Petri net modeling for probabilistic safety assessment and its application in the air lock system of a CANDU nuclear power plant

Andrew LEE, Lixuan LU*

Faculty of Energy Systems and Nuclear Science, University of Ontario Institute of Technology, Oshawa, ON, L1M 1E3, Canada

Abstract

Historically, fault trees are extensively used in Probabilistic Safety Assessment (PSA) to model and evaluate the probability and consequence of failures of complex engineering systems, such as nuclear power plants. Scenarios of hypothetical accidents resulting in severe core damage can be developed. Petri net is another modeling technique that offers many advantages when comparing with fault trees, such as is ability to represent the time sequence of the events along with their duration. In this research, Petri net theory is extended to model system failures. The transformations required to model logic gates by Petri nets are explored and examples provided. Methods for qualitative analysis for Petri nets are presented. The application of Petri net in the airlock system of a Canada Deuterium Uranium (CANDU) reactor is explored. It is further demonstrated that Petri net can be used in both coherent and non-coherent systems.

© 2012 The Authors. Published by Elsevier Ltd. Selection and/or peer-review under responsibility of the Beijing Institute of Technology. Open access under CC BY-NC-ND license.

Keywords: Petri net; fault tree; coherent; non-coherent; CANDU nuclear power plant

Nomenclature

| PT       | Petri net five tuple |
|----------|----------------------|
| p        | a finite set of places |
| T        | a finite set of transitions |
| M(0)     | initial marking of a Petri net |
| M(k)     | Petri net marking at an arbitrary time instant k |

Subscripts

| n       | number of positions |
|---------|---------------------|
| i       | number of transitions |

1. Introduction

Nuclear power plants are highly regulated from design, construction, and operation to decommissioning. Engineers must comply with stringent regulations to limit the risk of the release of radioactivity to the public. The objectives of nuclear safety are to operate in a safe manner and protect the public and environment. Release of radioactive material in the event of a design basis accident is strictly prohibited through rules and regulations.

* Corresponding author. Tel.: +1-905-721-8668; fax: +1-905-721-3046.
E-mail address: Lixuan.Lu@uoit.ca

1877-7058 © 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license,
doi:10.1016/j.proeng.2012.08.113
The deterministic approach is widely used in the design of nuclear reactors. This approach attempts to ensure that the plausible accident scenarios are taken into account, and that the monitoring systems, and engineered safety systems will be capable to contain any accident and prevent the release of radioactive material. This approach is based on two principles referred to as leak tight barrier and the concept of Defense-in-Depth. Leak tight barriers consist of fuel cladding, the primary reactor coolant system, and the containment building between the radioactive source and the public. Defense-in-Depth assumes accidents may still occur from equipment failures and human errors despite engineering designs and safety, and therefore systems are designed and installed to limit the consequences for both the public and environment. Defense-in-Depth incorporates several stages including: prevention and surveillance, protection, and safeguard.

Despite these provisions it was evident in the 1960’s that the deterministic approach had its limitations and new safety assessment techniques should be required to ensure the safeguard of the public and environment. The Three Mile Island-2 (TMI) accident in the United States (US) in 1979[1], and the Chernobyl accident in the former Soviet Union in 1986[2] demonstrated the necessity of probabilistic methods in nuclear power plant safety assessment. Probabilistic Safety Assessment (PSA) has seen much attention and development since, and is now an integral part of nuclear power plant operation and design. Two documents, one before the TMI accident, often referred to as WASH-1400[3]; and one after the Chernobyl accident, often referred to as NUREG-1150[4], are two key milestones in the PSA development. WASH-1400 demonstrated that Small Loss of Coolant Accident (SLOCA) could result in severe reactor accident, although SLOCA is not considered a design basis accident. This was unfortunately proved true by the ill-fated TMI reactor. Probabilistic Safety (PSA) Assessment has since gained much attention and the US Nuclear Regulatory Commission (NRC) performed PSA for five commercial light water reactors and the results were concluded in NUREG-1150.

Fault tree is the major modeling tool in PSA, together with event tree. Fault tree analysis was developed by H.A. Watson of Bell Laboratories, while in a contract with the U.S. Air Force to study the Minuteman Launch Control System. Since its discovery in the 60’s, FTA has been applied to many applications in the nuclear industry. Traditional FTA has always had a crucial limitation which is its ability to evaluate FT when a system is too large and the number of states become overwhelming; much like the massive systems in a nuclear power plant. Limitations of the 60s-80 eras were computer processing speeds. Thanks to the technological advances in microprocessor technology, they no longer pose a problem. The synthesis of the results is generally presented in a graphical model organized by the logic of the Boolean algebra and its symbols [5].

