Next-Generation System Analysis Model for Studying Alternative Waste Management Systems - 15232

William Nutt, Brian Craig, Kathy Lee Simunich, Evan Vander Zee,
Michelle Kehrer, and James St. Aubin
Argonne National Laboratory (USA)

ABSTRACT

The U.S. Department of Energy's Office of Nuclear Energy (DOE-NE) is conducting planning activities within the Nuclear Fuels Storage and Transportation Planning Project (NFST). The plans involve (1) laying the groundwork for implementing interim storage (including dealing with any associated transportation that is needed) per the Administration's *Strategy for the Management and Disposal of Used Nuclear Fuel and High-Level Radioactive Waste* of January 2013 and (2) developing a foundation for a new nuclear waste management organization. An overarching goal is to develop options for designing an integrated system that will be considered by decision-makers.

Based on the dynamic system analysis and software specification requirements that must be met, Argonne National Laboratory (Argonne) is leading the development of a consolidated waste management system tool known as the Next-Generation System Analysis Model (NGSAM). The development of NGSAM leverages the approaches and methods used previously to develop the Transportation Storage Logistics (TSL) and Total System Model (TSM) system analysis tools, but the new tool takes advantage of advanced computational capabilities.

INTRODUCTION

The management of used nuclear fuel (UNF), (and high-level nuclear waste (HLW) is complex. The management of such wastes requires the use of integrated systems analysis software tools that consider waste generation, on-site storage at various locations, centralized or regional interim storage, transportation and ultimate disposal, .

Although system analysis models have been developed to address the logistics involved in transporting these materials to a geologic repository, these tools cannot simulate the broad range of scenarios associated with both interim storage and ultimate disposition. Tools with such capabilities are needed to support future system architecture evaluations, in order to provide DOE and other stakeholders with information about the various alternatives for managing HLW and UNF.

In 2012 the Blue Ribbon Commission on America's Nuclear Future, in its report to the Secretary of Energy [1], recommended a strategy having eight key elements, including "prompt efforts to develop one or more consolidated storage facilities." Furthermore, in January 2013, the Administration issued its *Strategy for the Management and Disposal of Used Nuclear Fuel and High-Level Radioactive Waste* [2]. As stated,

> "This Strategy includes a phased, adaptive, and consent-based approach to siting and implementing a comprehensive management and disposal system. At its core, this strategy endorses a waste management system containing a pilot interim storage facility, a larger, full-scale interim storage facility; and a geologic repository in a timeframe that demonstrates the Federal commitment to addressing the nuclear waste issue, builds capability to implement a program to meet that commitment, and prioritizes the acceptance of fuel from shutdown reactors. A consent-based siting (CBS) process could result in more than one storage facility and/or repository, depending on the outcome of discussions with host communities; the Nuclear Waste Policy Act of 1982 (NWPA) envisaged the need for multiple repositories as a matter of equity between regions of the country. As a starting place, this Strategy is focused on just one of each facility."

Thus, system analysis tools must be able to simulate a broad range of scenarios that include both interim storage and ultimate disposition. Modern computational tools are needed to perform integrated system analyses of potential future system architectures that consider transportation, storage, and disposal. The NFST is developing the NGSAM to be more readily capable, sustainable, and maintainable in the future than are the existing legacy system analysis tools.

DESCRIPTION

The legacy tools being used to study the transportation of UNF include the TSL model and the TSM. NGSAM will leverage the methods and approaches that these models have taken, but the architecture will be based on a modern computational platform (and thus be more maintainable) that focuses on flexibility. This flexibility will allow subject matter expert (SMEs) having various levels of expertise to run alternative strategies and evaluate each one with respect to various performance criteria (e.g., cost, schedule). Table 1outlines a comparison of the TSL and TSM legacy tools with the NGSAM tool. The main design feature of the new tool is that it gives an SME who is not necessarily a programmer the ability to expand on the scenario logic. Another design feature is that it has access to a unified database that can be used to initialize the model. Legacy tools have local databases that are edited by the users and can therefore make an analysis across runs difficult to quantify.

DISCUSSION

NGSAM is poised to be able to address the current problems that could affect the future of UNF in the United States. One main difference between NGSAM and the currently available tools is the ability of the analysts to adjust this model to adapt to address future developments. The current tools require a software developer to make changes to the underlying code in order to handle many new parameters. Because NGSAM is based on a flexible agent-based simulation, the model will be able to handle almost any future developments related to the logistics and disposal of UNF. Another improvement NGSAM has over the current toolset is its resolution of the simulation. NGSAM uses a discrete event simulator that allows every action to be recorded and made available to the analyst after the run. Previous tools are being executed over time slices, distributing resources across either a 1-year or an 8-hour time slice.

