Chapter 6
Using Colored Petri Nets for Optimization of Healthcare Processes

Vijay Gehlot, Nilmini Wickramasinghe, Elliot B. Sloane, Michael Kirk, and Eric R. Miller

6.1 Introduction

In recognition of the need to ensure significant incorporation of health information technology into healthcare delivery, the US Government passed the Health Information Technology for Economic and Clinical Health (HITECH) Act, which included incentives to accelerate the adoption of health information technology (HIT) by the healthcare industry. Given that healthcare information technology can dramatically improve healthcare services delivery, reduce cost, improve care efficiency, and patient safety, under a government mandate, hospitals and medical care providers were required to adopt/introduce electronic systems for the management and delivery of healthcare services.

V. Gehlot
Villanova University, Villanova, PA, USA
e-mail: vijay.gehlot@villanova.edu

N. Wickramasinghe (✉)
Swinburne University of Technology, Melbourne, VIC, Australia

Epworth HealthCare, Melbourne, VIC, Australia

E. B. Sloane
Foundation for Living, Wellness, and Health, Sarasota, FL, USA

M. Kirk
Central Queensland Hospital and Health Service and The University of Queensland, Rockhampton, QLD, Australia
e-mail: michael.kirk@health.qld.gov.au

E. R. Miller
Central Queensland Hospital and Health Service, Rockhampton, QLD, Australia
e-mail: eric.miller@health.qld.gov.au

N. Wickramasinghe et al. (eds.), *Digital Disruption in Healthcare*, Healthcare Delivery in the Information Age, https://doi.org/10.1007/978-3-030-95675-2_6

The adoption of electronic health records (EHRs) and electronic medical records (EMRs) has resulted in a large amount of healthcare data in electronic form that can be computationally processed. Several healthcare organizations are utilizing data mining, machine learning, and related approaches to analyze healthcare data and improve the quality of care. However, data analysis alone cannot give insights into the underlying process. For example, the efficacy of clinical interventions identified by data analysis cannot be evaluated unless the underlying cause and effects are modeled. A report by the US Institute of Medicine emphasizes that many serious errors result from systems and their interactions rather than individual failures [1]. Thus, to effect changes to improve healthcare and to design and deploy better systems for improving human health, we also need to adopt tools and techniques for process modeling, simulation, and analysis.

Although modeling and simulation are widely used in many sectors, their adoption in healthcare has been challenging. A study, reported in [2], investigates modeling and simulation in healthcare against a context of defense and manufacturing industries. The authors report limited evidence of modeling and simulation being used to drive change in the healthcare delivery system. In addition to the complexities of a healthcare system, both [3, 4] identify stakeholder issues as a barrier to the successful and widespread use of simulation in healthcare. Results of a relatively recent survey dealing with modeling and simulation in healthcare are reported in [5]. The key summary of the survey is that modeling in healthcare is perceived to be different and more difficult across a range of factors. Reference [6] highlight three challenges for health modeling: First, how good is good enough, that is, what level of details should be included in models; second, clearly understanding how modeling is linked to decision-making; and third, dealing with the cultural barriers to adoption of modeling and simulation in the health sector.

In 2017, Academic Emergency Medicine convened a consensus conference on *Catalyzing System Change Through Healthcare Simulation: Systems, Competency, and Outcomes* to assess the impact of simulation on various aspects of healthcare delivery. The work reported in [7] is the summary of a breakout session on *understanding complex interactions through systems modeling*. Specifically, it explores the role that computer modeling and simulation can and should play in the research and development of emergency care delivery systems. The authors note that "One underutilized approach to addressing problems in healthcare quality and value, particularly in emergency care, is through the use of computer simulation modeling."

Furthermore, they emphasize that "Not unlike high-fidelity patient simulation for training clinicians in clinical care through the use of mannequins, computer simulation provides a platform to inform decision making prior to implementation in the real world."

Ample data are confirming that the number of emergency visits in the United States is going up whereas the number of emergency departments providing such services is on the decline. Furthermore, COVID-19 forced many hospitals to reevaluate and reengineer their workflows. For example, recently a healthcare facility had to transform from a traditional model of care to a virtual model of care in orthopedic surgery. They followed an OODA (Observe Orient Decide Act) approach toward this adaptation [8]. Although OODA is a powerful framework [9], it by itself does not provide a mechanism for validation to ensure, for example, patient safety is not compromised by the change. Access to a simulation-based tool, when used in conjunction with the OODA approach, can yield promising results. The authors of [7] note that "Computer simulation should be viewed as a necessary first step prior to implementation of a change in procedure or practice."

As noted earlier, stakeholder issues appear to be a barrier. However, in our own experience, part of the issue is the perceived learning curve associated with the simulation language (notation) and the lack of user-friendliness of associated tools. Even though stakeholders are not directly involved with actual model development, they need to be convinced that the adopted approach is user-friendly and, in particular, the adopted notation is understandable. This is where we see the strengths of a Colored Petri Nets (CPNs)-based approach and the underlying CPN Tools software [10–12]. The basic graphical/visual vocabulary of CPNs is small and intuitive, which renders them an attractive choice for modeling and simulation in healthcare.

The remainder of this chapter is organized as follows. Section 6.2 contains a hospital workflow example as described in [13]. We use this example to build our hierarchical CPN model, which we describe in Sect. 6.4. Before it, in Sect. 6.3, we give an overview of CPN and introduce the vocabulary of the CPN modeling language utilizing a simple example. Section 6.5 contains details of our simulation data collection and results. We present an approach to model verification and validation in Sect. 6.6. Finally, in Sect. 6.7, we present our conclusions.

6.2 Emergency Workflow Example

To illustrate our Colored Petri Nets-based approach, in this chapter we provide details of a CPN model of the emergency workflow described in [13]. The workflow, as described in the paper, is shown in Fig. 6.1. As depicted in this figure, there are two separate paths that a patient may take. The one on the left is taken by emergency patients whereas the one on the right is for elective surgeries where patients are initially hospitalized.

As part of the patient flow, the diagram explicitly depicts various resources that are needed at different stages of the flow. The aforementioned paper focuses on and distinguishes two types of resources: rooms (physical) and hospital staff (human). The various labels and their descriptions are given in [13] are as follows:

- Activity: reception (AA), transfer (AT), induction (AI), surgical operation (AO), and recovery (AR).
- Staff: nurse for reception (RI), anesthesiology staff for induction and operation (MSI), surgical staff for elective surgeries (MSH), surgical staff for emergency surgeries (MSU), nurse assistant (RAS), anesthesiology staff for recovery (MSR).
- Rooms: reception room (MA), induction room for elective surgery (MIH), induction room for emergency surgery (MIU), operating room for elective surgery (AOH), operating room for emergency surgery (AOU), recovery room (MR1).