Although fault tree analysis is one of the most widely used modeling methods for reliability analysis, Petri net modeling also offers many advantages [6]. When comparing three models to accident scenario for hazardous systems (Ammonia import (ship) and refrigerated storage), Nivoliannitou concluded that Petri nets offer the advantage of event dependencies and event interactions, preconditions for the realization of each event, ability to represent the time sequence of the events along with their duration, and modeling of assumptions for dynamic events.

The objective of this research is to advance the understanding of Petri net application for reliability analysis, and demonstrate their ability for failure analysis in the nuclear industry. It also demonstrates how Petri nets can transform into logic gates and obtain cut set for both coherent and non-coherent systems. The paper is organized as follows: a literature review on fault tree and Petri net theory is introduced in Section 2. Construction of Petri nets, transformation from fault trees to Petri nets, and the application of Petri net in PSA are presented in Section 3. The application of Petri nets in the airlock system of a Canada Deuterium Uranium (CANDU) reactor is explored is Section 4, both for coherent and non-coherent systems. Section 5 concludes the results and provides recommendation for future research.

2. Fault tree and Petri net

This section provides a basic introduction on fault tree and Petri net construction.

2.1. Fault tree

A fault tree provides a structured approach to determining the probability of failure in a complex system. This approach also illustrates the minimum set of events that can cause the failure of a system. In order to evaluate the industrial risk, it is necessary to have an estimation of the accidents probability. The analysis is restrained in a particular undesired event (accident or incident), defined as the top event. It is operated by means of graphic modeling allowing the visualization of the possible combinations of malfunction and wrong actions that can generate it. The synthesis of the results is generally presented in a graphical model organized by the logic of the Boolean algebra and its symbols. Common symbols used for constructing fault trees are shown in Table 1[7].
2.2. Petri net

The Petri net is a directed graph consisting of two types of nodes, called places and transitions. Weighted and directed arcs connect places to transitions, or vice versa. Systems are modeled as a set of conditions and events. Places represent conditions, and transitions represent events. Transitions have a set of input and output places which represent the preconditions and post-conditions of the transition. The state of a net is modeled by the presence or absence of a token in the places. The tokens in a place are referred to as the marking of the place. The initial marking represents the initial condition or state of the net. The states change by the firing of transitions, which depicts the events occurring. An event occurs only when the preconditions are met and is represented by an enabled transition. The firing of a transition changes the marking of its input and output places, modeling a change in its precondition and post-conditions [8].

A Petri Net is a five-tuple [9]:

\[
 PN = \{P, T, Wpt, Wtp, M(0)\}                                                                 (1)
\]

Where,

\[
 P = \{p_1, p_2, \ldots, p_n\} \quad n \geq 0                                                 (2)
\]

is a finite set of places, and

\[
 T = \{t_1, t_2, \ldots, t_i\} \quad i \geq 0                                                  (3)
\]

is a finite set of transitions.

There are two weight functions, Wpt and Wtp, which attach a positive integer weight to each arc of the net connecting places to transitions (pt) and transitions to places (tp), respectively. The initial marking is represented by \( M(0) = [m_1(0), m_2(0), \ldots, m_n(0)]^T \) and is a function from the set of places to non-negative integers. The marking at an arbitrary time instant \( k \) is represented as \( M(k) = [m_1(k), m_2(k), \ldots, m_n(k)]^T \) and can also be referred to as the number of tokens in each place.

Petri nets consist of four basic elements: places, transitions, tokens and arcs[7]:

- \( \bigcirc \): Place, drawn as a circle, denotes event
Places represent a condition in the process. Transitions represent the instantaneous, stochastic or time-based changes in the model. Transitions can be immediate, deterministically time-delayed, or time-delayed based on a probability distribution defined by the user. Tokens represent objects in the model. In graphical user interface (GUI) applications, tokens are represented as black solid circles and in more sophisticated applications, such as color Petri nets, the circles can take on various colors to signify the age. A transition allows the movement of a token and is said to have “fired” when this happens. Arcs determine the path that tokens take throughout the model. Arcs can either enable or inhibit movement in the model, depending on their use.

The graphical construction of a Petri net involves the understanding of how the places, transitions, tokens and arcs will interact. The basic interactions involved are Sequential, Synchronization and Merging Transitions. The most basic model of the Petri net is the state transition from input place $p_1$ to output place $p_2$. Figure 1 shows an initial state $M(0) = [1, 0]^T$. Figure 2 represents the firing of a token from $p_1$ to $p_2$ based on transition $t_1$.