TABLE I. Comparison of Legacy Tools and NGSAM

Feature	Legacy Tools (TSL and TSM)	NGSAM
Platform	Windows only	Windows, Mac OS, Linux
Web-based access	None	Yes — planned for Phases 2 and 3
Platform	Visual Basic 6 and licensed commercial software	Agent-based simulation environment and Java
Input database	Local — maintained by individual users	NFST unified database, controlled and maintained by NFST
Output/Results	Post-process	Internal and post-process
Users	Experts	All levels
Expandable	By programmers	By users

Other features, in addition to those involving the functionality of the existing tools, are scheduled to be added to NGSAM. The model has the capability for detailed cost modeling. By assigning individual costs to each process, the model will be able to generate a detailed cost report, and this will, in turn, feed decision support frameworks and allow for better decision making. Along with costs, the model can assign worker exposure levels to each task. By accumulating the exposures of a model run, analysts will be able to quantify the safety of a scenario based on ALARA (as low as reasonably achievable) radiation safety requirements.

Another key benefit of NGSAM is that it will be integrated with the UNF Storage, Transportation & Disposal Analysis Resource and Data System (ST&DARDS) database (also referred to as “Unified Database”) that is being developed and maintained by Oak Ridge National Laboratory. According to [3], “UNF-ST&DARDS provides a unified domestic UNF system database and key analysis capabilities to support numerous DOE fuel cycle–related objectives, as well as providing the foundation for tracking UNF from reactor power production through ultimate disposition.” By using data based on the UNF ST&ARDS database, analysts will be have access to the most current and verified UNF information available.

Process Analysis Tool: Description, Background, and Capabilities

The NGSAM tool is an agent-based, discrete-event model written in the Java programming language. Agent-based models, or ABMs, are a class of simulations used to study the complex interactions of individuals and/or organizations by defining simple “rules” that they follow. Emergent behavior, or large-scale patterns, may be modeled within these types of simulations without having to explicitly code every interaction. Since the overall purpose of the NGSAM tool is to be a logistics model, we combine discrete event simulation with agents (e.g., casks, canisters, railcars) that are following plans and executing tasks. The underlying NGSAM software development is built from the Process Analysis Tool (PAT), an application developed by Argonne National Laboratory that is designed to allow users to do micro-level planning and analysis without having to write code (i.e., “hard-code” the processing).

The underlying framework for the PAT is the Recursive Porous Agent Simulation Toolkit (Repast) [4]. Repast (renamed Repast Simphony in its latest version) is a widely used free and open-source, cross-platform, agent-based modeling and discrete-event simulation toolkit created and maintained by the University of Chicago and Argonne National Laboratory. Modeling complex interactions through software agents in a discrete-event simulation allows modelers to build components iteratively.

Although the Repast Simphony toolkit is a great platform for modelers, it does require the modelers to know the Java programming language to construct their models. Many application subject matter experts (SMEs) are not Java programmers, but are able to describe their domain within the construct of an agent-based paradigm. With this need in mind, PAT was developed by Argonne National Laboratory on top of Repast Simphony technology.

Development has continued over the past decade, with refinements made as the tool is applied to various domains. PAT has been used in modeling public health, emergency response, military medical readiness response operations, emergency room studies, and National Guard response to natural disasters. PAT was designed to enable users to conduct micro-level and macro-level planning and analysis through a high-level graphical user interface (GUI) intended for SMEs (not programmers) to create, execute, and analyze their scenarios.

Because of the generic nature of PAT, scenarios can be tailored by the analyst to emulate scenario- and site-specific activities at installations for any domain (e.g., for NGSAM, at a reactor, interim storage facility [ISF], or mined geological repository [MGR]). The SME also defines the UNF transportation logistics and procedures to be performed at the locations through the GUI. The PAT tool will be the framework used to define the agents, resources, activities, and events to model the various scenarios to be studied. By defining the tasks that agents need to complete at a location, as well as the duration of the tasks and the specific resources required to accomplish each task (e.g., cranes, personnel), simulation results include graphs showing resource usage and contention for resources that signify bottlenecks. The advantage to using the generic PAT modeling tool is that the analyst can easily add new types of resources, change or add activities at locations, or add new installations — thereby assembling scenarios without having to program in Java.

PAT concepts and definitions follow here (with NGSAM-specific modeling concepts provided in parentheses):

- **Scenario –** A scenario consists of one or more installations. The installations within a scenario contain the agents, resources, plans, tasks, events, and generators for each installation. (A scenario maps to the different cases that will be evaluated for NGSAM.)

- **Installation –** An installation represents a location where agents follow plans and complete tasks (e.g., reactor sites, ISF, MGR, repackaging location, railcar maintenance facility).

