

# Applications of the Petri net to simulate, test, and validate the performance and safety of complex, heterogeneous, multi-modality patient monitoring alarm systems

E. B. Sloane<sup>1</sup>, V. Gelhot<sup>2</sup>

<sup>1</sup>Department of Decision and Information Technologies, Villanova University, Villanova, PA, USA

<sup>2</sup>Department of Computer Science, Villanova University, Villanova, PA, USA

*Abstract*—This research is motivated by the rapid pace of medical device and information system integration. Although the ability to interconnect many medical devices and information systems may help improve patient care, there is no way to detect if incompatibilities between one or more devices might cause critical events such as patient alarms to go unnoticed or cause one or more of the devices to become stuck in a disabled state. Petri net tools allow automated testing of all possible states and transitions between devices and/or systems to detect potential failure modes in advance. This paper describes an early research project to use Petri nets to simulate and validate a multi-modality central patient monitoring system. A free Petri net tool, HPSim, is used to simulate two wireless patient monitoring networks: one with 44 heart monitors and a central monitoring system and a second version that includes an additional 44 wireless pulse oximeters. In the latter Petri net simulation, a potentially dangerous heart arrhythmia and pulse oximetry alarms were detected.

## *Keywords*

Petri nets, software validation, system validation, medical devices, medical information systems

## I. INTRODUCTION

Advances in IT and related fields are rapidly changing the way health care is managed and delivered. Many hospitals are rapidly moving to extensive integration of both patient care and general business networks and information systems, using wired and wireless LAN architectures (Briggs 2004, Cohen 2001, Sloane 2001, Sloane 2002). In addition, major medical device companies like General Electric and Siemens have designed patient monitoring systems that automatically reconfigure themselves for different hospital applications (Farah 2003, Várady 2002). A major, multinational effort called Integrating the Healthcare Enterprise (IHE) is underway to facilitate multi-vendor interconnection of medical devices and hospital information systems, as well, and it is also embracing the emerging electronic health record initiatives (IHE/Cardiology, IHE/Hospital, IHE/Laboratory, IHE/Radiology 2004).

On the one hand, such advances have been a boon, but on the other hand this has meant that we are moving towards integration and automation of increasingly complex and critical processes and tasks with little or no human oversight or intervention. The risks of system failures or defects can be very serious; government reports have documented that as many as 90,000 American patients may be injured or killed by medical errors each year, and complex systems with defects could only make the problem worse (IOM). Although medical

device manufacturers must validate their software-based products to FDA standards, no such requirements exist for hospitals.

The central problem with hybrid, multi-vendor, multi-generational devices and information systems is that the complexity of the interaction of the underlying systems multiplies with the introduction of each new subsystem. Safe and reliable operation of such systems depends heavily on their ability to function as desired, but systems of validation are not presently available in healthcare.

One way that has been used to ascertain desired behavior of a system in other industries is to make use of a formal modeling and validation/verification techniques. There are a good variety of such techniques that have been developed in a different context (Gehlot 1988, Reisig 1985). This current research is a first attempt applied such techniques in health-care settings and to show that there are benefits to be drawn from such an approach. In particular, we have tried to show by means of a simple example how Petri Nets (Reisig 1985) may be used to model healthcare scenarios.

Petri nets are a good tool for modeling systems with interacting concurrent components. The fundamental idea behind such modeling is that systems are composed of separate interacting components. Each component has its own state of being and its state may change over time via interactions. Furthermore, as a modeling tool, Petri nets have the following advantages:

- Flexibility: there are a wide range of extensions to suit different needs. For example, timed and stochastic extensions of Petri nets make them suitable for performance analysis.
- Adaptable: since they are based on very few key abstract ideas, they are easily adaptable to a variety of modeling domains.
- Visual: Petri nets are a graphical modeling notation. This makes them easy to understand and work with.
- Analysis tools: there are a good number of verification/validation tools available for Petri nets.

In this paper, we will a) show how a specific tool, known as Petri Nets, is used to document interacting devices or systems, b) how a Petri net can be used to document a clinical device that is part of a larger hospital information system and b) discuss the next step of using the available verification/validation tools detect/prevent/avoid faults and failures. Automated tools are crucial to the healthcare domain since a manual verification/validation of a system with even few tens of states becomes daunting. Fortunately, as of the writing of

this paper, about 50 different automated tools for Petri Nets are currently available.

