Development of a Dynamic Simulation Approach to Mission Risk and Reliability Analysis

Ian Miller and Andrew Burns

GoldSim Technology Group LLC: 22516 SE 64th Place, Suite 110, Issaquah, Washington, 98027,
imiller@goldsim.com and aburns@goldsim.com

Abstract – *This paper describes a NASA-funded project to develop reliability analysis software capable of modeling complex, highly dynamic systems over the duration of a mission, taking into account variation in input parameters and the evolution of the system. It is designed as an extension to GoldSim, a simulation program which is widely used for Performance Assessment in the nuclear arena, most notably at Yucca Mountain.*

To illustrate the GoldSim approach to reliability modeling, two NASA examples that have previously been evaluated using classical PRA models were developed using the simulation approach. Issues surrounding the translation of the classical PRA models to a simulation-based approach are discussed, and areas where the simulation approach provided additional insights into the system's behavior are highlighted.

I. INTRODUCTION

The nuclear industry, with its intense public and regulatory scrutiny, has fostered a number of advances in the fields of risk analysis, quality assurance and quality control. These advances have been adapted for use in a number of other fields. In particular NASA, in its 2002 documents Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners [1] and Fault Tree Handbook with Aerospace Applications [2], has developed its own implementation of the Probabilistic Risk Assessment (PRA) methodology originally used for nuclear power plants. The approach focuses on the analysis of initiating events and subsequent event sequences that could lead to failures, and on enumerating and calculating the probabilities of different outcomes through tree-based analytical procedures.

The PRA approach used for nuclear power plants was developed to do risk analysis of complex systems composed of highly-reliable and frequently redundant components, which are required to have an extremely low risk of a catastrophic failure. As such, the methodology is well-suited to apply to many NASA mission risk assessments.

However, the existing PRA approach is not universally applicable. Apostolakis [3] commented that "*NASA … is still exploring what QRA's can do and how to apply them realistically to its systems, some of which are very different from nuclear power applications. For example, methodological improvements are needed to handle the dynamic nature of space flights.*".

Also, the existing PRA approach can be complex, and is highly dependent on the skill of the analyst. Labeau et al [4] commented that "*Classical PRA practitioners claim that component, time and process variable dependency can be incorporated in the Event Tree/Fault Tree approach. However, systematically incorporating all these dependencies and representing them using the ET/FT approach requires a tremendous amount of preprocessing effort. Therefore, the quality of the PRA is completely analyst dependent. … It can therefore be seen that any explicit incorporation of dynamics constitutes an improvement in modeling of the scenario and on classical PRA.*".

In fact, in the critical area of nuclear waste management the nuclear industry itself has had to develop an alternative PRA approach that is suitable for a quite different kind of system. Predicting the safety of buried wastes for periods of up to a million years poses unique challenges, some of which have parallels in NASA missions:

- Because of the long time scale involved only very limited testing can be carried out, so the safety analysis of the system has to rely to a large extent on numerical models.
- The systems are very complex and require the integration of analyses from many disciplines.
- There can be substantial uncertainty about how the boundary conditions for the system will evolve over time.
- There can be substantial uncertainty about some of the internal processes that affect the system.
- Much of the uncertainty is epistemic, and as such requires expert judgment to develop the appropriate probability distributions, as opposed to statistical analysis of historical or test datasets.

- The system can be affected by discrete events, but to a large extent the analysis has to consider its dynamic evolution as driven by gradual time-based processes.

In response to these challenges, the nuclear industry has evolved a safety analysis approach that is referred to as Performance Assessment (PA). In essence, PA analyses use stochastic dynamic simulation to develop probability distributions of the system's performance. These simulations use fully-coupled but somewhat simplified ("abstracted") models of the system components, which are typically brought together within a single integration framework code. The integrated simulations explore the range of possible 'trajectories' of the systems as they evolve through time, subject to both gradual and sudden changes.