- **Node –** A synonym for location. The transportation logistics within a network may be defined as movements from node to node. (Transportation in NGSAM is not explicitly modeled in Prototype 1, but in Prototype 2, the Transportation Operations Model will be accessed for scheduling and costs.)

- **Resource –** Resources are anything needed to complete a particular task (e.g., people, equipment, supplies). (Many resources are being modeled in NGSAM: reactor pool capacity, empty casks, railcar availability, etc.).

- **Agent –** Agents are the simulation objects that complete the tasks within the scenario. Attributes are defined for types of agents, and these attributes are checked in the simulation to help the user decide which task to do next. They can be altered as the result of doing a task. Agents may also be resources for tasks when other agents are completing a task. (NGSAM models the assemblies, casks, railcars, trucks, barges, and transportation overpacks as agents.) The agents are the entities that make the decisions and move through the plans and tasks; definitions follow).

- **Task –** Tasks are the basic building blocks of the simulation. Tasks can be done by an individual (an individual task [IT] executed by a single agent) or by a group [GT] (executed by all members of a group). A task can be executed because it is triggered by an event or as part of the decision logic in an agent's plan. (Refer to the section discussing the NGSAM scenario: All tasks are user-defined and, as such, create the logic of the scenario run).

- **Unified Task –** These special types of tasks are modeled as an individual task [IT], however, the individual is a unified group. A unified group is created from group task [GT] when the user sets the "unified" flag to true signifying that the agent moves forward to decide on the next task to do as a unified group. All members of the group remain unified until the user sets the flag to false in a task that the group executes. (In NGSAM, examples would be assemblies in a cask, once packed, move as a cask group, then when the cask is loaded onto a railcar, it moves now as a railcar group. Disbanding the railcar group, when the cask is unloaded, and disbanding the cask group if the assemblies are removed from the cask.) All tasks that members do as a group are recorded on their agenda as having executed the task so that when viewing the agent traversal graph in the results, the analyst can graph the flow the agent took through the model.

- **Plan –** Plans define the overall flow through a series of tasks. There may be many plans in an installation. Agents flow through the tasks at an installation by checking the task's attributes against the conditions in the plan to decide which task in the plan should be completed next. (Refer to the section discussing the NGSAM scenario: The plans organize the tasks for the casks, assemblies, etc. to execute.)

- **Event –** Events provide a way for agents to alter their plan of action based on a number of different conditions. These conditions might be resource levels, the state of some other agents, or the state of the agent itself. Events may be defined to execute at a certain time, or they may be triggered conditionally based on a change in the value of an attribute of an agent or resource.

- **Distribution –** Distributions may be used to define the duration of a task, set an attribute of an agent, set a value for a resource within a task, and create (generate) agents within a simulation. The analyst is given a selection of the most common distributions (e.g., normal, triangular) and defines the properties that will shape the curve (e.g., mean, standard deviation). The analyst can also create custom distributions by providing a comma-separated value (csv) file containing the data (e.g., assembly packing time)

- **Generator –** Agents and resources can be generated during the simulation, either at the start or as the result of an agent finishing a task.

The modeler/analyst/SME uses the concepts just described to design the scenario for modeling within the NGSAM tool. The design process for creating simulations generally follows this sequence of steps:

- Define the agents and installations for the domain.

- Define attributes for the agents that are important to the model.

- Define the plans at the location — including decision points called *triggering conditions* — that affect the flow of decisions made in the simulation (i.e., what task an agent will decide to do next).
    - Diagram the initial process flows and use the output to create the tasks and decision points in a plan. Make as many plans as needed to represent all the activities at the installation.
    - Tailor the activity durations for each task and the resources required to complete each task.
    - Execute the tasks selected and analyze alternative courses of action.

- Once the initial model is tested,
    - Change the timing distributions of processes/tasks;
    - Change the type/number of available resources;
    - Add/delete/modify agents required to perform the processes;
    - Run the alternative/contingency plans;
    - Analyze the bottlenecks/resource utilization; and/or
    - Work with experts to develop new scenarios.

The NGSAM user interface has a canvas area where the user uses drag and drop tools to create the tasks for a plan (see Figure 1). This helps the user to sketch out the flow of logic that the agents will follow. Selecting the nodes in the graphs brings up task-specific editors in which the user inputs the requirements for any resources and the duration of the task and indicates the attribute values that change after the task is executed. Another option is for the tasks to designate resources or agents to be generated or indicate NGSAM-specific methods to use to execute thermodynamic calculations.

The next sections discuss the specific scenario created for the NGSAM year one prototype.

Scenario Discussion

The scenario prototype includes one managed geologic repository (MGR), one interim storage facility (ISF), and two power-generating operational reactor sites. The two reactor sites in the scenario generate used nuclear fuel and include simulation processing for transporting that fuel away from the reactor site. It is easy to add reactor sites to the model by using a reactor template provided with the NGSAM prototype, but the number of reactor sites has been kept small during initial model development because scenarios run faster and are easier to test when there are only a few reactor sites.

