〈Technical Note〉

DISCRETE EVENT SYSTEM SIMULATION APPROACH FOR AN OPERATION ANALYSIS OF A HEADEND PROCESS FACILITY

HYO JIK LEE*, SUNG HYUN KIM and BYUNG SUK PARK
Korea Atomic Energy Research Institute
1045 Daedeokdaero, Yuseong, Daejeon 305-353, Korea
*Corresponding author. E-mail : hyojik@kaeri.re.kr

Received August 1, 2008
Accepted for Publication December 31, 2008

This paper introduces facility operation modeling and simulation based primarily on a discrete event system modeling scheme. Many modern industrial facilities are so complex that their operational status cannot be estimated by simple calculations. In general, a facility can consist of many processes and transfers of material between processes that may be modeled as a discrete event system. This paper introduces the current status of studies on operation modeling and simulation for typical nuclear facilities, along with some examples. In addition, this paper provides insights about how a discrete event system can be applied to a model for a nuclear facility. A headend facility is chosen for operation modeling and the simulation, and detailed procedure is thoroughly described from modeling to an analysis of discrete event results. These kinds of modeling and simulation are very important because they can contribute to facility design and operation in terms of prediction of system behavior, quantification of facility capacity, bottleneck identification and efficient operation scheduling.

KEYWORDS : Modeling and Simulation, Discrete Event System, Operation Analysis, Headend Process Facility

1. INTRODUCTION

Spent nuclear fuel from pressurized water reactors (PWRs) accumulates annually. Accordingly, international interest has been focused on research into means of reprocessing or disposing of spent fuel. The Republic of Korea (ROK) started studying what is called pyroprocessing [1] about 10 years ago and has been acknowledged as one of the leading countries in this field, along with the US. The ROK believes that pyroprocess technology is a competitive alternative to the aqueous process and that it will solve the problem of spent fuel accumulated in the ROK. Pyroprocess technology separates transuranic elements from spent nuclear fuels and then recycles them within a sodium-cooled fast reactor that the ROK considers appropriate for the proposed fuel cycle concept. Both domestically and internationally, there is an increasing need for the construction of a spent fuel reprocessing facility that can demonstrate the feasibility of the technology. However, construction of such a facility will be expensive, so a long-term schedule must precede construction of the facility. Detailed design and optimal layout of the facility processes through modeling and simulation can save money and time, as well as contribute to verification of facility design before construction and to operation scheduling, annual throughput estimation, and system diagnosis after construction. In terms of systems classification, operation of a nuclear facility is macroscopically related to a discrete event system (DES) rather than to a continuous variable dynamic system (CVDS). In other words, a nuclear facility operation is very similar to a system that is driven by asynchronous events rather than by time change, and the current state in a nuclear facility can be described with discrete states rather than with continuous variables. However, microscopically, each unit process may need to be described by a CVDS. This paper is focused on the modeling and simulation of a nuclear facility operation in terms of a DES approach. This study addresses the current status of studies on relevant nuclear facility operation modeling and simulation and describes how to apply discrete event system modeling and simulation methodology to a relevant nuclear facility such as a pyro-headend process facility.

2. BRIEF REVIEW OF DISCRETE EVENT SYSTEMS

A textbook treatment of DESs can be found in Cassandras and Lafortune [2]. DESs have the following properties: the state space is a discrete set, and the state transition mechanism is event driven. In discrete-state systems, the state variables are elements of a discrete set. In continuous-state systems, the state variables are generally any real (or complex) values. In event-driven systems, it is only the occurrence of asynchronously generated discrete events that forces instantaneous state transitions. In between event occurrences the state remains unaffected. In time-driven systems, the state continuously changes as the time changes. For an easy understanding of DESs, they may be compared with CVDSs that have two key properties: state variables are continuous and a state transition mechanism is time-driven. An apparent comparison of the sample paths for a CVDS and a DES is shown in Fig. 1. For the CVDS, the state space X is a set of real numbers R, and $x(t)$ can take any value from this set. The function $x(t)$ is the solution of a differential equation of the general form $\dot{x}(t)=f(x(t),u(t),t)$, where $u(t)$ is the input. For the DES, the state space is a certain discrete set $X=\{s_1,s_2,s_3,s_4,s_5\}$. The sample path can only jump from one state to another when an event occurs. No mechanism is provided to specify how events might interact over time or how their occurrence time might be determined. There are various examples of DESs drawn from the real world and common engineering experience. Representative examples are queuing systems, computer systems, communication systems, manufacturing systems, traffic systems, etc. In order to study the logical behavior of a DES, formal means such as the differential or difference equations in a CVDS are necessary. Such modeling formalisms for a DES include automata and Petri nets. However, the automata and Petri net theories will not be dealt with in this paper. The focus of this study is on the application of a DES scheme using commercial software.