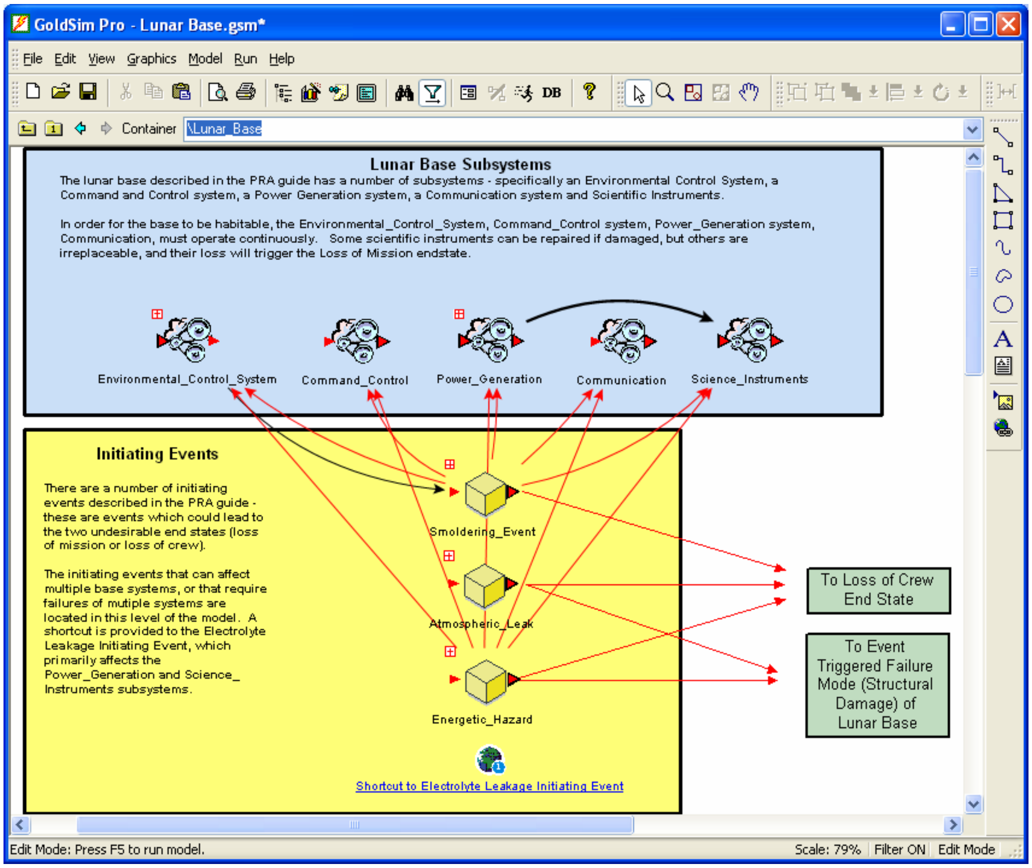
The NASA SBIR project described in this paper had two phases. The first phase included enhancements to the GoldSim software so that the PA approach is more applicable to NASA missions. This involved the development of a GoldSim extension module specifically designed for reliability modeling. The second phase involved constructing PA models of two NASA PRA examples in order to develop basic procedures for dynamic mission modeling.

Note that the two examples that are discussed in this paper are entirely fictitious, and all quantitative data and results are purely for illustrative purposes.

II. BACKGROUND TO GOLDSIM AND THE RELIABILITY MODULE

GoldSim can be thought of as a high-level programming language, where the program is the model. The analyst joins together objects called "elements" using "links" to create the model of the system. Elements, which may represent either physical or logical components of the system, will often have a stochastic component, and the links carry information between the elements.

The modeler creates a representation of the system in its initial state, imbuing the elements with the appropriate properties, behaviors, and relationships. Then, when the simulation is started, the software takes over and evaluates the entire history of the system, saving selected results for subsequent analysis.


GoldSim Pro - Lunar Base.gsm*
File Edit View Graphics Model Run Help
DB
Container \Lunar_Base
Environmental_Control_System
Command_Control
Power_Generation
Communication
Science_Instruments
Smoldering_Event
Atmospheric_Leak
Energetic_Hazard
Shortcut to Electrolyte Leakage Initiating Event
To Loss of Crew End State
To Event Triggered Failure Mode (Structural Damage) of Lunar Base
Edit Mode: Press F5 to run model.
Scale: 79%
Filter ON
Edit Mode

Lunar Base Subsystems

The lunar base described in the PRA guide has a number of subsystems - specifically an Environmental Control System, a Command and Control system, a Power Generation system, a Communication system and Scientific Instruments.