Each reactor site includes a model of the management of the reactor pool. The inventory of fuel rods in the pool is monitored, and when the inventory reaches a high enough level that there is no longer enough space reserved for a full reactor core, the model initiates processes that will extract rods from the pool, place the rods in a canister, and place the canister in dry storage. Figure 1 shows many of the tasks involved in monitoring the pool, such as detecting when more space has been made available in the pool because either bare fuel has been transported off site or fuel has been placed in dry storage at the reactor site.

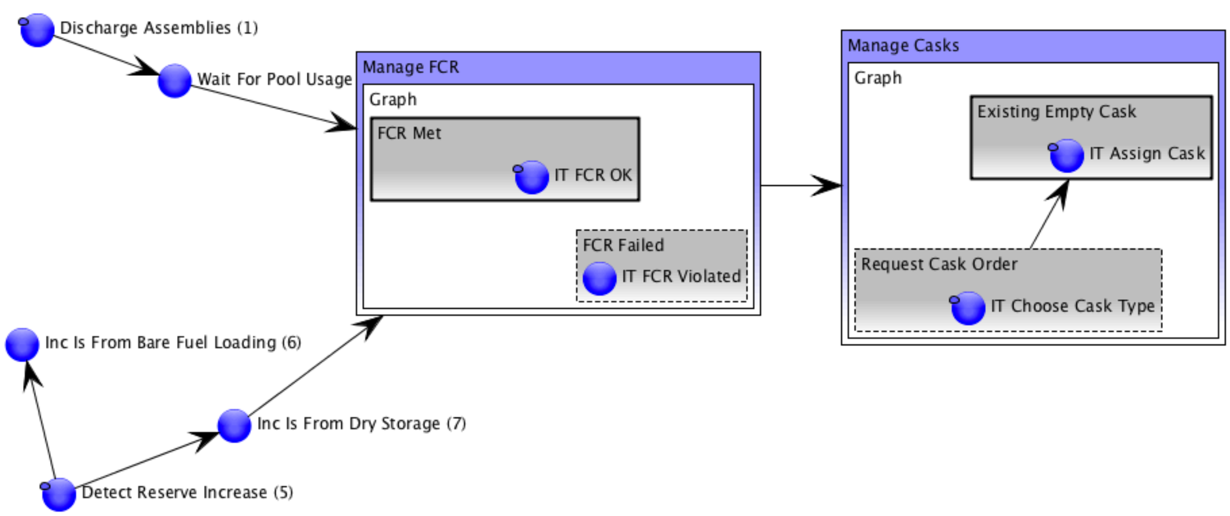

Figure 1. Reactor Pool Inventory Monitoring Plan Managing the Fuel Core Reserve (FCR)

In the prototype scenario, transportation is modeled as railcars that can move around independently. The transportation is mostly scripted in the prototype. It is controlled by an input file that specifies required transportation itineraries from reactor to ISF. Individual railcars are assigned to these itineraries as the itineraries are introduced into the model. In the future, the model itself will produce these itineraries as it makes higher-level allocation and transportation decisions. A future version of the model is planned that will include the modeling of trains with their appropriate railcar configuration and also some representation of transportation on modes other than rail.

The reactor site model in the prototype supports several different methods of transporting used nuclear fuel on a railcar. One transportation configuration supports taking canisters out of dry storage at the reactor, placing one or more canisters into a transportation overpack, and loading the overpack onto the railcar. Another transportation configuration supported by the model involves taking bare fuel out of the pool and placing it into a reusable transportation cask. Once the reusable cask is loaded with bare fuel, the cask is loaded onto the empty railcar for transportation away from the reactor site. Both of these types of transportation have been implemented to allow different methods to be used for choosing which UNF will be transported away from the reactor site first (e.g., first in, first out; last in, first out; coolest first)[1]. Those methods can also be applied when the model is choosing which UNF will be taken out of the pool to be placed in dry storage.

The prototype model of the ISF interacts with arriving loaded railcars to unload the UNF they are carrying and place it into storage. Similarly, when empty railcars arrive at the ISF, UNF is taken out of storage and loaded onto the railcars. Figure 2 shows the part of this process that involves placing one or more canisters into a transportation overpack after those canisters have been removed from dry storage at the ISF.