In the next section we give a formal definition of a Petri Net and show how a verification/validation tool can help detect design/interface problems. Following this we take a simple example from the medical/clinical domain and show how we can represent it using Petri nets and how we would approach verification/validation of the system prior to deployment. We present our conclusion and future work in the last section.

### A. Petri Nets

To understand the definition of a Petri net, it is helpful to have a very abstract level view of the workings of a system. Such a view stipulates existence of "state-like" objects ( $S$ ) and "event-like" objects ( $T$ ) and dependencies between these objects ( $F$ ). The basic idea being that "any" phenomena or system can be described in terms of "cause and effect". The state-like objects become the cause for the event-like objects to "occur" and the effect of which is "another" state-like object. Thus, A Petri net consists of the following:

- A finite set of *states* or *places* (denoted  $S$ )
- A finite set (disjoint from  $S$ ) of *transitions* or *events* (denoted  $T$ )
- A finite subset of  $(S \times T) \cup (T \times S)$  called the *flow relation* or the *dependency relation* (denoted  $F$ )
- A mapping from  $S$  to natural numbers (including infinity) called marking (denoted  $M$ ), i.e.,  $M : S \rightarrow \mathcal{N}$

There is an elegant graphical representation of nets. In this representation, it is customary to denote places by a circle ( $\circ$ ) and transitions by a box/bar ( $\dashv$ ). The flow relation is depicted by directed arcs ( $\rightarrow$ ) joining places and transitions and vice versa. In the graphical notation, markings are represented by a bullet ( $\bullet$ ) in circles representing the places of the net. In Petri net parlance, these bullets are called *tokens*. The distribution of tokens represents the combined current state of the underlying system being modeled. An example Petri net is depicted in Figure 1.

In this example, the Petri net consists of two places ( $s1$  and  $s2$ ) and three transitions ( $t1$ ,  $t2$ , and  $t3$ ). The place  $s1$  has one token. In modeling any general system, the places and transitions would be given some meaningful interpretation. For example, let us give the following interpretations to this net:

- $s1$ : ready to accept coin
- $s2$ : coin inserted
- $t1$ : insert coin
- $t2$ : accept coin
- $t3$ : reject coin

With this interpretation, the Figure 1 net captures the behavior of a simple vending machine.

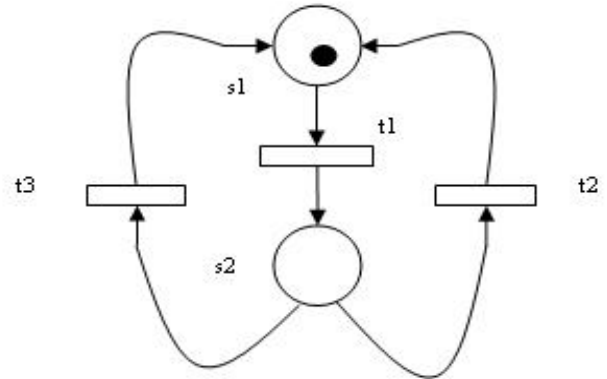


Figure 1 Example of a simple Petri net

The analysis of a Petri Net model can be divided into a) reachability analysis and b) performance analysis. The former is useful in situations where only functional correctness is of importance whereas the latter is useful if the temporal behavior of the underlying system is crucial. In our domain, both types of verification/validation would be essential and Petri nets, therefore, give a unified setting to carry out both types of analyses.

For the simple Petri net in Figure 1 above, an example of functional requirement may be that the machine will not simultaneously be in the state "ready to accept coin" and "coin inserted." In terms of reachability, this would entail establishing that the state  $\{s1, s2\}$  is not reachable from the state  $\{s1\}$ . Of course, for the simple net in the example, this is trivial to establish, but a complex net would require a more powerful analysis tool. An example of temporal behavior in this case would be to stipulate that the machine will be ready to accept another coin within  $t$  seconds of inserting a coin. Such an analysis would require introducing timed transitions and then doing the performance analysis. Again, in the simple example here, this is a trivial task but a complex example would require use of an automated tool.