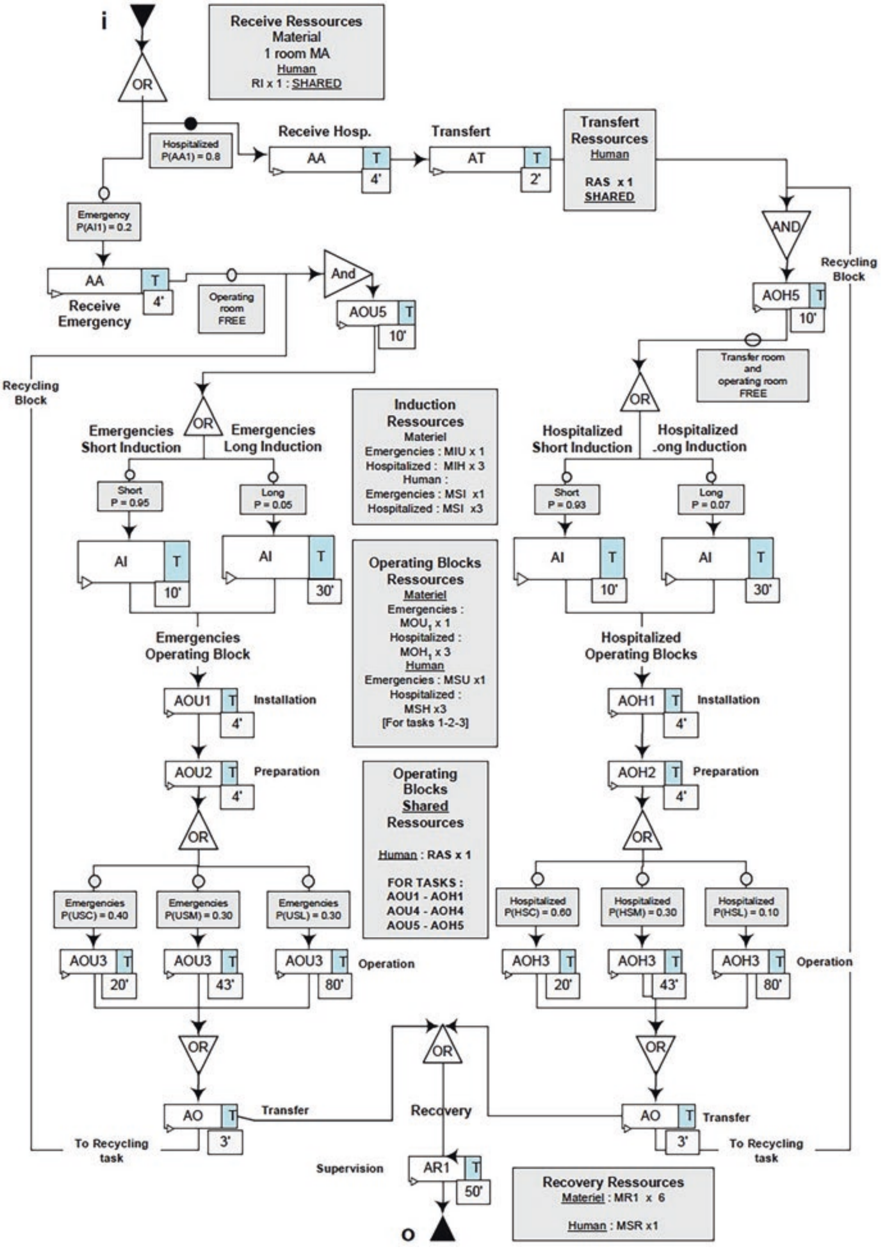


Fig. 6.1 The emergency workflow as described using a Workflow Management Systems (WFMS) notation in [13]. It describes the overall patient workflow in a healthcare system focusing on two different paths to OR, namely, Emergency workflow and Elective workflow

The shown diagram also gives delays in minutes for various activities as well as the probability of various choices. For example, the probability of a patient needing short induction on the emergency side is specified as 0.95, whereas the probability of short induction on the elective side is given as 0.93. In building our model, we use the same label and values where possible. For the benefit of the reader, before going into the details of our model, we give a brief introduction to the CPN vocabulary and modeling approach next.

6.3 Colored Petri Nets

Colored Petri Nets (CPNs) provide a graphical (visual) modeling notation well suited for concurrent and distributed systems in which communication, synchronization, and resource sharing play an important role. A key aspect of the CPN vocabulary is the ability to express a cause and its effect, which allows one to capture a workflow naturally. In terms of depiction, a CPN consists of *places* (depicted as circles or ovals), *transitions* (depicted as rectangles), and *arcs* (depicted as arrows) that connect a place to a transition or a transition to a place. Figure 6.2 shows a very basic CPN consisting of two places (P1 and P2) and one transition (T). We can interpret P1 as “Healthy,” T as “Bug Bites,” and P2 as “Sick,” thereby expressing a cause and its effect.

Places are containers of *tokens*. Depending on the context, tokens may represent a state, a data value, a resource, or some other entity. Transitions represent (abstraction of) actions. The cause and effect dynamics of a CPN are defined using the *firing rule*, whereby tokens are removed from input places of a transition and deposited in the output places of a transition. Thereby, recording the fact that the associated action has occurred. The distribution of tokens across places in a net is called a *marking* and describes the global state of the system being modeled. As mentioned earlier, another crucial aspect of the CPN notation is its ability to express sharing of resources and associated constraints, which are also inherent to healthcare workflows. For example, the availability of an operating room or an infusion pump is a resource constraint that would be part of the flow of care in a hospital dealing with trauma patients.

The basic execution semantics of a CPN in terms of the firing rule above gives rise to several interesting net configurations and associated interpretations that are natural in modeling workflows. Figure 6.3 depicts some net configurations useful for expressing various communication and coordination activities that form part of typical healthcare workflows. For example, the *Sequential* configuration is useful in capturing the dependency that a patient must register at the front desk before being

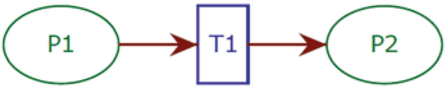


Fig. 6.2 A simple Colored Petri Net with two places and one transition

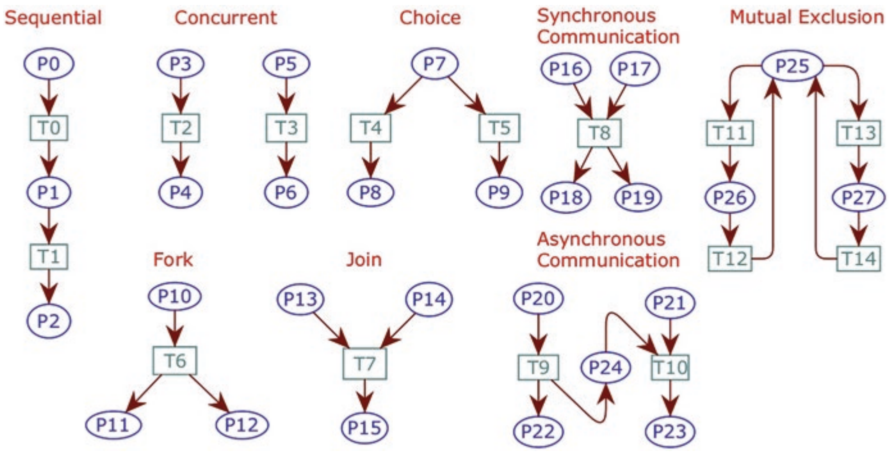


Fig. 6.3 Useful CPN configurations for modeling workflows and associated constraints [10]

examined by a nurse or a doctor. The *Concurrent* configuration captures the independence of events or flows. For example, a patient being checked for blood pressure is totally independent of another patient being checked into a trauma center. Therefore, these two actions can happen concurrently. The *Choice* configuration is useful in capturing the flow where two or more options are possible. For example, surgery or medication option for treating a tumor. The *Join* configuration provides a synchronization mechanism. For example, all test results must be in before proceeding further with the possible diagnosis or treatment. The *Synchronous Communication* is a generalization of the *Join* whereby it allows multiple outcomes. The *Asynchronous Communication* easily captures the flow where a test sample can be delivered to a lab by the clinical staff and then the lab can process it asynchronously without the staff waiting for it. Finally, the depicted *Mutual Exclusion* is useful in expressing resource-sharing constraints such as a single nurse cannot be attending to two different patients at the same time or a single monitor cannot be hooked to two different patients at the same time.