After firing, $M(1) = [0,1]^T$

3. Transformation from fault tree gates to Petri net transitions

Logic gates in fault trees can be transformed to corresponding Petri nets. Figure 3 show the Petri net representation of AND, OR and NULL gates [7].
The following is an example of converting a fault tree represented by logic gates into an equivalent Petri net. Figure 4 represents a 2 out of 3 voting logic gate, and is constructed using AND and OR gates. The fault tree in Figure 4 was created using the Software Relex 2011 Evaluation. Figure 5 is the Petri net model of a 2 out of 3 voting gate where all logic gates have been converted to their Petri net equivalent. Working from Node G1, we notice that G1 is an AND gate and will become true only when Basic Event B1 and Node G4 are true. G4 is a subset of G1 and it is an OR gate which is true if any of its inputs, B4 or B6, are true. The transformation of gates start from bottom up, therefore we can start with G4. The OR gate will be modeled by 2 basic event places (B4, B6), each with an arc to their transitions and an arc from the two transitions to place G4. Then we move on to Node G1, which is an AND gate. They can be represented by one transition and two places (B1, G4) with arcs pointing to the transition and another arc pointing to place G1.
4. Application of Petri net in the airlock system

An airlock is a vessel in the containment wall of the reactor vault with two doors. One door opens to the inside reactor vault and the other door to the outside of the reactor vault. The Airlock, which is considered part of the Negative Pressure Containment (NPC) system, is required to perform the safety function of maintaining the containment boundary. At least one airlock door must be closed, with sufficient pressure maintained in the seal, to ensure preservation of the containment boundary. This implies that both sets of seals on each door are qualified. Airlock door seals may be deflated only when the door is in use. All airlocks and transfer chambers have valves to equalize the pressure across the service and containment doors before the doors are operated.

Following all design basis accident conditions, including all Loss of Coolant Accidents (LOCAs), flooding events, fuel handling system failures, Loss of Emergency Coolant Injection (LOECI) and Secondary Side Line Break (SSLB) events, the credited safety function of the Airlock door seals is to remain inflated with sufficient seal pressure to ensure a containment boundary. Electromechanical indicators on the containment panel in the control room are operated by pressure switches monitoring the pressure in the air seals and by limit switches monitoring the state of the doors. The components associated with the door seals must maintain a pressure boundary and not leak externally for up to three months. Maintaining pressure boundary is crucial to remain compliant with the safety objectives in place for nuclear power plants. A potential leak or equipment failure at this point would prove costly.

4.1. Airlock safety system during a design basis accident

In this case, the scenario involves a design basis accident occurred in 2011. The airlocks seals which have a credited safety function to maintain pressure boundary have switched to back up air supply tanks. For a simplified airlock system, the top event is failure to maintain pressure boundary. The causes for this failure are: failure of the equalizer valve, door fails to close or latch not locked, and the inflatable seals fail to perform their credited safety feature. Equalizer valves are designed to equalize the pressure between the reactor bay and the service side. They function by opening vents in the equalizer valve allowing airflow to enter the airlock chamber until pressure is equalized and then the vents in the equalizer valve will close. The equalizer valves fail when the gear box fails, which prevents the vents from opening or closing and allows constant flow between the reactor bay and service side. To meet the safety function for maintaining pressure boundary, the airlock doors must be closed by a latch, or else a beacon light and annunciation to the Main Control Room (MCR) will occur. If the door is not properly closed and latched, the equalizer valves and seals would not be available to provide their safety functions either. In the event of a design basis accident, the airlock door seals are designed to inflate thus creating the necessary seal to maintain pressure boundary. During normal operation, the door seals are inflated via the instrument air system. During a design basis accident however, the inflation of the seals are switched to the back-up air supply tank. This presents a potential for system failure because we have a limited air supply compared to during normal operation when a constant supply came from the instrument air system.

Therefore, the contributing factors to the airlock system failures can be summarized as follows:

- Seals deflating during the course of the accredited safety function
- Back up tank being empty
- Cracks in the seal leading to loss of air
- Pipe leak leading to the loss of air
- Valve failure, including check valves allowing back flow or relief valves releasing prematurely.
- Seals do not inflate after a design basis accident
- Back up tanks don’t engage
- Major pipe leak
- Valve failure, including check valve not allowing positive flow and relief valves constantly venting.

Figure 6 and 7 models the airlock system in Fault Tree and Petri net format respectively. The variables represent the failure events are listed below:

- G1 – Gearbox fails
- E1 – Exhaust pipe doesn’t close
- D1 – Door fails to close/lock
- S1 – Crack(s) in Seal
- V1 – Valve failure; Check valve preventing flow or allowing back flow, Relief valve prematurely releasing or not closing, any valve that operates outside of the design function causing leaks
- P1 – Leaks in the piping system
P2 – Major leaks in the piping system
T1 – Back up tank is empty
T2 – Back up tank fails to engage

After performing qualitative analysis to determine the minimal cut sets for Figure 6, the following minimal cut sets are obtained.