### B. Using available Petri Net tools

In our preliminary exploration, we will be using the HPSim ([http://www.winpesim.de/petrinet/e/hpsim\\_e.htm](http://www.winpesim.de/petrinet/e/hpsim_e.htm)) tool. It is a semi-automated freely available tool with graphical editor. The tool allows token game animation and exploration of state spaces. For a fuller analysis, JARP can export the data to delimited format files that are easily translated to comma separated format that is easily imported into Excel. HPSim is a free tool, with a complete basic graphical interface. It accepts the following Petri net models:

- Place/transition nets (numeric markings, no time information);
- Timed nets (firing intervals  $[t_{min}, t_{max}]$  associated to transitions);
- Extended timed nets (probability function associated to firing intervals).

In addition to this tool, there are many other (commercial as well as free) tools available. A complete list of Petri net software and system tools can be found at: [http://www.daimi.au.dk/PetriNets/tools/complete\\_db.html](http://www.daimi.au.dk/PetriNets/tools/complete_db.html)

C. Application of the Petri Net model for healthcare monitoring

Our goal is to begin using the Petri net tools to simulate medical situations in order to detect and prevent situations that could cause risks to a patient and to help validate new system configurations as clinical information systems are designed and modified. There are many possible ways to complex ways to connect multiple patient monitors together, but a simple example can help illustrate the basic process. Figure 2 shows a simple Petri net that could be used to model a Portable Patient Monitor connected to a Central Station Monitor. The Petri net can be expanded to support any target number of portable devices. Each device does not have to be the same, nor is there any limit to the number of different devices that be included in the model. To build the Petri net simulation, all states and transitions for each device will be entered. Then, the Petri net tool can be used to identify problems, such as these five:

1. missed alarms
2. simultaneous alarms
3. ambiguous, stuck, or hung system states
4. false/confused alarms
5. missing or mis-mapped alarms

D. Development and testing of a Petri net for clinical information systems

In order to apply the Petri net tools in a clinical setting, a working model must be constructed to simulate a real, multi-hospital central monitoring system.

As a representative example, we modeled a new, 44-station wireless heart monitoring system that was recently described in the literature (American Health Consultants 2003). Figure 3 shows an HPSim Petri net model for such a system.

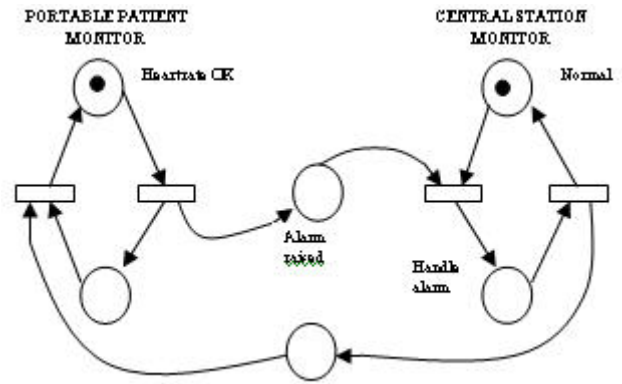


Figure 2 Petri net example of patient monitoring system

The Petri net model created in Figure 3 was created with the following parameters: 44 heart monitors that could send one of two possible alarms (arrhythmia or low battery) to the central station; out of the 44 heart monitors, arrhythmia alarms can occur in a random uniform distribution ranging from 2 to 20 minutes; the low battery alarms can occur in a random uniform distribution ranging from 100 to 500 minutes; and there are two nurses available to review and reset the central station alarms at a random uniform distribution ranging from 1 to 5 minutes. For the purpose of illustration, a 10,000 minute simulation was run (approximately 1 week). In this configuration, the central station operators always cleared the alarms quickly enough to avoid any missed or delayed arrhythmia alarms.