To explain the basic CPN notation further and its capability, we consider a concrete example of a very simple workflow where patients waiting for surgery can be taken in for surgery only if there is an operating room available. For this example, we are ignoring other resources, such as surgical staff, surgical instruments, and patient monitoring devices. The net in Fig. 6.4 captures this basic workflow. In this net, the active tokens are shown in small green circles. In this initial state, there are two *Available Operating Rooms*, as depicted by the associated token, and five *Patients Waiting for Surgery* as indicated by the associated token. The transition *In Surgery* can fire only if a patient is waiting (at least one token in the place named *Patients Waiting for Surgery*) and an operating room is available (at least one token in the place named *Available Operating Rooms*). The net in Fig. 6.5 is a snapshot of the next simulation step showing the state where one surgery is in progress (one token in the place named *Surgery in Progress*) and only one operating room is available, that is, the token count of *Available Operating Rooms* is now down to 1. At this

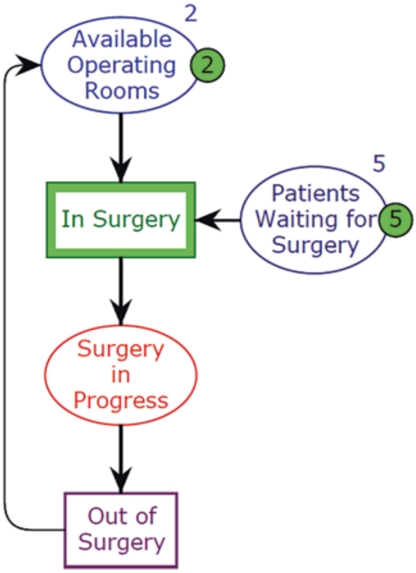


Fig. 6.4 A CPN model of a very simplified operating room workflow taking into account just the room availability. The net shows initially we have two operating rooms and five patients waiting

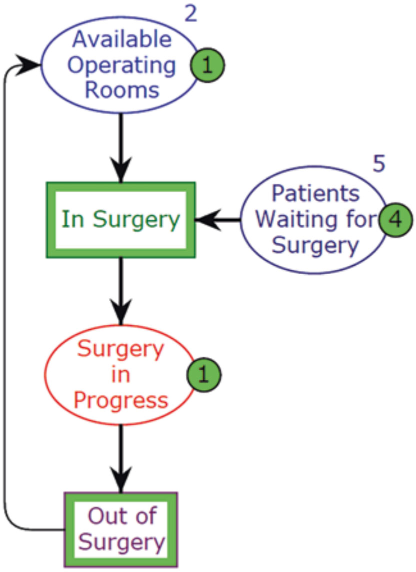


Fig. 6.5 The net showing a simulation state with 1 surgery in progress and 1 room available

stage, either another waiting patient can be taken in the surgery, or the current in surgery patient can be out of surgery or both since in the depicted net, both *In Surgery* and *Out of Surgery* transitions are simultaneously enabled (highlighted in green) and can fire. The net in Fig. 6.6 depicts the state where we have two patients in active surgery and we cannot take the next patient in since there is no token in *Available Operating Rooms* thereby disabling the *In Surgery* transition (not highlighted in green) although we have three more patients waiting. Once one of the

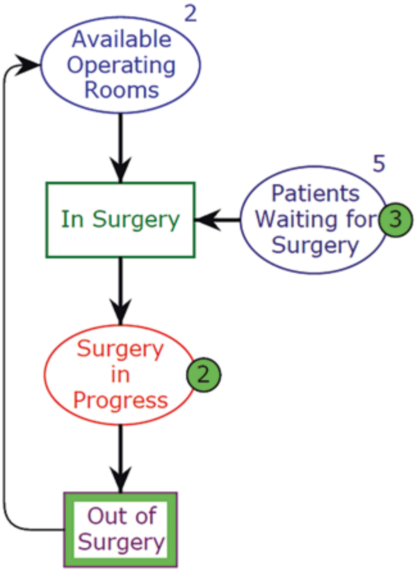


Fig. 6.6 The net showing the state where 2 active surgeries are in progress and we cannot take any more patients since the transition *In Surgery* is not enabled (highlighted in green)

currently active surgeries is done, a token representing room availability will be deposited in *Available Operating Rooms* via the arc connecting the transition *Out of Surgery* to *Available Operating Rooms*.

With this given background, we are now ready to describe the details of our CPN model. Readers interested in more details of CPN, including formal definitions and theoretical foundations, may refer to [14, 15].

6.4 CPN Model Details

We give details of our hierarchical CPN model that captures the details of the workflow shown in 1. The creation of hierarchical nets is based on the simple idea that any transition can be replaced or substituted by a (sub) net that details the activities underlying it. Such transitions are called *substitution transitions* (or *modules*) in the CPN parlance. Pictorially, a substitution transition is drawn with double rectangles.

The (hierarchical) net in Fig. 6.7 shows the overall patient workflow starting with the entry of a patient from reception to the exit from the recovery system. The shown patient workflow net consists of four modules, namely, *Patient Entry*, *Emergency Workflow*, *Elective Workflow*, and *Recovery*, and five places namely *To Emergency*, *To Elective*, *From Emergency*, *From Elective*, and *Discharge*. The diagram in Fig. 6.8 shows the module hierarchy, that is, the various sub-modules and their nesting structure that comprises our hierarchical model.

The tokens in the basic model in Fig. 6.4 do not carry any information. For a detailed analysis, we may want to carry additional information in tokens. For example, we may want to distinguish different types of operating rooms or patients with