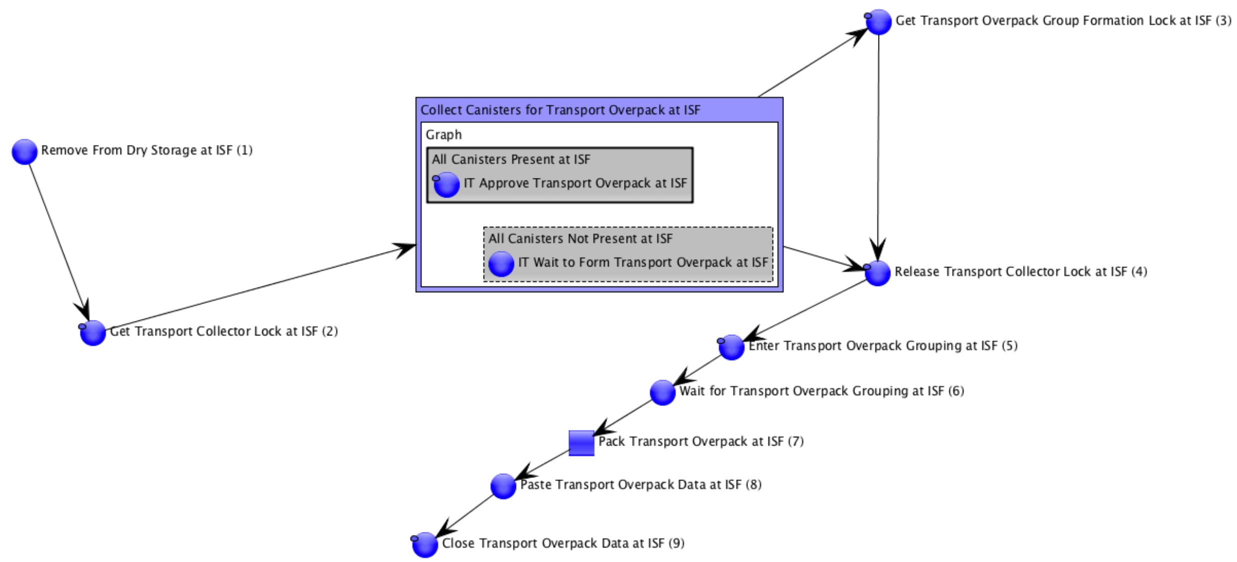

Figure 2. ISF Railcar Loading Plan

The MGR model, like the ISF model, is a fairly rough sketch that will be expanded with more details in the future. The current model supports the arrival of loaded railcars, the offloading of those railcars, the packing of a UNF canister into a disposal overpack, the emplacement of a

[1] The Standard Contract for the Disposal of Spent Nuclear Fuel and/or High-Level Radioactive Waste (10 CFR Part 961) that DOE has in place with nuclear utilities orders acceptance and transportation priority based on oldest fuel first, i.e., based on date of discharge from the reactor. However, the Secretary of Energy has discretion under the Standard Contract to decide whether to give priority acceptance to used nuclear fuel at shutdown sites (10 CFR 961.11, Article VI.B.1.(b)).

disposal overpack into the geologic repository (Figure 3), and the temporary storage of UNF canisters that do not meet certificate of compliance requirements for emplacement into the geologic repository. Since certificates of compliance are not really part of the model yet, checking whether a canister satisfies the certificate of compliance is implemented as a random draw in the prototype model. “Swapping in” a proper check of a certificate of compliance should be quite easy, however, once the certificate of compliance data and radioactive heat dissipation models are implemented in the simulation.

Most aspects of the current prototype focus on dealing with the UNF from commercial reactors. We intend the model to eventually be expanded to include the storage and transportation of HLW too. The processes for HLW are quite similar to those for UNF from commercial reactors, but they are usually handled by different resources with different storage capacities.