The Petri net model was then revised to reflect the hospital's stated expansion future plan by adding 44 pulse oximetry monitors to the system. The addition of pulse oximetry could create a very different demand on the central station because some pulse oximetry systems alarm very frequently if the patient's are moving around. This is a known limitation of many pulse oximeters because the patient's movements change the position of the light-based sensors. The question we were interested in answering was whether the system that was shown in Figure 3, with two nurses responding to alarms, could

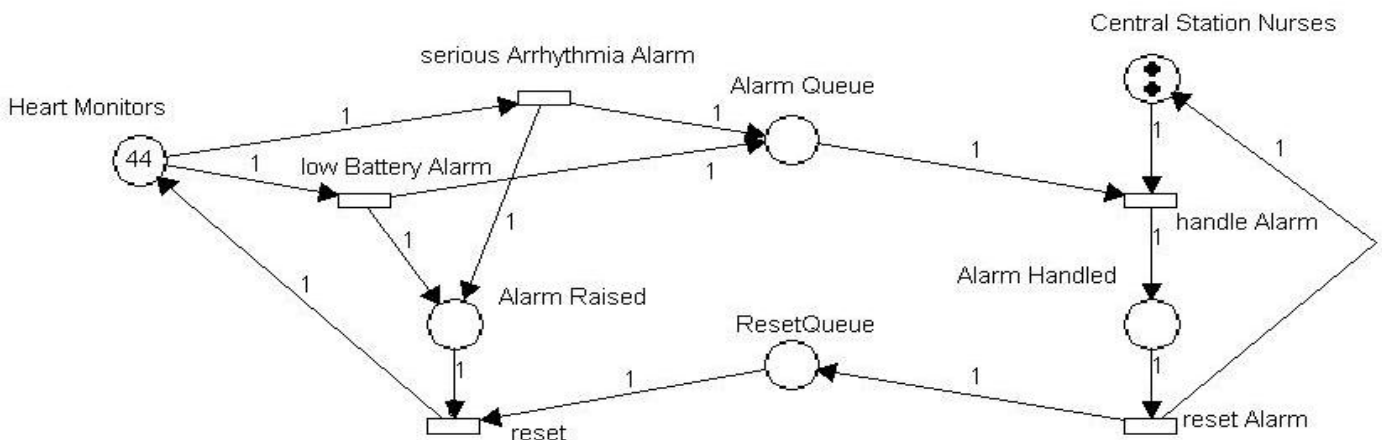


Figure 3 Petri net model for 44-station wireless heart monitoring system

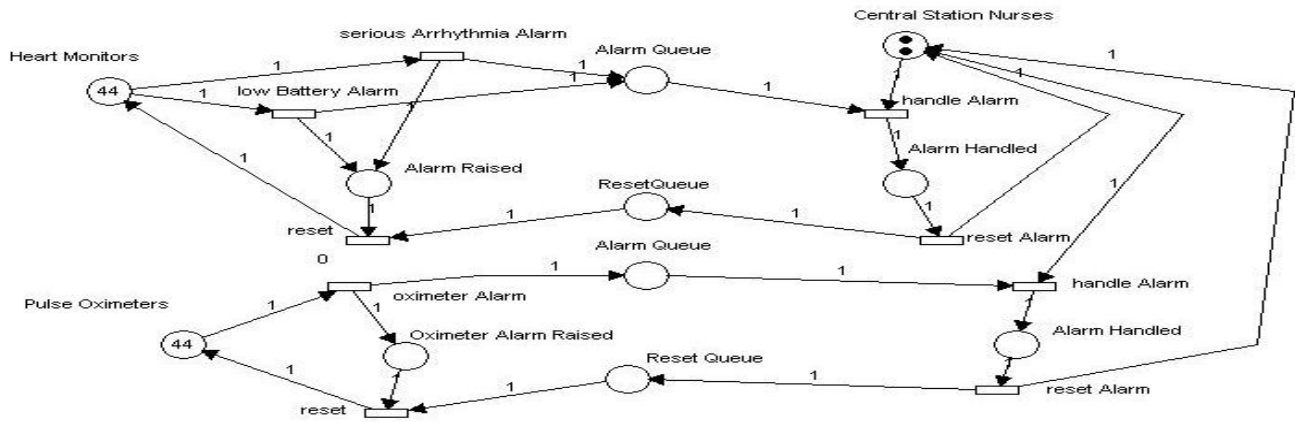


Figure 3 Mixed, multimodality patient monitoring Petri net with pulse oximeters and heart arrhythmia monitors

handle the additional burden of 44 pulse oximeters, as shown in Figure 4.