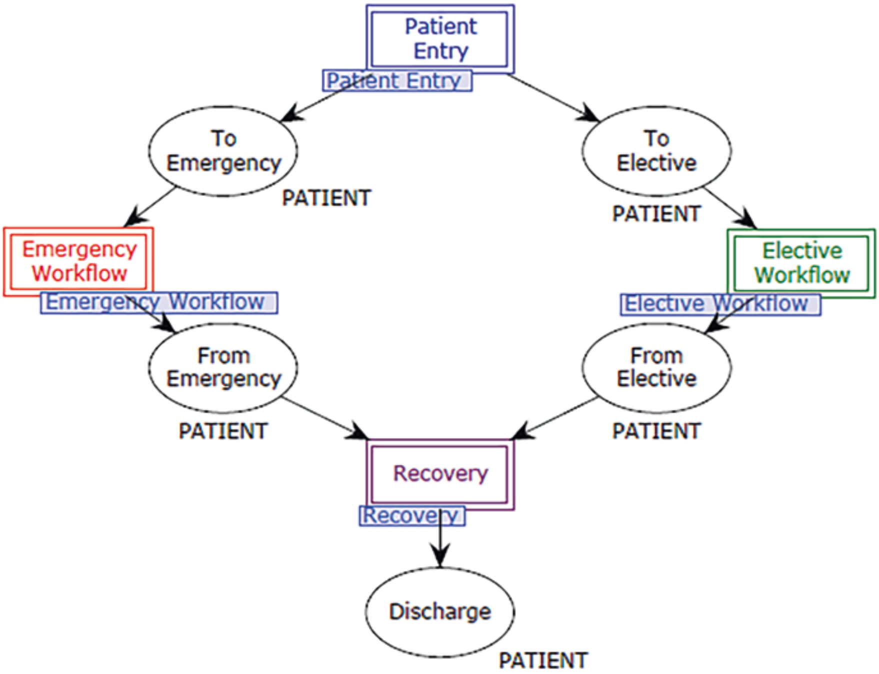
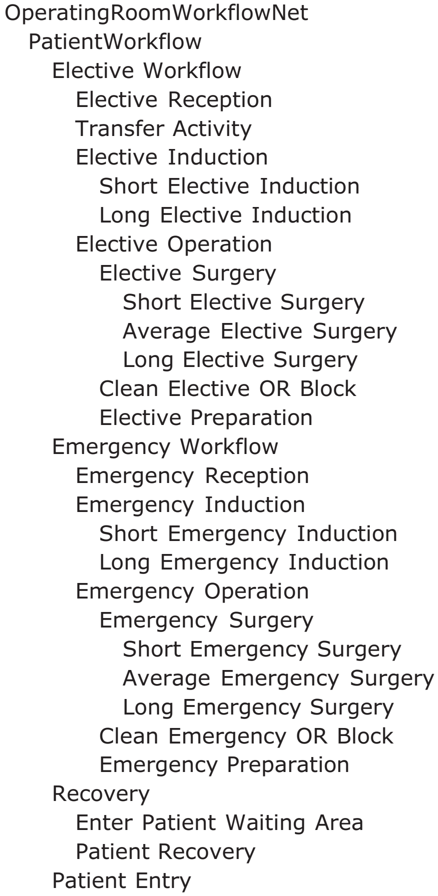


Fig. 6.7 The top-level net showing the overall workflow and associated modules

different conditions. CPNs provide an enhanced vocabulary to create tokens of different data types (or *colorsets* in CPN parlance) and utilize the full functionality of the underlying inscription language CPN ML, which is built on top of the functional programming language SML [16]. Before going into details of some of the submodules, we give a brief description of key colorsets used in this model below:

```
(* Model colset declarations *) colset PTYPE = with EM | EL; colset
PID = INT;
                colset PID_T = PID timed; colset AT = INT;

colset PATIENT = product PTYPE * PID * AT; colset PATIENTS = list
PATIENT;
               colset ROOM = with MA | MIU | MIH | AOU | AOH | MRI
| WR; colset ROOMS = list ROOM;
                colset HR = with RI | MSI | MSH | MSU | RAS | MSR;
colset STAFF = list HR;
                colset PSTAT = product PATIENT * ROOMS * STAFF;
colset PSTAT_T = PSTAT timed;
                colset HRACT = product HR * ROOM timed;
```



```
OperatingRoomWorkflowNet
  PatientWorkflow
    Elective Workflow
      Elective Reception
      Transfer Activity
      Elective Induction
        Short Elective Induction
        Long Elective Induction
      Elective Operation
        Elective Surgery
          Short Elective Surgery
          Average Elective Surgery
          Long Elective Surgery
        Clean Elective OR Block
        Elective Preparation
    Emergency Workflow
      Emergency Reception
      Emergency Induction
        Short Emergency Induction
        Long Emergency Induction
      Emergency Operation
        Emergency Surgery
          Short Emergency Surgery
          Average Emergency Surgery
          Long Emergency Surgery
        Clean Emergency OR Block
        Emergency Preparation
    Recovery
      Enter Patient Waiting Area
      Patient Recovery
    Patient Entry
```

Fig. 6.8 The module hierarchy of the CPN model. The module hierarchy depicts the various sub-modules and their nesting structure that comprises our hierarchical model

These types are used to carry the following information, which is used in the model description, creation, and simulation:

- `PTYPE` or patient type allows us to distinguish emergency `EM` from elective (`EL`). In general, a more complex type may be associated that will allow other patient or application-specific attributes.
- `PID` is patient ID and `PID_T` is the associated timed version. The latter allows the creation of the timed tokens to account for various delays and processing times.
- `PATIENT` is a compound type consisting of patient type, patient ID, and patient's arrival time. `PATIENTS` is a list of patients useful in describing a queue.
- `ROOM` is a room type based on the workflow described above and `ROOMS` is used to represent a set of rooms.
- `HR` is a human resource type per the workflow described above and `STAFF` is a list of those.
- `PSTAT` is a compound type that captures the status of a patient in terms of assigned rooms and assigned staff. `PSTAT_T` is its associated timed version for

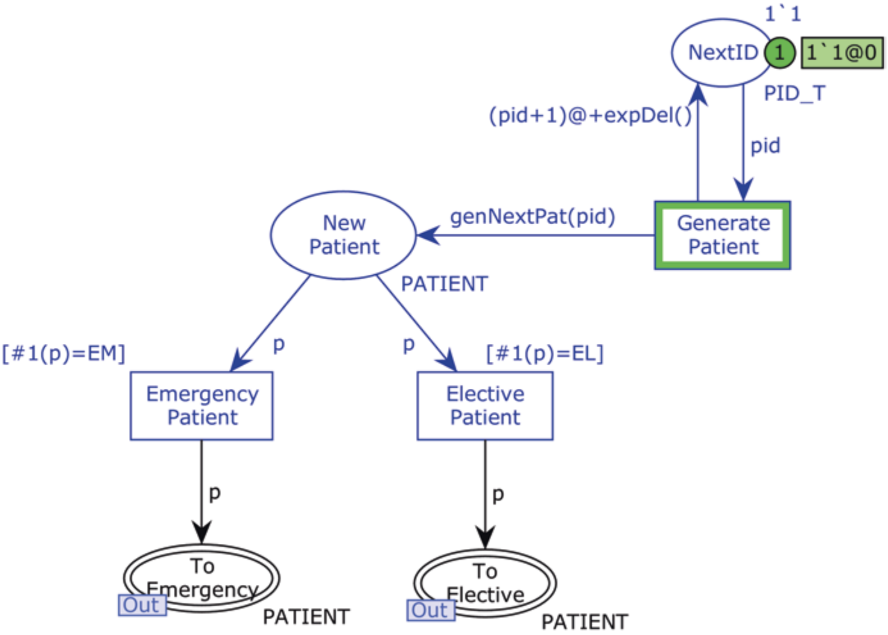


Fig. 6.9 The *Patient Entry* module responsible for the generation of patient traffic

performance metrics. HRACT is a compound type denoting which human resource is active (or assigned to) in which room. It is a timed colorset for performance metrics.

We start with the module *Patient Entry* module, which is shown in Fig. 6.9. This module is responsible for generating patients who either go for elective or emergency surgery. The original paper [13] specifies 80% to be elective surgeries and 20% to be emergency surgeries as shown in Fig. 6.1. However, it does not specify any arrival pattern or rate. Thus, in this chapter, we have assumed the interarrival time to be exponentially distributed. Using the file input/output and external process communication faculties of the CPN Tools, we can certainly drive a CPN simulation based on an actual log of patient arrivals if available. Internally, this module utilizes the type PID_T to generate a timed token with the next patient ID and arrival time. Based on this information, a token of type PATIENT is generated, which will move either to *To Emergency* or *To Elective* depending on the PTYPE value of the token.