In order for the base to be habitable, the Environmental_Control_System, Command_Control system, Power_Generation system, Communication, must operate continuously. Some scientific instruments can be repaired if damaged, but others are irreplaceable, and their loss will trigger the Loss of Mission endstate.

Initiating Events

There are a number of initiating events described in the PRA guide - these are events which could lead to the two undesirable end states (loss of mission or loss of crew).

The initiating events that can affect multiple base systems, or that require failures of mutiple systems are located in this level of the model. A shortcut is provided to the Electrolyte Leakage Initiating Event, which primarily affects the Power_Generation and Science_Instruments subsystems.

Fig. 1. Screenshot of a GoldSim Reliability Model

II.A Dynamic Monte-Carlo Simulation

GoldSim uses dynamic Monte-Carlo simulation to predict the evolution of the system. In Monte-Carlo simulation, multiple "realizations" of the same system are run for a specified duration with different stochastic input values (produced using a repeatable pseudo-random number generator). Element values are updated, stochastic values are resampled, and random events are triggered as appropriate as simulated time passes. Each realization is considered equally likely, and results from the realizations are combined to provide probabilistic information about the likely performance of the system.

II.B The Reliability Module

The reliability module is an add-on to the standard GoldSim simulation framework, consisting of two new element types: the Function element and the Action element. The Function element is used to model components that perform their function over a period of time (e.g., a battery or an environmental control system), while Action elements are used to model components that perform their duties only when triggered by a specific condition or conditions (e.g., a relay, or an actuator). The primary output of a reliability element is its operating state at any given time during the simulation: whether it is operating or not.

Both reliability element types contain features to accurately represent components of a reliability system in a dynamic Monte-Carlo simulation. These include:

- Requirements/Fault Trees – Reliability elements can be dependent on other peer (same functional level) and child (subcomponent) reliability elements. They can also contain logical conditions referring to any other elements in the model. These trees are evaluated each time the model is updated.
- On/Off Switches – Reliability elements can be turned on and off at specific times or when certain conditions are met.
- Failure Modes – Up to 99 different failure modes can be specified for each reliability element. Each failure mode is given a probability distribution (e.g., exponential, lognormal, Weibull), or a specified probability of failing when a certain event occurs.
- Repair Logic – Automatic repairs can be specified for each individual failure mode using three built in distributions. Multiple failure modes can be repaired using a Preventive Maintenance event, or the entire component and all of its children can be replaced during a Replace event.
- Containment – Each reliability element can either act as a simple element, with its failure distribution specified by failure modes, or as a more complex system which contains models of subcomponents. Frequently, an initial model is constructed using simple reliability elements with failure modes, and in subsequent versions of the model these are enhanced with subsystem models until an appropriate overall degree of modeling realism is achieved.

Virtually all of the inputs to a Reliability element can vary over time, meaning that failure mode parameters, aging rates, and repair times can vary as the simulation progresses through time. This fully-dynamic capability distinguishes the simulation approach from a fault-tree approach, which can at best only view the system as being in one of a number of predefined states. For example, with the simulation approach the failure rate for a component could change dynamically with the system's operating temperature, which was computed dynamically as part of the simulation.