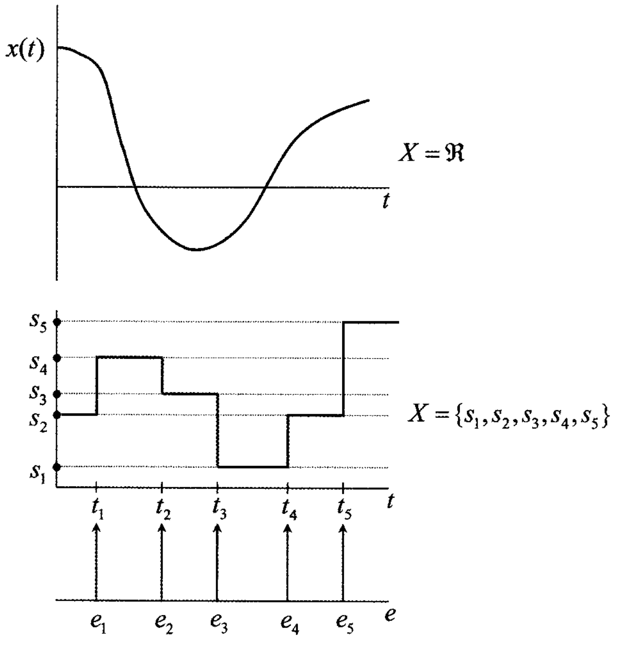


Fig. 1. Comparison of the Sample Paths for a Continuous-Variable Dynamic System (CVDS) and a Discrete Event System (DES)

3. CURRENT STATUS

There are a few examples of studies addressing operational modeling and simulation in relevant nuclear facilities. Idaho National Laboratory (INL) has been performing various operational modeling and simulation studies based on discrete event system modeling and simulation of its own nuclear facilities, including its Experimental Breeder Reactor-II (EBR-II), Hot Fuel Examination Facility (HFEF) and Fuel Conditioning Facility (FCF). Through DES modeling and simulation, INL analyzed its operational scheduling, annual throughput, process equipment capacity and bottleneck process. As a result, the study suggested efficiency improvements and enhanced the facility operation strategy [3-7]. This study claimed that the annual throughput in the FCF could be enhanced by 8 times, from 0.6 MTHM to 5 MTHM. Sandia National Laboratory (SNL) has developed and used a number of simulation models to represent the processing, transportation, and disposal of radioactive waste [8]. In their study, SNL developed a supply chain model for the cradle-to-grave management of radioactive waste and used this model to assist the Department of Energy (DOE) in developing a cost effective, regulatory compliant and efficient approach to the disposal of radioactive waste from 25 sites across the country over the next 35 years. The simulation model was developed on the basis of a DES simulation scheme. Argonne National Laboratory (ANL) also designed a commercial spent fuel reprocessing plant using a DES modeling and simulation scheme.

4. DES APPLICATION TO A NUCLEAR FACILITY

In this section, the application of DES modeling and simulation to a nuclear facility will be explained. Except for the reaction dynamics of a unit process in a nuclear facility, process operations including transfers of material between processes can be shown as a DES. For example, if process 'A' and 'B' are sequential, the end of the 'A' process triggers the 'B' process; that is, the start and end of a given process are events that can cause other processes to transit from one state to another. The following example is a headend process, for which detailed modeling and simulation procedures will be presented.

4.1 Introduction to a Headend Process

The headend process is the first part of the pyroprocess. The headend process starts with a cask connection to the headend facility and ends with separated fuel pin segments. This process does not include any chemical reaction process but only mechanical processes such as withdrawal, cutting, and transportation. The entire process flow can be modeled as a discrete event system; therefore, the headend process is an appropriate example to use for understanding the DES modeling and simulation procedures. Figure 2 presents the flow of the headend process. The top solid line loop accommodates the sequential processes of assembly transfer, inspection and storage, and all the processes are performed at assembly levels. The bottom solid line loop houses processes occurring at the assembly and fuel pin levels. The dotted line loop embedded in the bottom solid line loop collects a fuel pin level process. This loop is repeated for each fuel pin in an assembly. Principal operations in a headend process are as follows:

0. Connection: the connection of a cask used for the transportation of assemblies with an interface system door in a headend process facility.
1. Assembly transfer: the transfer of assemblies to the next processing equipment.
2. Inspection: the inspection of assemblies for any anomalies.
3. Storing: the storage of the transported assemblies in a temporary storage facility.
4. Transportation: the transport of stored assemblies to what it is called the down-ender.
5. Up/down transposition: the laying down of the assembly using the down-ender equipment.
6. Withdrawal of a fuel pin: the withdrawal of a fuel pin from a horizontally fixed assembly in the down-ender equipment.
7. Cutting of a fuel pin: the cutting of the withdrawn fuel pin into a desired segment.

8a. Transportation of a fuel pin vessel: the transport of a fully charged vessel to a designated position and its replacement with a new vessel.

8b. Transportation of assembly waste: the transport of an assembly structure without fuel pins to a designated position.

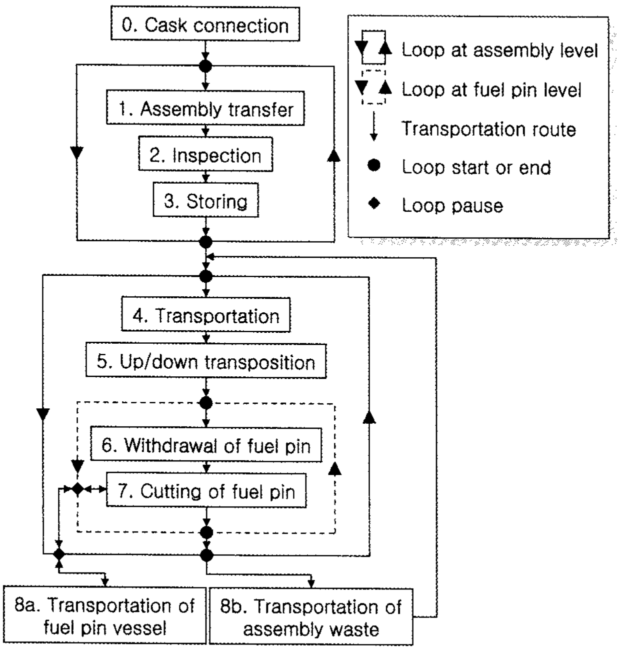


Fig. 2. Headend Process Flow

4.2 DES Modeling Environment

There are many commercial software packages for modeling a DES like the headend process. In this paper, Stateflow [9] and SimEvents [10], developed by MathWorks Inc, were used. Stateflow is the Simulink [11] toolbox for logic-driven system modeling and simulation, and it is based on the Statechart modeling technique. The Statechart modeling technique is actually a state machine with several useful enhancements suggested by David Harel [12] that has been adopted by a worldwide community and is now included as part of the standard Unified Modeling Language (UML). SimEvents is a DES modeling and simulation toolbox in a Simulink environment for CVDS modeling. SimEvents has various blocks relevant to DES that enable the easy undertaking of DES modeling tasks through a drag-and-drop procedure. However, SimEvents has difficulty processing various state transitions, which can be alleviated by complementing SimEvents with Stateflow. In Simulink, communication across blocks is based on signals. In SimEvents, it is based on both signals and entities. The "entity" concept is motivated from the view of a DES as an environment consisting of "users" and "resources": users request resources in order to perform various tasks, occupy these resources for a certain amount of time, and then relinquish them so that other users may access them [13]. For example, in a manufacturing system, users are parts and resources are machines in a factory. SimEvents consists of a number of libraries containing blocks with different system functionalities. The main libraries are as follows:

1. Generators: blocks that generate entities, or function calls (i.e., events that call Simulink blocks), or random variates.
2. Queues: blocks where entities can be temporarily stored while waiting to access a resource.
3. Servers: blocks that model various types of resources.
4. Routing: blocks that control the movement of entities as they access queues and servers.
5. Gates: blocks that control the flow of entities by enabling/disabling the access of entities to certain blocks.
6. Event Translation: blocks that enable a communication between SimEvents and Simulink by translating events into function calls.
7. Attributes: blocks that assign and modify data to entities. Various control actions are then taken based on the values of these data, allowing the blocks to differentiate between the entities they process.