NextID place represents the state of the number of patients with their waiting times. When *Generate Patient* transition occurs, it puts back a token in *NextID* with the next number and randomly generated a waiting time for the next patient. The CPN ML function genNextPat(pid) on the arc from *Generate Patient* is responsible for generating a patient token and depending on the patient type of this token, it will move either *To Emergency* or *To Elective*.

After this, the patient (or token) will follow the *Emergency Workflow* module or the *Elective Workflow* module of the net shown in Fig. 6.7. The two workflows essentially differ in terms of the *Transfer Activity* module as given by the module hierarchy diagram in Fig. 6.8. We, therefore, focus mainly on the details of the *Emergency Workflow* module. Specifically, we present details of the following sub-modules: *Emergency Induction*, and its sub-module *Long Emergency Induction*; *Emergency Operation* and two of its sub-modules, namely, *Emergency Preparation* and *Long Emergency Surgery*; and finally the *Patient Recovery* module.

The next two modules, *Emergency Induction* and its sub-module *Long Emergency Induction* are shown in Figs. 6.10 and 6.11, respectively. Neither [13] nor Fig. 6.1 indicates an explicit queue, but in our model, we have put an explicit queue at the start of various activity stages for better accounting of delays. Otherwise, multiple tokens in a place are viewed as a multi-set with no specific order. As shown in the figure, when the transition *Add to Queue* fires, the incoming patient token will be added to the *Emergency Induction Queue*. The next patient in the queue enters the

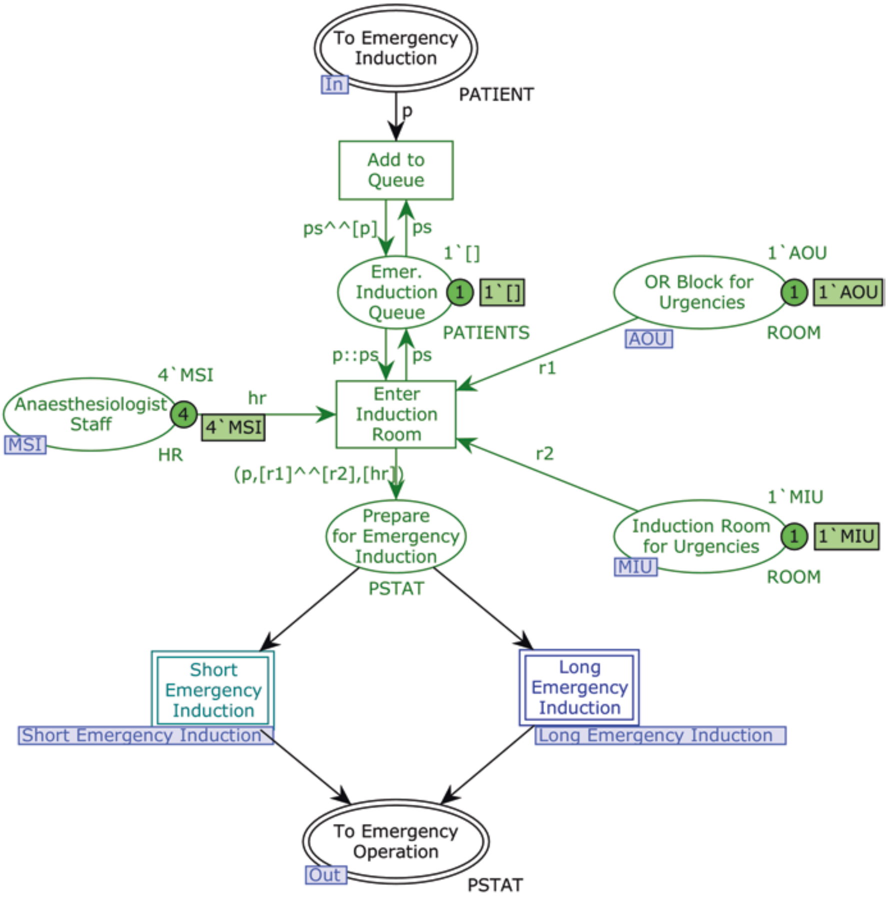


Fig. 6.10 The *Emergency Induction* module as shown in the module hierarchy of *Emergency Workflow* in Fig. 6.8

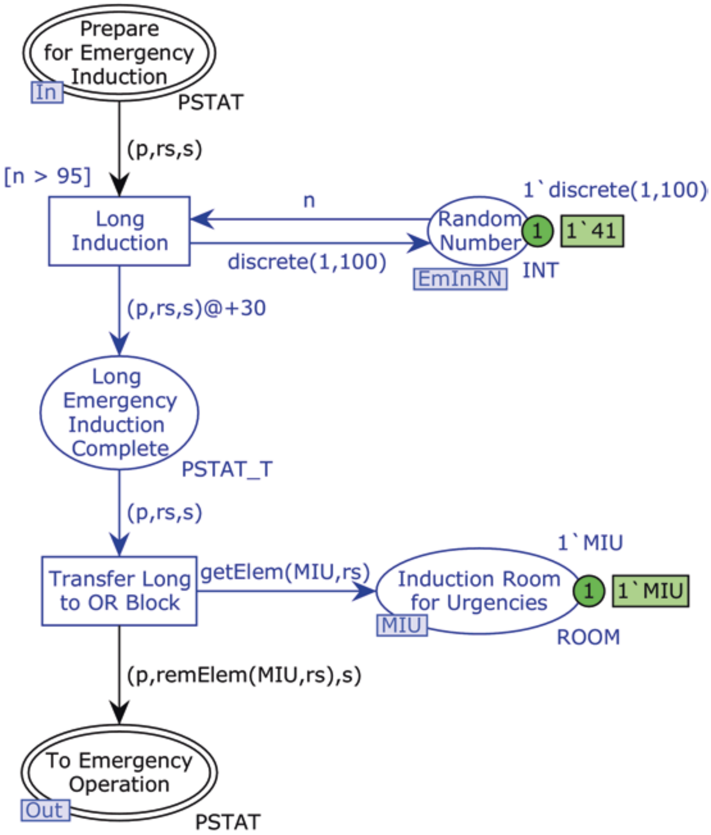


Fig. 6.11 The sub-module *Long Emergency Induction* of the *Emergency Induction* module as shown in the module hierarchy of *Emergency Workflow* in Fig. 6.8

induction room only if *OR Block for Urgencies* and *Induction Room for Urgencies* is available.[1] Additionally, it requires the availability of an *Anesthesiologist Staff*. All these resource constraints are captured in a very simple and visual manner by the incoming arcs of the *Enter Induction Room* transition in the figure.

The net in Fig. 6.11 shows the *Long Emergency Induction* module. The Boolean condition `[n > 95]` on the transition *Long Induction* and the random number in the connecting place *Random Number* guarantee the probability of long induction to be 0.05, as specified in Fig. 6.1. Note that the place *Random Number* is shared with the activities of the corresponding *Short Emergency Induction* (not shown) to ensure that both modules are using the same number in determining the firing of the associated transition. This sharing is achieved via the CPN notion of a *fusion set* whereby a set of places may be fused as one by associating a fusion tag with those

[1] We are using the term Urgency instead of Emergency per the original paper.

places. We have used the fusion tag *EmInRN* as shown in the figure above. An associated timed token in *Long Emergency Induction Complete* determines the time for long induction.