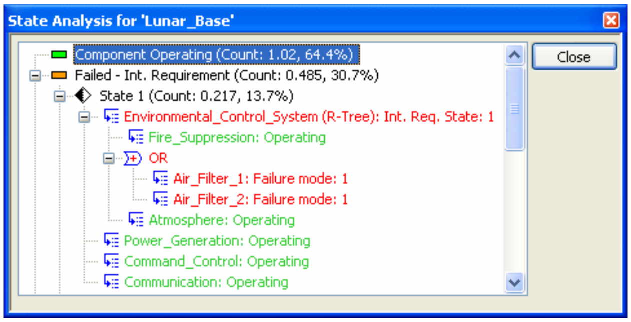
In a standard GoldSim model, the simulation is updated after regular time intervals called timesteps. However, since many reliability events happen quickly, and are unlikely to occur on a timestep, the reliability elements (like other time-based events in GoldSim) have the power to control GoldSim's simulation clock. If an event occurs at a particular time, GoldSim can interrupt the simulation and update the model. For example, consider a failure mode that occurs 12.54 days into the simulation, and is repaired with an exponential distribution with a mean repair time of 1 day. The simulation would be updated at 12.54 days to reflect the failure, and the repair time sampled. If the sampled repair time was 0.72 days, the simulation would subsequently be updated at 13.26 days to reflect the repair.

In this way, GoldSim can be used to model systems with high reliability accurately over long periods of time, without an inordinate level of computational effort.

Each of the reliability elements also contains tools to analyze the performance of the modeled component. These include standard metrics such as operational availability, inherent availability, and reliability over the course of the simulation, and statistics for the distributions of times between failures and times to repair.

This is another distinguishing feature of the simulation approach, as result metrics are provided not just as an expected value or a probability of success/failure, but also as full probability distributions. For example, a measure such as the amount of time for which a mission was able to gather scientific data might be expressed as a 17% likelihood of no data at all, a 15% likelihood of between 0 and 20 days, a 42% likelihood of between 20 and 100 days, and so on.

In addition, GoldSim records all of the unique states that each reliability element experiences during the simulation (i.e., whether the component is operating, if a particular failure mode has occurred, if it is undergoing maintenance, is turned off, or its requirements- or fault-tree shows the component cannot operate). These unique states also record the states of any reliability elements referenced as part of a requirements- or fault-tree. These are stored, and can subsequently be analyzed based on state occurrence frequency or the time spent in each state. If a component's fault-tree prevents it from operating, GoldSim will remember the condition of the fault-tree, and of the fault-trees of any referenced elements that have failed, allowing the user to browse to the root cause of the failure.



```
Component Operating (Count: 1.02, 64.4%)
Failed - Int. Requirement (Count: 0.485, 30.7%)
  State 1 (Count: 0.217, 13.7%)
    Environmental_Control_System (R-Tree): Int. Req. State: 1
      Fire_Suppression: Operating
      OR
        Air_Filter_1: Failure mode: 1
        Air_Filter_2: Failure mode: 1
      Atmosphere: Operating
    Power_Generation: Operating
    Command_Control: Operating
    Communication: Operating
```

Fig. 2. Causal Analysis Result Example.

This ability to explore the causes of failures is another way in which the simulation approach is distinguished from the fault-tree approach. In a sense, the simulation approach 'discovers' failure scenarios, as opposed to the classical PRA approach where the analyst has to define the failure scenarios *ab initio*.

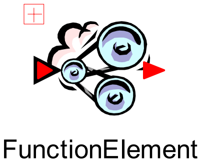
III. TRANSLATING PRA CONCEPTS INTO A DYNAMIC SIMULATION MODEL

As part of the SBIR project, two models of existing NASA case studies were built using GoldSim, to demonstrate the process and to help develop methodologies for building dynamic reliability models. During this process, a number of strategies and techniques were devised to help port classical PRA analyses to the simulation approach.

Conventional PRA analysis approaches address failure scenarios that may arise following the occurrence of initiating events. Typically, during the systems analysis phase of the PRA a list of all credible initiating events is generated, and event sequence diagrams (ESDs) are used to catalog the potential consequences of each initiating event.

III.A Modeling Initiating Events in GoldSim

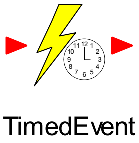
In the GoldSim approach, the same type of systems analysis is required in order to identify system components, their behavior, and initiating events. The identified initiating events can be subdivided into two categories: **the failure of one or more system components** and **the occurrence of disruptive events**, as discussed below.