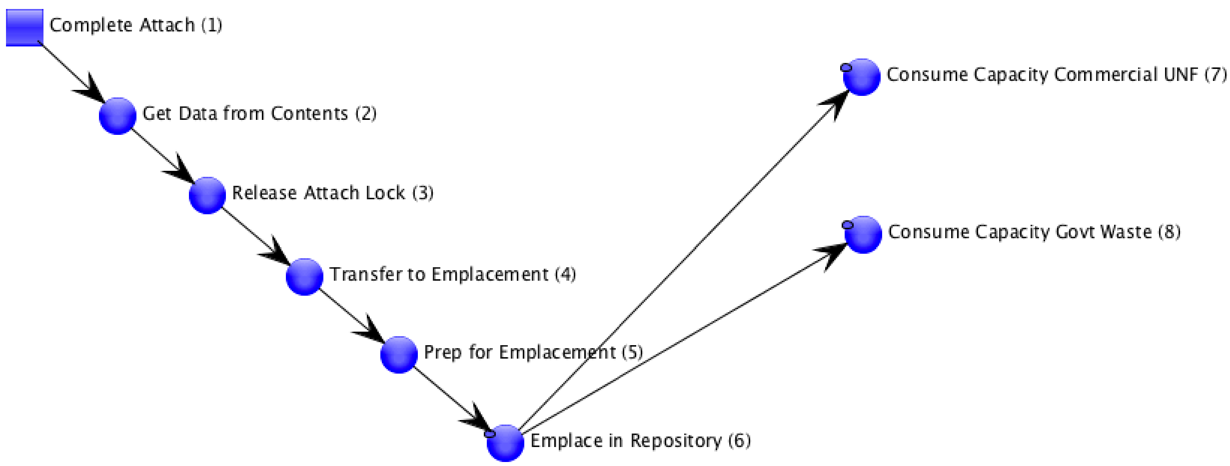

Figure 3. MGR Emplacement Plan

Results

The NGSAM tool provides many types of results that are viewable at the end of a run, including: plan coloring, agent traversal graphs, and resource graphs.

After the simulation is executed, the plan view updates to include labels to indicate movement through tasks and colors to indicate the relative number of agents that waited to perform each task as compared to other tasks. For tasks colored green, there was no waiting for resources; task colors white to red indicate increasing wait times. Figure 4 shows an example of plan coloring. In this case, most tasks were completed without a wait, but the IT Assign Cask task involved many assemblies that had to wait for an available empty cask.

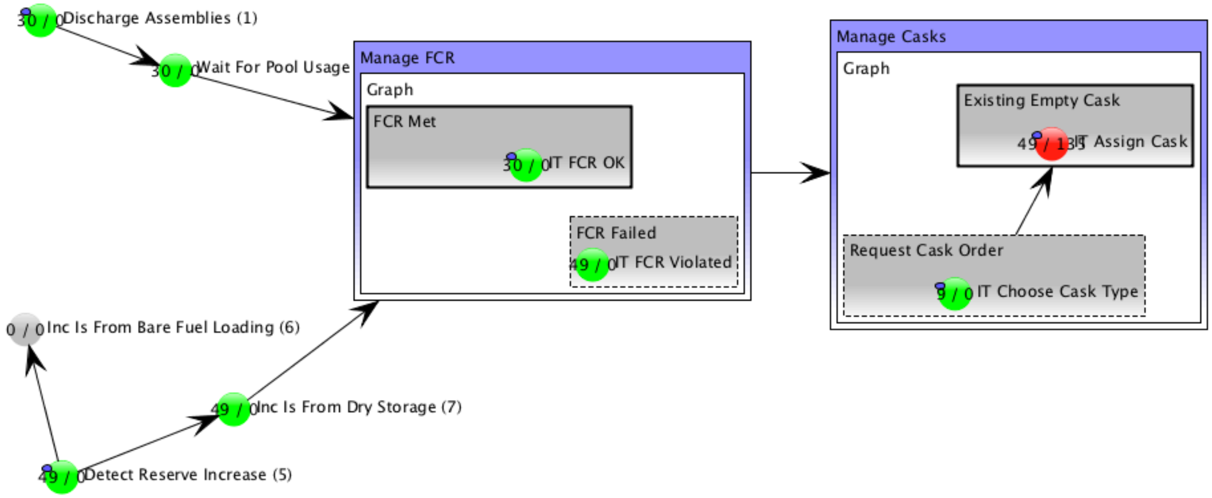

Figure 4. Reactor Pool Inventory Monitoring Plan Results

Agent traversal graphs provide a visualization of what actions an individual agent performed during the simulation. The graphs use colors similar to those of the plan just described. In these graphs, a task that shows a wait indicates the number of minutes the agent spent waiting to perform the task. Squares indicate tasks performed as part of a group, and rounded squares indicate the task that the agent did as part of a unified group. Figure 5 shows a portion of the agent traversal graph for an assembly. The assembly waited 13 minutes to perform the individual task called “IT Pack in Cask,” then performed the group task (GT) called “GT Complete Loading Cask.” This GT creates the unified group for the loaded cask. In this case, the unified group consists of the cask and the assemblies packed into it. This unified group goes on to do tasks together as a group until it is disbanded later (usually during repackaging or at final disposal). Because the assembly is loaded into the cask, it now tracks all of the tasks performed as part of the loaded cask on its own agenda. These tasks are shown as rounded rectangles to indicate that the assembly did not directly perform the task but performed it as part of the loaded cask.