After induction, a patient moves to *Emergency Operation*, which itself consists of two sub-modules: *Emergency Preparation* and *Emergency Surgery*. As depicted in Fig. 6.1, emergency surgeries can be either of short duration or average duration or long duration. We only include the *Long Emergency Surgery* module here since the other two are similar. Figures 6.12 and 6.13 show the two sub-modules *Emergency Preparation* and *Long Emergency Surgery*, respectively. As shown in the associated net, *Patient Installation* requires the availability of *Medical Staff for Urgencies* and *Nurse Assistants*. Once *Patient Preparation* is finished, the *Nurse Assistant* becomes available for other patients as captured by the outgoing arc from *Patient Preparation* to *Nurse Assistant*. At this stage, the human resource *Medical Staff for Urgencies* is considered still in use, that is, busy. The prepared patient then enters *Emergency Surgery*. A patient requiring long surgery will follow the net depicted in Fig. 6.13. The Boolean condition `[n > 7]` on the transition *Long Emergency Surgery* and the random number in the connecting place *Random Number* guarantee the probability of long surgery to be 0.30, as specified in Fig. 6.1. An associated timed token in *Patient in Long Emergency Surgery* determines the time for surgery. When done, that is, the transition *Complete Long Surgery* fires,

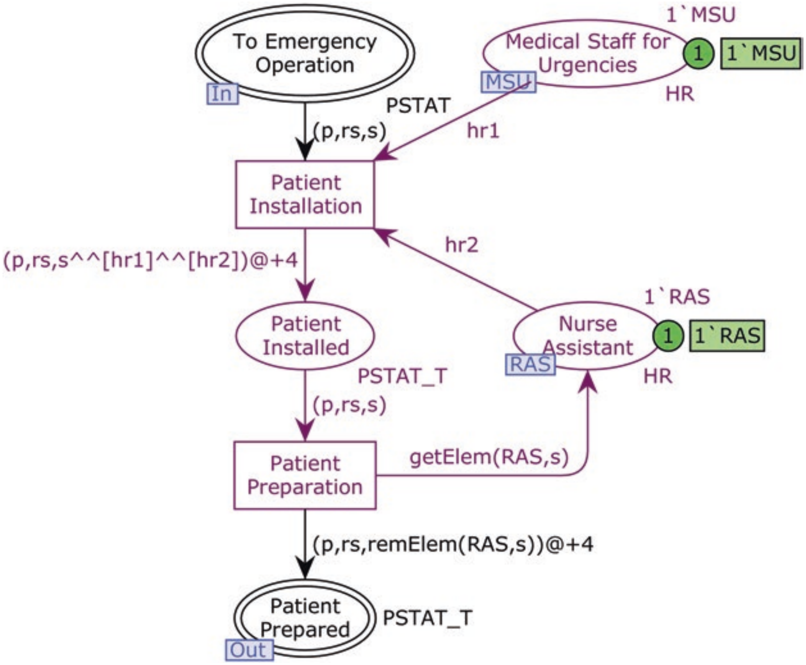


Fig. 6.12 The *Emergency Preparation* sub-module of *Emergency Operation* as shown in the module hierarchy of *Emergency Workflow* in Fig. 6.8

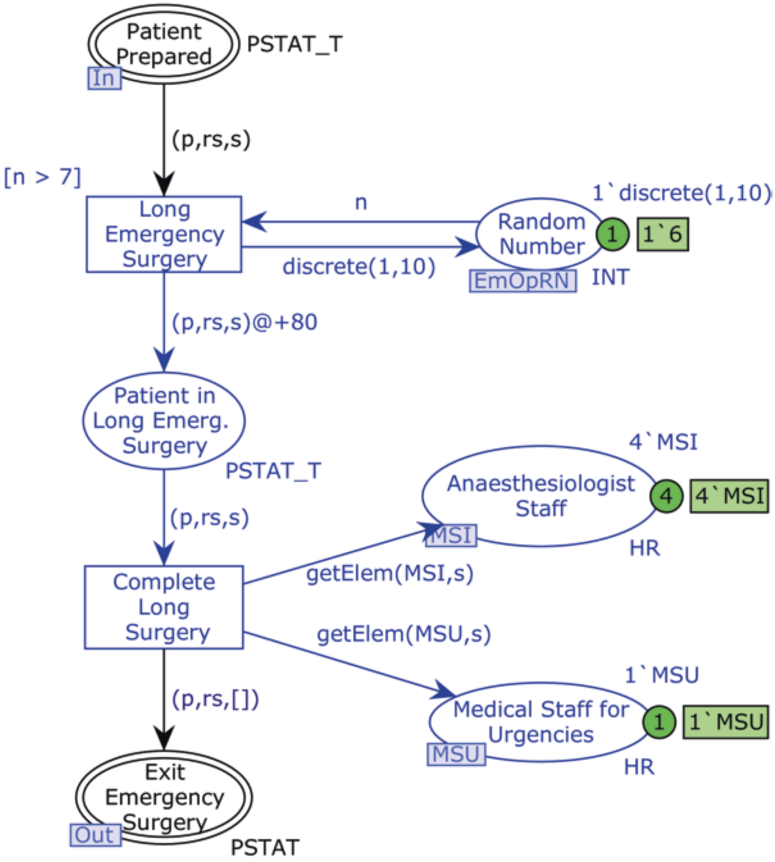


Fig. 6.13 The *Long Emergency Surgery* sub-module of *Emergency Operation* as shown in the module hierarchy of *Emergency Workflow* in Fig. 6.8

both human resources, namely, *Anesthesiologist Staff* from the induction stage and *Medical Staff for Urgencies* from the patient preparation stage are returned to their respective free pools.

The final stage is patient recovery. The associated *Patient Recovery* sub-module is shown in Fig. 6.14. As depicted in the associated net, Transfer to Recovery Room requires availability in the *Recovery Room* and an available *Anesthesiologist Staff for Recovery*. At this stage, the *Nurse Assistant* and the *Waiting Room* from the previous stage are returned to their respective free pools. An associated timed token in *Enter Recovery Room* determines the time for recovery. Once the recovery is complete, that is, the model time reaches the time stamp on the timed token, and the *Recovery* transition fires, the room and the staff are returned to their respective free pools, and the patient is moved to *Discharge*.