8. Subsystems: these allow for a combination of blocks to be executed upon an occurrence of specific events (not upon Simulink sample times).
9. Timers and Counters: blocks that measure event occurrence times or a time elapse between events, and blocks that count the occurrences of particular event types. These data are supplied to a standard display or scope blocks in Simulink or specialized scopes designed specifically for SimEvents.

The above event translation blocks are necessary for a hybrid system modeling. One of the problems in accomplishing a DES or a hybrid system modeling and simulation is accommodating an event concurrency [14]; that is, allowing the processing of two or more events that arise concurrently. The order in which these events are executed is controlled by means of a priority scheme that is part of the underlying DES design. Such an event ordering functionality does not exist in a time-driven environment. However, SimEvents in a Simulink environment can provide this functionality for modeling hybrid systems.

4.3 DES Modeling for a Headend Process

Unit operations in a headend process are basically modeled as server blocks in SimEvents. Instances of transport between processes are modeled as gates that have a function of blocking entity flows. To make a decision on opening or closing gates, Stateflow blocks are sometimes required. Simple decisions to control gates can be done via SimEvents blocks, but complex logical decisions are difficult to implement via only SimEvents blocks. Since Stateflow provides the language elements required to describe complex logic in an understandable form, it was used for the gate control of the headend process model in this paper. The overall flow of a headend process is shown in Fig. 2. Implementations of a loop control in Fig. 2 can be done in various ways. A use of two latch blocks, which is a typical way to implement a loop control, is shown in Fig. 3. Two latch blocks control a gate prior to a withdrawal server in a way that the gate is closed while a withdrawal or cutting operation is being performed, and the gate is open while neither a withdrawal nor a cutting operation is being performed. Simulink and