The same initializing conditions were used for the heart monitors and the central station. However, although the pulse oximeters were configured to send alarms every 1-2 minutes, they were allowed 5-10 minutes to clear the pulse oximeter alarms. This configuration was used to reflect the possible intention of using the pulse oximeter for relatively stable patients who are well enough to ambulate throughout areas of the hospital. Because pulse oximetry is often considered a late indicator of serious distress, we made the assumption for this model that the patients were not high-risk patients who are likely to require immediate intervention. We elected to leave the heart monitor response the same, however, as a dangerous arrhythmia might need immediate intervention to save a patient's life.

#### IV. ANALYSIS

To analyze the purely functional aspect of our model, we built the reachability graph and did the accessibility analysis using the JARP/ARP tools [1, 2] and found the underlying net to be live. We then introduced the possibility of signal loss. This, as expected, made the net not live thereby implying a possibility of a deadlock situation.

We then carried out a performance analysis using HPSim tool [3]. When the Figure 3 model was run in simulation mode for 10,000 minutes, the heart alarms were never overloaded. However, for the multi-mode heart/oximetry model in Figure 4, after a few dozen minutes, virtually 100% of the oximetry alarms were queued up, representing missed or grossly delayed alarm situations!. In fact, a very close examination of the simulation data shows that one heart alarm went unanswered for several minutes, and one of the pulse oximetry alarms is not handled for about five minutes, implying thereby a probable endangerment of the associated patient(s).

#### V. CONCLUSION

We argued the use of formal modeling techniques to evaluate and ensure safety of patient monitoring systems.

Our belief is that a lot of such systems are being built and used in a very ad hoc manner. We showed the use of Petri net in modeling and analyzing a typical patient monitoring system. Our analysis showed how an overload situation can occur, and as a result, how critical alarms may be missed. The use of Petri net based modeling allows us to create different scenarios easily that can then be analyzed and compared.

#### REFERENCES

- [1] Health Consultants (2003) Patient-focused care boosts patient satisfaction while enhancing safety: wide variety of technology brings care directly to the patient, *HealthCare Benchmarks and Quality Improvement*, October.
- [2] Briggs, B. (2004) Hospitals Bring Medical Devices into the I.T. Loop, *Health Data Management*, February.
- [3] Cohen, T. (2001) The convergence and integration of medical and information technology. Engineering in Physical Science and Medicine Annual Meeting. Fremantle, Australia.
- [4] Farah, G. (2003) Current and future patient monitoring technologies. ACCE Clinical Engineering Symposium at the Annual Meeting of the Association for the Advancement of Medical Instrumentation. San Diego, CA.
- [5] Gehlot, V. (1988) Performance Specification and Livelock Detection/Correction of a Protocol Using Timed Petri Nets, *Proc. International Conference on Communications*, pages 1286-1290.
- [6] IHE/Cardiology. American College of Cardiology information system IHE program at [www.acc.org/media/news/feb03/initiative.htm](http://www.acc.org/media/news/feb03/initiative.htm).
- [7] IHE/Hospital. IHE/HIMSS Hospital information system IHE program. Available at [www.himss.org/ihe](http://www.himss.org/ihe).
- [8] IHE/Laboratory. IHE/France Laboratory information system IHE program. Available at [www.gmsih.fr/fr/ihe.htm](http://www.gmsih.fr/fr/ihe.htm).
- [9] IHE/Radiology. IHE/RSNA Radiology information system IHE. Available at [www.rsna.org/ihe](http://www.rsna.org/ihe).
- [10] Institute of Medicine. (1998) To err is human: building a safer health system Linda T. Kohn, Janet M. Corrigan, and Molla S. Donaldson, editors.
- [11] Reisig, W. (1985) Petri Nets. Springer-Verlag, New York.
- [12] Sloane, E. B. (2001) The Rebirth of Clinical Engineering in the Information Age. *Biomedical Instrumentation*. March/April.
- [13] Sloane, E. B. (2002) eHealth: IT and the Reinvention of Healthcare. Association of Information Technology Professionals, Montgomery County, PA.
- [14] Várady, P., Benyó, Z. and Benyó, B. (2002) An Open Architecture Patient Monitoring System Using Standard Technologies. *IEEE Transactions of Information Technology in Biomedicine*, Vol. 6, No. 1, pages 95-98.