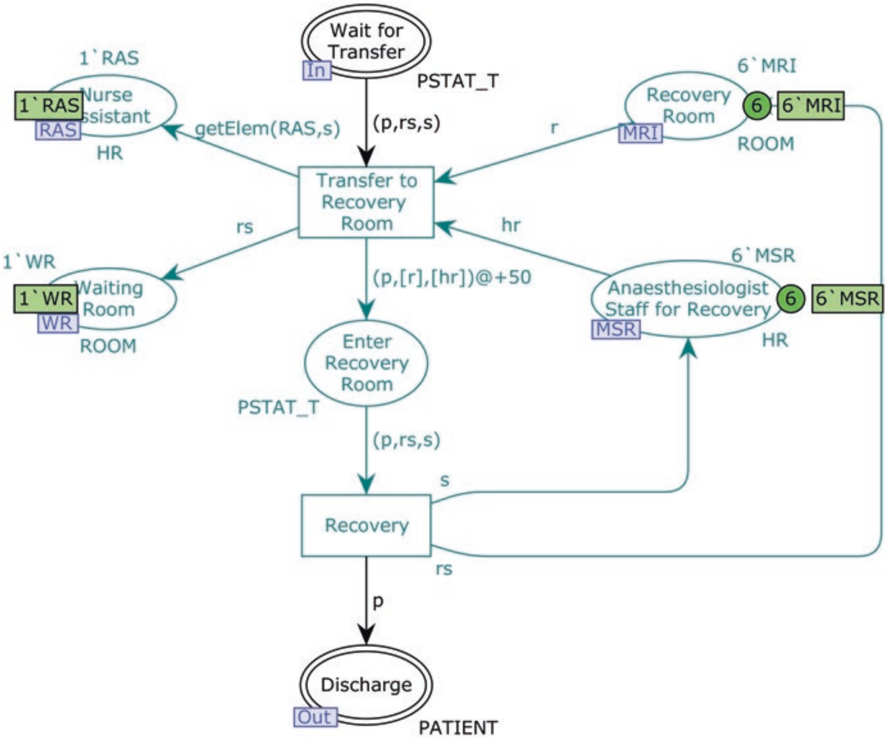


Fig. 6.14 The *Patient Recovery* sub-module as shown in the module hierarchy of *Emergency Workflow* in Fig. 6.8

This completes the discussion of our hierarchical CPN model. Next, we briefly describe the monitoring faculties of CPN Tools we utilized to collect data and generate performance reports.

6.5 Data Collection and Results

CPN Tools provide a monitoring facility to conduct performance analysis of a system [17]. Monitors are used to extracting relevant data during a simulation run. Monitors can be associated with any subnet of interest. Different types of monitors can be defined for a net. For example, a *simulation breakpoint monitor* can be used to stop a simulation run based on a specified condition. A *data collector monitor* is used to extract numerical data from a model during a simulation and to calculate statistics for the extracted data. The statistics that are calculated for a particular data collector are either untimed statistics or timed statistics (that is, time-dependent weighted statistics). The statistics that are computed and can be accessed from each data collector monitor are: count (number of data observations), minimum, maximum, sum, average, confidence intervals for average, variance, standard deviation, the sum of squares, the sum of squares of deviation, first value observed, and last value observed. Once monitors have been created, the built-in function `CPN'Replications.nreplications` can be used to run any number of simulation replications, collect data, and calculate, among other values, 90%, 95%, and 99% confidence intervals for averages. It also auto-generates a *performance report* containing statistics, including confidence intervals, that are calculated for the *independent and identically distributed* (IID) data values in the replication output log files.

We set a breakpoint monitor for a 24-h period and ran simulation replications with a medium traffic flow with an average inter-arrival of 1 h and another with intense traffic flow with an average inter-arrival of 10 min. Table 6.1 contains some data from the first replication run. Our results show that the utilization rates of both the anesthesiologist staff and recovery rooms were low, highlighting a potential area to save resources. Furthermore, while the nurse assistant maintained a comfortably high utilization rate, the rate of the reception nurse was much lower, showing the potential of reclassifying them into a shared resource.

6.6 Model Verification and Validation

The starting point of building a simulation model should always be a conceptual model [18]. One may utilize an informal notation or a formalized notation in describing a conceptual model. Typically, the notation should be expressive enough to capture the key requirements. In general, a conceptual model is a blueprint for the computer (or simulation) model to be built. Once a model has been created, a key exercise is to carry out the verification and validation of the model. Towards this end, we recommend adopting the approach described in [19].

According to [19], *verification* is the process of determining that a model implementation accurately represents the conceptual description and specifications whereas model *validation* is the process of determining the degree to which a model is an accurate representation of the real world. In particular, "...operational validation is carried out to determine the simulation model's output behavior has the accuracy required for the model's intended purpose over the domain of the model's intended applicability."

Model building is a collaborative process and both verification and validation steps require input from the stakeholders and subject-matter experts [20]. Additionally, the validation step requires access to data from actual operations. Verification ensures that the key requirements have been captured by the model. Both verification and validation are iterative processes and should be carried out hand in hand with model building, preferably following an agile approach. In the past, quality data for healthcare applications may not have been readily available in all situations of interest, but with the advent and progress of digital health, a modeler may have easier access to data of interest. In the validation step, data generated

Table 6.1 Sample data from the auto-generated performance report after five simulation replications run of the model for a set of input parameters

Name	Avrg	90% half length	95% half length	99% half length	StD	Min	Max
AOH_Util_Rate							
count_iid	42.400000	6.052778	7.881103	13.070820	6.348228	32	49
max_iid	1.000000	0.000000	0.000000	0.000000	0.000000	1	1
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	0.558624	0.070717	0.092078	0.152711	0.074168	0.457873	0.658333
AOU_Util_Rate							
count_iid	7.800000	2.730292	3.555015	5.895997	2.863564	4	12
max_iid	1.000000	0.000000	0.000000	0.000000	0.000000	1	1
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	0.230602	0.134498	0.175125	0.290445	0.141063	0.080378	0.427827
Elective_MWT_Before_Induction							
count_iid	21.800000	3.529088	4.595098	7.620976	3.701351	16	26
max_iid	32.800000	25.112390	32.697934	54.229570	26.338185	9	70
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
sum_iid	88.200000	64.103394	83.466708	138.429656	67.232433	29	182
avrg_iid	3.916977	2.503606	3.259854	5.406473	2.625813	1.260870	7.000000
Emerg_MWT_Before_Induction							
count_iid	4.000000	1.348395	1.755697	2.911825	1.414214	2	6
max_iid	15.600000	15.382938	20.029566	33.219064	16.133815	0	41
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
sum_iid	16.200000	15.672738	20.406904	33.844881	16.437761	0	41
avrg_iid	3.366667	2.801055	3.647152	6.048806	2.937781	0.000000	6.833333
MR1_Util_Rate							
count_iid	48.000000	7.627675	9.931720	16.471771	8.000000	36	58
max_iid	1.000000	0.000000	0.000000	0.000000	0.000000	1	1

min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	0.476619	0.028060	0.036535	0.060594	0.029429	0.432046	0.513455
MSI_Util_Rate							
count_iid	50.200000	8.591722	11.186970	18.553606	9.011104	36	61
max_iid	1.000000	0.000000	0.000000	0.000000	0.000000	1	1
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	0.491670	0.066555	0.086659	0.143724	0.069804	0.386145	0.581944
RAS_Util_Rate							
count_iid	191.600000	30.995497	40.358114	66.933990	32.508460	140	230
max_iid	1.000000	0.000000	0.000000	0.000000	0.000000	1	1
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	0.319927	0.051208	0.066676	0.110583	0.053708	0.234559	0.383495
RI_Util_Rate							
count_iid	52.600000	9.667157	12.587255	20.875980	10.139033	37	65
max_iid	1.000000	0.000000	0.000000	0.000000	0.000000	1	1
min_iid	0.000000	0.000000	0.000000	0.000000	0.000000	0	0
avrg_iid	0.071194	0.013382	0.017424	0.028898	0.014035	0.049827	0.088520
Total_Patient_Time							
count_iid	24.000000	3.813838	4.965860	8.235886	4.000000	18	29
max_iid	192.600000	24.267368	31.597661	52.404766	25.451915	167	220
min_iid	92.000000	6.105119	7.949254	13.183850	6.403124	85	97
sum_iid	2996.600000	627.867521	817.523564	1355.864008	658.515224	2061	3864
avrg_iid	123.928798	6.786832	8.836888	14.655991	7.118113	114.500000	133.241379

by a verified model is compared against real-life data and the model is fine-tuned by changing parameters, if necessary, to align with the real data.

