analysis of discrete-time stochastic Petri Nets. Eindhoven University of Technology, Department of Mathematics and Computing Science. p.237-255

[16] SIFAKIS J., 1980, Performance evaluation of systems using nets, in: W. Brauer (ed.) Net theory and applications: Proceedings of the advanced course on general net theory, processes and systems (Hamburg, 1979), volume 84 of Lecture Notes in Computer Science, Springer-Verlag, Berlin. p.307-319

[17] FLORIN G. and S. NATKIN, 1982, Evaluation based upon stochastic Petri nets of the maximum throughput of a full duplex protocol, in: C. Girault and W. Reisig (eds.), Application and theory of Petri nets: selected papers from the first and the second European workshop, volume 52 of Informatik Fachberichte, Berlin, Springer-Verlag, Berlin. p.280-288

[18] M. AJMONE-MARSAN 1989. Stochastic Petri nets: An elementary introduction. In Advances in Petri Nets, Springer-Verlag. p.80-87

[19] ARMIN ZIMMERMANN and MICHAEL KNOKE, Time Net USER MANUAL, Technische Universit¨at Berlin Faculty of EE&CS Technical Report 2007.

[20] T.MURATA, Petri Nets. Properties, analysis and applications in: Proceedings of the IEEE, vol7, 1989. P.541-580

[21] FLYNN EA, BARKER KN. Medication error research. Medication errors. Causes, prevention, and risk management. Washington, D.C.: American Pharmaceutical Association; 1999.

[22] COHEN MR. Preventing dispensing errors. Medication errors. Causes, prevention, and risk management. Washington, D.C.: American Pharmaceutical Association; 1999.

[23] C.A. PETRI. Kommunikation mit Automaten. PhD thesis, Universit¨at Bonn,1962

[24] R.Z.ZURAWSKI, M.ZHOU. Petri Nets and Industrial Aplications:Atutorial, IEEE transactions on industrial electronics, December 1994. p.575