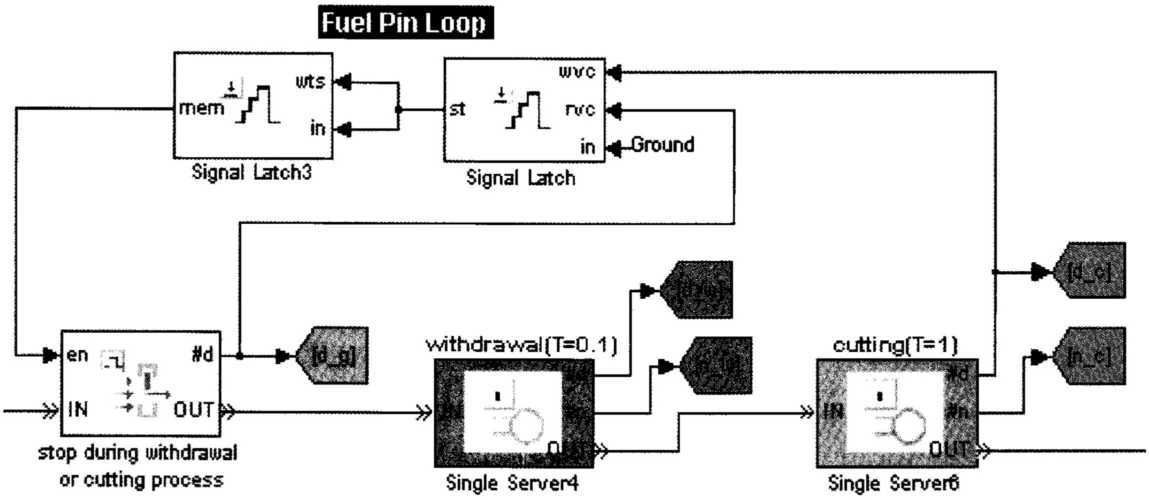


Fig. 3. Two Latches for Fuel Pin Loop Control

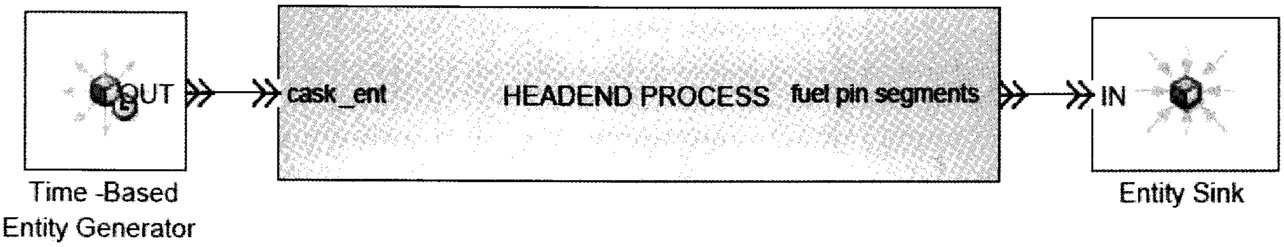


Fig. 4. Top Model of Headend Process

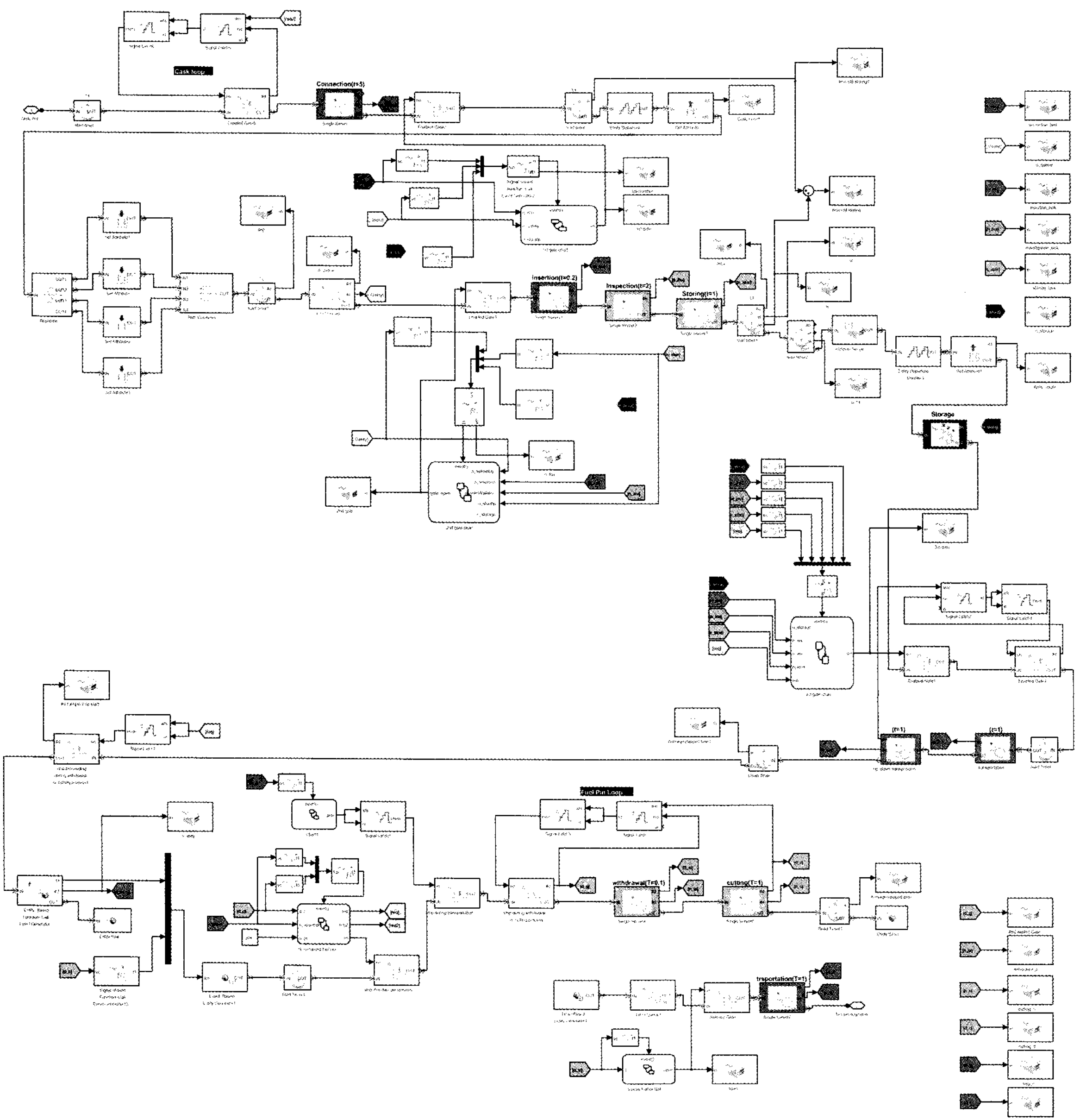


Fig. 5. A Child Model of the Top Model

SimEvents provide a hierarchical modeling approach that allows the construction and efficient maintenance of highly complex models (consisting of thousands of objects) without loss of the ability to explain the models to a non-technical audience. Figure 4 is the top model for a headend process and Fig. 5 is a child model of the centered block in Fig. 4. Fig. 5 also has several child models using Stateflow that are not shown.