As mentioned earlier, for data generation and validation purposes, CPN Tools software provides an extensive monitoring and simulation report generation facility [17]. A simulation report provides a complete execution trace of the model whereas a monitor is a mechanism in the CPN software that is used to observe, inspect, control, or even modify a simulation of a CPN. A variety of monitors can be defined for a given net. Monitors can inspect both the markings of places and enabling of transitions during a simulation, and they can take appropriate actions based on the observations as well as extract relevant data. It is only after the validation step that one should use a simulation model to evaluate "what-if?" scenarios for implementing changes in the underlying actual operations. The interactive simulation tool available in the CPN Tools software can be used for incremental model verification. It allows a modeler to step through various markings and even set desired markings in an interactive manner. Using this facility, a modeler can check whether the desired specifications have been captured in the model.

6.7 Conclusion

Adoption of modeling and simulation in healthcare continues to be a challenging issue. One key barrier is buy-in from the stakeholders. Certainly, as noted by [7], simulation-based approaches can help improve patient safety and help better manage resources in a costly and constrained system like healthcare. Of particular importance is emergency care since there is data confirming that the number of emergency visits in the US is going up whereas the number of emergency departments providing such services is on the decline. Furthermore, COVID-19 forced many hospitals to re-evaluate and re-engineer their workflows but in absence of any simulation-based tools, there is no simple way to evaluate the impact of such changes. In our own experience, we have found a Colored Petri Nets-based approach to be less of a barrier for the stakeholders owing to a simple and visual graphical representation of the net model and its associated intuitive semantics. Furthermore, the free CPN Tools software with its visual editing and simulation capabilities renders it a very user-friendly environment for model development and analysis. We illustrated our approach by employing an operating room workflow and taking into account a variety of resources and constraints (room and staff availability) in a natural manner using the hierarchical CPN notation. The modular approach offered by the hierarchical CPNs allows a model to be constructed incrementally and, therefore, supports a very agile approach. We presented details of data collection and summarized our results. We provided an approach for the verification and validation of CPN models.

As we now move into a post-COVID world, healthcare organizations also need to find the new normal. At this stage there are too many unknowns and uncertainties; however, we do know that more and more focus is now being placed on virtual care models and possibilities. We contend that CPN modeling can be of great help and a strategic tool when trying to model and understand specific scenarios in healthcare contexts in which several divergent elements such as effectiveness, value, human input, and interactions must be tracked. In conclusion, given the advances in digital health and the availability of rich digital health data, we can make model-driven healthcare a reality to help improve patient safety and reduce cost.

References

1. Reid, P. P., Compton, W. D., Grossman, J. H., & Fanjiang, G. (2005). *Building a better delivery system: A new engineering/healthcare partnership. Tech. rep*. National Academies Press.
2. Jahangirian, M., Naseer, A., Stergioulas, L., Young, T., Eldabi, T., Brailsford, S., Patel, B., & Harper, P. (2012). Simulation in health-care: Lessons from other sectors. *Operational Research, 12*, 45–55. https://doi.org/10.1007/s12351-010-0089-8
3. Brailsford, S. C., Bolt, T., Connell, C., Klein, J. H., & Patel, B. (2009). Stakeholder engagement in health care simulation. In *Proceedings of the 2009 Winter Simulation Conference (WSC)* (pp. 1840–1849). https://doi.org/10.1109/WSC.2009.5429190
4. Eldabi, T. (2009). Implementation issues of modeling healthcare problems: Misconceptions and lessons. In *Proceedings of the 2009 Winter Simulation Conference (WSC)* (pp. 1831–1839). https://doi.org/10.1109/WSC.2009.5429192
5. Tako, A. A., & Robinson, S. (2015). Is simulation in health different? *Journal of the Operational Research Society, 66*(4), 602–614. https://doi.org/10.1057/jors.2014.25
6. Young, T., Eatock, J., Jahangirian, M., Naseer, A., & Lilford, R. (2009). Three critical challenges for modeling and simulation in healthcare. In *Proceedings of the 2009 Winter Simulation Conference (WSC)* (pp. 1823–1830). https://doi.org/10.1109/WSC.2009.5429186
7. Laker, L. F., Torabi, E., France, D. J., Froehle, C. M., Goldlust, E. J., Hoot, N. R., Kasaie, P., Lyons, M. S., Barg-Walkow, L. H., Ward, M. J., & Wears, R. L. (2018). Understanding emergency care delivery through computer simulation modeling. *Academic Emergency Medicine, 25*(2), 116–127. https://doi.org/10.1111/acem.13272
8. King, D., Emara, A. K., Ng, M. K., Evans, P. J., Estes, K., Spindler, K. P., Mroz, T., Patterson, B. M., Krebs, V. E., Pinney, S., Piuzzi, N. S., & Schaffer, J. L. (2020). Transformation from a traditional model to a virtual model of care in orthopaedic surgery: COVID-19 experience and beyond. *Bone & Joint Open, 1*(6), 272–280. https://doi.org/10.1302/2046-3758.16.BJO-2020-0063.R1
9. Richards, C. (2004). *Certain to win: The strategy of John Boyd, applied to business*. Xlibris, Corp.
10. Gehlot, V. (2019). From Petri nets to colored Petri nets: A tutorial introduction to nets based formalism for modeling and simulation. In *2019 Winter Simulation Conference (WSC)* (pp. 1519–1533). https://doi.org/10.1109/WSC40007.2019.9004691
11. Jensen, K., & Kristensen, L. M. (2015). Colored Petri nets: A graphical language for formal modeling and validation of concurrent systems. *Communications of the ACM, 58*(6), 61–70.
12. Jensen, K., Kristensen, L. M., & Wells, L. (2007). Coloured Petri nets and CPN tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer, 9*(3), 213–254.
13. Barkaoui, K., Dechambre, P., & Hachicha, R. (2002). Verification and optimisation of an operating room workflow. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences* (pp. 2581–2590). https://doi.org/10.1109/HICSS.2002.994236
14. Jensen, K. (1994). An introduction to the theoretical aspects of coloured Petri nets. In J. W. de Bakker, W. P. de Roever, & G. Rozenberg (Eds.), *A decade of concurrency. Lecture Notes in Computer Science* (Vol. 803, pp. 230–272). Springer.

15. Jensen, K., & Kristensen, L. M. (2009). *Coloured Petri nets. Modelling and validation of concurrent systems*. Springer.
16. Ullman, J. D. (1998). *Elements of ML programming*. Prentice-Hall.
17. Wells, L. (2006). Performance analysis using CPN tools. In *Proceedings of the 1st International ICST Conference on Performance Evaluation Methodologies and Tools* (pp. 59–68). https://doi.org/10.1145/1190095.1190171
18. Gunal, M. M. (2012). A guide for building hospital simulation models. *Health Systems, 1*(1), 17–25. https://doi.org/10.1057/hs.2012.8
19. Sargent, R. G. (2013). Verification and validation of simulation models. *Journal of Simulation, 7*(1), 12–24. https://doi.org/10.1057/jos.2012.20
20. Lane, D. C., Monefeldt, C., & Husemann, E. (2003). Client involvement in simulation model building: Hints and insights from a case study in a London hospital. *Health Care Management Science, 6*(2), 105–116. https://doi.org/10.1023/A:1023385019514