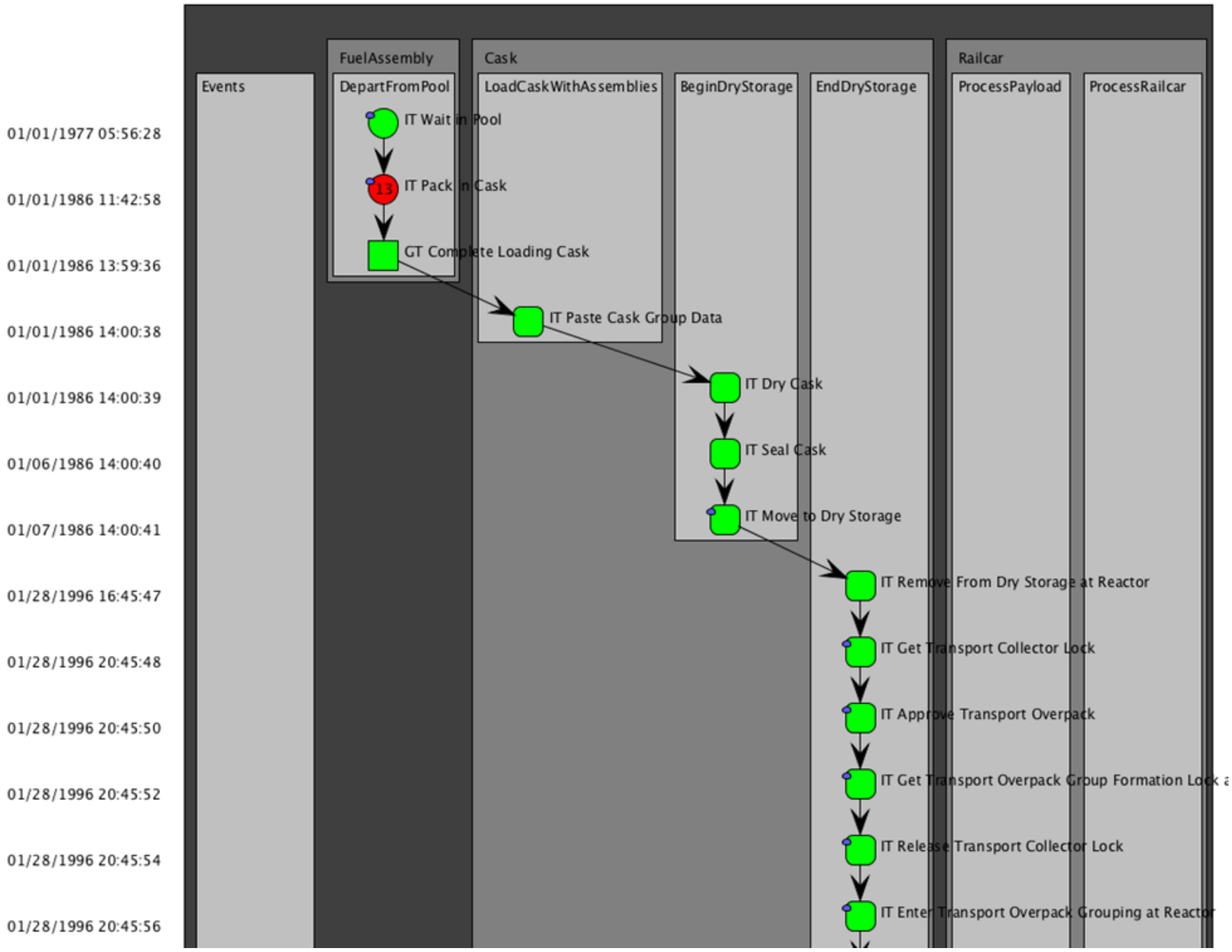

Figure 5. Agent Traversal Graph for an Assembly

The resource graphs show the amount of resource on hand and the number of resources in use over time. Figure 6 shows the number of assemblies in the pool at the reactor, where the pool capacity is modeled as a resource. Initially, the number of assemblies continues to grow, then as the capacity at the pool is reached, the assemblies begin to be removed from the pool and placed in casks for transport.