5. SIMULATION

There are a couple of assumptions involved in the simulation of a headend process because a headend process facility does not exist at present. First of all, an elapsed time corresponding to each unit operation should be estimated. Server blocks have a dialog window for inserting basic input requirements such as the time. Additional assumptions are the number of assemblies that can be contained in a cask, the number of storage racks, the cask connection frequency, the number of remote handling tools for assembly transportation, the labor work schedule and the frequency of process equipment failure. Modeling of labor work scheduling was not included in the following model, for simplicity. However, a process equipment failure was assumed. Table 1 shows the elapsed time for each unit operation and Table 2 shows the additional assumptions for the headend process simulation. In Table 2, TTF means time to failure and TTR means time to repair. In addition, it was assumed that an assembly has 500 kg of heavy metal spent fuel.

The simulation was performed under the above assumptions. The average elapsed time from a connection to a storing operation is shown in Fig. 6 on an assembly basis. In Fig. 6, the dotted data represents the event loggings at each time when a storing operation is finished. The x-axis indicates the time history for a year in hours. There are five sets of data, and each set consists of data for four assemblies. The second cask contains the 5th to 8th assemblies, and the corresponding average elapsed time

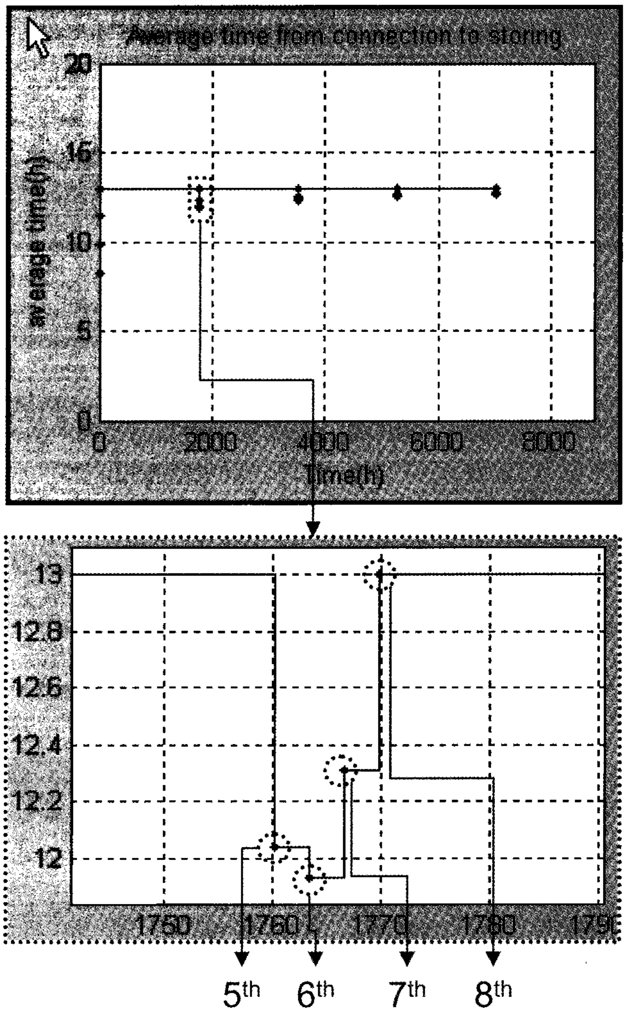


Fig. 6. Average Elapsed Time from a Connection to a Storing Operation

Table 1. Elasped Time for Each Unit Operation

Operation	Elapsed Time (h)	Resource
Connection	5	Cask
Carry	0.2	Assembly
Inspection	2	Assembly
Storing	1	Assembly
Transportation	1	Assembly
Transposition	1	Assembly
Withdrawal	0.1	Fuel Pin
Cutting	1	Fuel Pin
Transportation	1	Fuel Pin Segment
Transportation	1	Assembly Structure