1st Order: [D1], [P2], [T2], and [V1]
2nd Order: [E1, G1], [P1, T1], and [S1, T1]
4.2. Airlock system maintenance

In the following case an airlock system is scheduled for routine maintenance during an outage. It is discovered that the Pressure Regulating Valve (PRV) is damaged and therefore needs to be replaced. In the warehouse there are no other spares of the same make and model. The engineering department finds another model which has the same pressure range, maximum inlet pressure, and silicon O-ring and determines that it’s an item equivalency. What the engineering and their supervisor failed to realized was the previous PRV was a non-relieving type while this PRV is a relieving type. The detectors were set so that any minor fluctuation in reference to the set point would trigger the Main Control Room (MCR).

In the following scenario a NOT function can be used because its use means that a component NOT failing contributes to the system failing. While the pressure regulating valve is functioning as designed, its working condition will cause the system to annunciate, thus meeting the criteria for failure. The top event for system is the detection of a leak in the system.

In this system we have two pressure transmitters (P1 and P2) used to detect leakages in the seal. The signals from the pressure transmitter are sent back to a Monitoring System (MS) in the MCR. On receiving a signal which prevents the seals’ ability to maintain pressure boundary, the system will respond by:

- Informing operator in the MCR (A)
- Isolate the seals to prevent further loss of air (CV)
- Isolate the backup air receiver and switch to a high pressure tank (BT)

Fig. 8 models the airlock leak detection system. A qualitative analysis was performed to obtain the following minimum cut sets: [CV, BT], [P1, P2] and [MS].
Fig. 8. Seal leak detection system Petri net.

5. Conclusions and future research

In this paper, Petri net theory is introduced with an extension to the original definition to convert nets into logic gates which bridge the gap between Fault Trees (FT) and Petri nets for non-dynamic systems. Examples demonstrated how to construct a Petri net using logic gates or convert an existing FT into a Petri net. In the case studies, the Airlock systems were introduced, which has an accredited safety function to maintain pressure boundary. Through the understanding of Petri net construction we were able to model the system, and bridge the gap between FT analysis and Petri nets with gate equivalences. Qualitative analysis was performed to obtain minimum cut sets. Furthermore, a modification to the Airlock system was constructed where the use of the NOT gate was necessary, and qualitative analysis was conducted for this non-coherent system.

It is recommended that future research should investigate Petri nets’ role with dynamic systems. Dynamic systems encompass the variables of time, failure and repairs, which more accurately describe systems in practice. Petri nets are limited by its software and the unavailability of code to model dynamic gates. While it is possible to use basic symbols to model dynamic gates, the system would immediately become too large and cumbersome for simulation and analysis. The first step to streamlining Petri nets as a viable tool for reliability analysis is to develop a software library for the proper
conversions. For too long has Petri nets only been thought of as a simulation tool for queuing and network systems. The preliminary results in this paper show that it can also be a powerful tool, with the ability to simulate and model systems, to provide a frame by frame account of operation. Once the areas of stable software and dynamic conversions are resolved, Petri nets will be able to model and simulate system behaviour with higher accuracy.

Acknowledgements

The comments and feedback from faculty members at the University of Ontario Institute of Technology (Dr. Rachid Machrafi, Dr. Dan Zhang and Dr. Igor Pioro) are greatly appreciated.

References

[1] Collier J. G., Davies L. M., 1980. The accident at Three Mile Island, Heat Transfer Eng. 56(1).
[2] Ahearne J. F. 1987. Nuclear power after Chernobyl, Science 236, p.673.
[3] U.S. Nuclear Regulatory Commission, Reactor Safety Study—an Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants, WASH-1400, 1975.
[4] U.S. Nuclear Regulatory Commission, Severe Accident Risks: an Assessment for Five U.S. Nuclear Power Plants, NUREG-1150, 1990.
[5] Kumamoto H, Henley E., 1996. Probabilistic Risk Assessment and Management for Engineers and Scientists, 2nd ed. IEEE Press.
[6] Nivoliannitou Z.S., Leopoulos V.N., Konstantinidou M., 2004. Comparison of Techniques for Accident Scenario Analysis in Hazardous Systems, Journal of Loss Prevention in the Process Industries, 17, p. 467-475.
[7] Liu T.S., Chiou S. B., 1997. The Application of Petri Nets to Failure Analysis, Reliability Engineering and System Safety, 57, p. 129-142.
[8] Bowden F. D. J., 2000. A Brief Survey and Synthesis of the Roles of Time in Petri Nets, Mathematical and Computer Modeling, 31, p. 55-68.
[9] Reisig W., 1985. Petri Nets: an Introduction, Springer-Verlag.