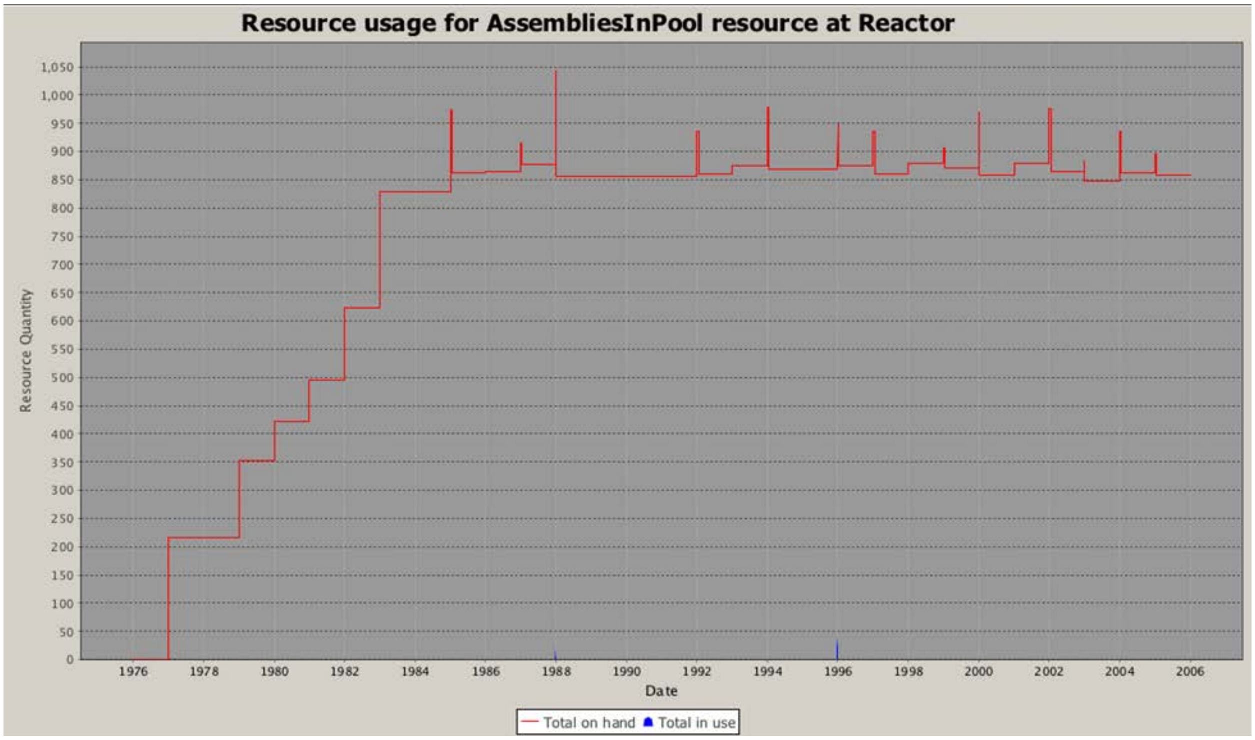

Figure 6. Assemblies in Pool Resource Graph

CONCLUSIONS

The purpose of the multiyear NGSAM project is to design and develop a consolidated UNF system analysis tool for DOE-NE. Argonne is using an iterative development approach to simulate a broad range of scenarios, including the transportation of UNF from reactors to interim storage to ultimate disposition. NGSAM objectives are to:

- Provide quantitative information with respect to a broad range of alternatives and considerations;
- Develop an integrated approach for evaluating storage, transportation, and disposal options — with emphasis on flexibility;
- Evaluate the impacts of storage options on handling and disposal;
- Identify alternative strategies and evaluate them with respect to various performance criteria (e.g., cost, flexibility, schedule); and
- Consider various factors (including repository emplacement capacity, thermal and dosage constraints, repackaging needs, storage and transportation alternatives) and their impacts on utility operations.

NGSAM will leverage methods and approaches from legacy tools, but its architecture will be based on a modern computational platform (and so will thus be more maintainable) that focuses on flexibility. This flexibility will allow SMEs with various levels of expertise run alternative strategies and evaluate each one with respect to various performance criteria (e.g., cost, schedule). The scope, plan, and architecture for the NGSAM will be managed by using a “spiral” methodology or phased approach to the development and implementation of both the technical infrastructure and database application software. The NGSAM system application framework will be built in layers by using the PAT framework augmented with PAT enhancements and Java-based software. NGSAM will be a consolidated tool that will let NFST users perform micro-level and/or macro-level system analyses of alternative system architectures.

REFERENCES

1. Blue Ribbon Commission on America's Nuclear Future, Report to the Secretary of Energy, January 2012.
2. DOE (U.S. Department of Energy), *Strategy for the Management and Disposal of Used Nuclear Fuel and High-Level Radioactive Waste,* Jan. 2013.
3. J.M. Scaglione, J.L. Peterson, K. Banerjee, K.R. Robb, R.A. LeFebvre, Integrated Data And Analysis System For Commercial Used Nuclear Fuel Safety Assessments, Proceedings of the WM2014 Conference, March 2-6, 2014, Phoenix AZ (paper no. 14657)
4. North, M.J., N.T. Collier, J. Ozik, E. Tatara, M. Altaweel, C.M. Macal, M. Bragen, and P. Sydelko, "Complex Adaptive Systems Modeling with Repast Simphony," *Complex Adaptive Systems Modeling* 1, Springer, Heidelberg, FRG, 2013, http://www.casmodeling.com/content/1/1/3.

ACKNOWLEDGMENTS

This work is supported by DOE under Contract No. DE-AC02-06CH11357. The authors wish to acknowledge the excellent leadership and strong support provided by DOE-NE for this project. This work was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process or service by trade name, trademark, manufacturer or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.