Table 2. Other Assumptions

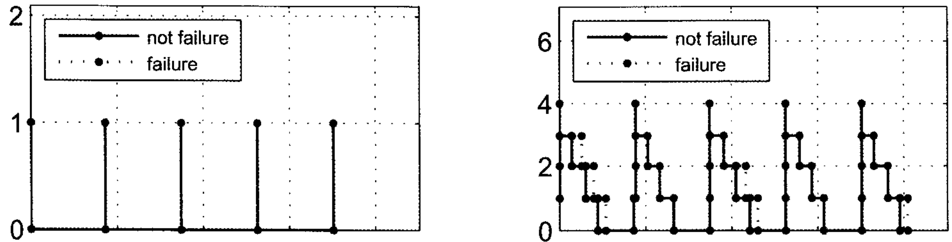
Assumptions	Values
The number of assemblies contained in each cask	4 ea
The number of storage racks	4 ea
Cask connection frequency	5 times/year
The number of assembly transportation devices	1 ea
The number of fuel pins contained in an assembly	264 ea
The vessel capacity for fuel pin segments	100 kg
Labor work schedule	24 hours/day
Process equipment failure at cutting equipment	TTF mean=1500 h, exponential TTR mean=200 h, constant

from a connection to a storing operation for assemblies in that cask can be theoretically estimated by the following equation. Detailed derivation of the following equation will be omitted here.

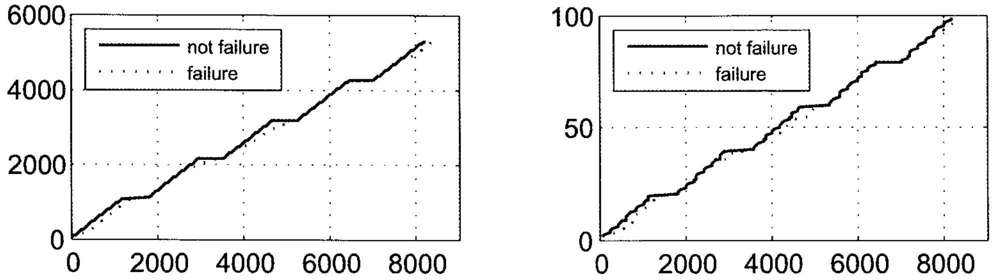
$$T_{connection}+(T_{carry}+T_{inspection}+T_{storing})[(1+2+3+4)(n^{th}\ Quo\ 4)+(n^{th}\ Rem\ 4)(1+(n^{th}\ Rem\ 4))/2]/\,n^{th} \quad (1)$$

In the above equation, '*Quo*' means quotient, for example, '*x Quo y*' means a quotient when x is divided by y. '*Rem*' means a remainder, for example, '*x Rem y*' means a remainder after x is divided by y. 'n^{th}' means the sequence of an assembly. By using Eq. (1), it takes 12.04 hours, 11.93 hours, 12.31 hours and 13 hours, respectively, for the 5th through 8th assemblies to be processed from a connection to a storing operation. The results from Eq. (1) are the same as the results from the simulation shown in Fig. 6. The more complex a process is, the more difficult it is to obtain a closed form solution like Eq. (1).

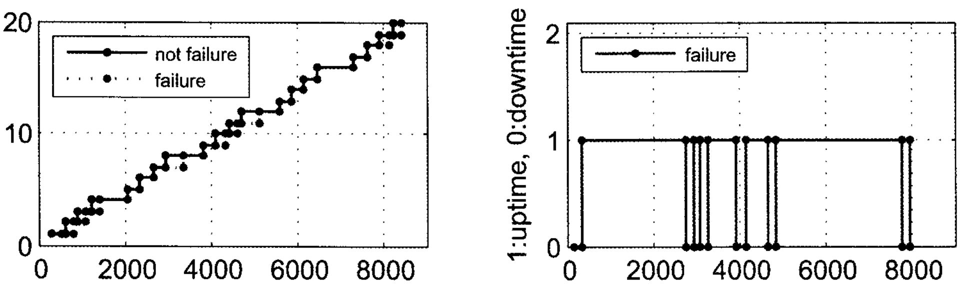
In order to analyze the effect of an equipment failure on the process operation, it was assumed that the cutting equipment failure occurs with an exponential probability distribution of the mean time of 1500 hours TTF and with a uniform distribution of the mean time of 200 hours TTR. Fig. 7 shows a clear comparison of failure mode to non-failure mode. It is assumed that a cask connection event occurs five times a year with a period of 1752 hours to process 10 tons for a year, as shown in Fig. 7(a). Figure



(a) cask connection events (b) # of remained assy



(c) # of fuel pins cut (d) # of rodcut vessel transported



(e) # of assy waste transfer (f) cutting equipment status

Fig. 7. Various Simulation Results under Non-failure and Failure Mode

7(f) presents the failure status of the cutting equipment, where 1 indicates uptime and 0 indicates downtime. In a year, failures occurred 6 times and the associated headend process operations were delayed due to the failures. It can be observed in Fig. 7(b) that the transportation of an assembly from the temporary storage to the down-ender is delayed due to equipment failure. Apparent delayed operations can also be observed in Fig. 7(c) and Fig. 7(d), which show the number of fuel pins cut by the cutting equipment and the number of replacements of the vessel for fuel pin segments, respectively. When the cutting operation is finished, the assembly waste should be removed from the down-ender equipment. Accumulated assembly waste transfer is presented in Fig. 7(e), and it can be seen that the overall operation lags when stochastic cutting equipment failure is assumed.

6. CONCLUSIONS

This paper reviews the current status of KAERI's investigation on the operation modeling and simulation of typical nuclear facilities. This paper also provides insights into how discrete event system modeling and simulation can be applied to a nuclear facility operation analysis. SimEvents and Stateflow from Mathworks Inc. were used for operation modeling and simulation of a headend process to demonstrate a concrete DES modeling and simulation procedure. It was demonstrated that DES simulation could provide an elapsed time between processes without difficulty in obtaining a closed form solution, and could estimate overall delays due to a process equipment failure under various stochastic assumptions.

These kinds of modeling and simulation are very helpful for facility design in terms of determining an optimal process and equipment layout. They are also useful in establishing efficient operation scheduling and in providing a generic understanding of overall plant operation to regulators, stakeholders and operators. Furthermore, modeling and simulation can contribute to estimating the operational and maintenance costs of a facility. In the near future, DES modeling and simulation techniques are expected to contribute to a design study of a pyroprocess facility in the Republic of Korea before its construction, as well as to establishing its operation scheduling, annual throughput and operating cost estimate.

ACKNOWLEDGMENT

The authors are grateful for the support provided by a grant from the Atomic Energy R&D Program of the Ministry of Education, Science and Technology in the Republic of Korea.

REFERENCES

[1] S. W. Park and et al., "Development of advanced spent fuel management process," KAERI/RR-2427/2003, Korea Atomic Energy Research Institute, (2004).

[2] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, **2nd ed.**, p. 769, Springer, (2008).

[3] H. E. Garcia, A. Houshyar and G. R. Imel, "Planning and supervision of reactor defueling using discrete event techniques," *Proc. INRIA/IEEE Conf. Emerging Technologies and Factory Automation*, Paris, France, Oct. 10-13, 1995.

[4] H. E. Garcia and A. Houshyar, "Discrete event simulation of fuel transfer strategies for defueling a nuclear reactor," *Simulation*, **70(2)**, 104-118 (1998).

[5] H. E. Garcia, "Operation analysis and improvement of a spent nuclear fuel handling and treatment facility using discrete event simulation," *Computers & Industrial Engineering*, **38**, 235-249 (2000).

[6] A. Houshyar, "Review of Extend Performance in Modeling a Nuclear Fuel Transfer Activity," *Simulation*, **68(6)**, 403-412, (1997).

[7] A. Houshyar and G. Imel, "A Simulation model of the fuel handling system in a nuclear reactor," *Computers industrial engineering*, **30(1)**, 117-135 (1996).

[8] J. Trone, A. Guerin and A. D. Clay, "Simulation of waste processing, transportation, and disposal operations," *Proc. winter simulation Conf. 2000*, Orlando, USA, Dec. 10-13, 2000.

[9] MathWorks, *Stateflow and Stateflow Coder 6 User's Guide*, **R2007a**, MathWorks Inc., (2007).

[10] MathWorks, *SimEvents User's Guide*, **R2007a**, MathWorks Inc., (2007).

[11] MathWorks, *Simulink User's Guide*, **R2007a**, MathWorks Inc., (2007).

[12] D. Harel, "Statecharts: a visual formalism for complex systems," *Science of Computer Programming*, **8**, 231-274 (1987).

[13] M. I. Clune, P. J. Mosterman and C. G. Cassandras, "Discrete event hybrid system simulation with SimEvents," *Proc. the 8th int. workshop on discrete event systems*, Ann Arbor, USA, July 10-12, 2006.

[14] C. G. Cassandras, M. I. Clune and P. J. Monsterman, "Hybrid system simulation with simevents," *Proc. the 2nd IFAC Conf. on Analysis and Design of Hybrid Systems*, Alghero, Italy, June 7-9, 2006.