The potential failure of a system component in GoldSim is modeled by inserting a reliability-module element that represents the component. A component failure may occur in several ways:

- A defined failure mode for the component may occur. Failure modes can be:
  - Purely internal to the component.
  - Internal to the component, but affected by its external environment.
  - Failures triggered by external events. For event-triggered failure modes the user specifies the probability of failure if the initiating event occurs.
- The user can specify 'external requirements' for the component to operate, using either a fault-tree or a requirements-tree that references other entities in the model. If critical requirements are not available, the element will stop operating. For example, a sensor might fail to operate if not provided with electrical power.
- Similarly, for components that have a complex internal structure the user can specify a fault-tree or requirements-tree for its internal requirements, and the element will fail to operate if the requirements become unavailable. For example, an engine might fail to operate if its internal electrical system fails.
- Some GoldSim elements can be designed to carry out a specific action when required, and these elements may have a probability of failing to successfully perform their actions, even though the element is still operational.
- A logic or control system failure could result in an operable component not being turned on.

Disruptive events are modeled in GoldSim by event generator elements. The disruptive events may be random in time or may be triggered by circumstances, and can arise from a variety of mechanisms:



- A time-based random external event might occur, such as a solar flare or a terrorism event.
- A random internal event (other than a component failure) may occur, such as an unplanned human action.
- An internal event may occur due to an unusual combination of circumstances, such as running out of fuel or overheating.

The disruptive event may trigger a failure of one or more system components. Alternatively, it may modify the operating environment of the system, changing the system's behavior and potentially modifying the stresses on the system components.

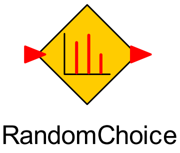
Note that the failure of a system component is itself an event in GoldSim. Thus it is possible for the failure of one component, or the occurrence of a random event, to trigger failures in other components leading to common cause failures.

III.B Event Sequences in GoldSim

In conventional PRA analyses the potential effects of an Initiating Event are represented using an Event Sequence Diagram (ESD) and/or an Event Tree, with the consequences of an initiating event cascading through a series of chance nodes known as 'pivotal events'. The outcome of each pivotal event reflects the probabilistic state of a particular component of the system: does the detector detect the smoke? These tree-based approaches rely on the judgment of analysts who have the experience and imagination to identify all potentially-significant event sequences.

In the simulation model, however, the user has only to define the elements that are directly affected by an event. The effects of an initiating event then arise naturally out of the model's logic, as the elements that are affected respond to the event and its consequences propagate through the model.

Two quite different approaches can be taken to represent a pivotal event. The simpler of these is to add a 'Random Choice' element, which has a set of user-defined outcomes with associated probabilities. Upon receiving notice of an event this element 'rolls the dice' to randomly select which outcome should occur, and emits an event signal from that output.



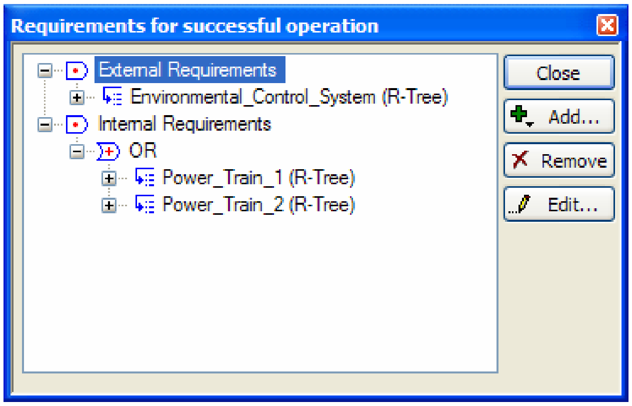
The alternative approach is to add a reliability element, and simulate its state dynamically. When the precedent event occurs this element may be operating normally, or failed, or undergoing maintenance or repair, or inoperable because of a missing requirement, and so on. The Monte Carlo process effectively samples these possibilities as it cycles through a number of realizations. Compared to a simple Random Choice element the reliability element requires much more input data, and is more complex computationally.

The modeler thus has to choose between the simplicity of a Random Choice element and the verisimilitude of a reliability element. Typically, Random Choice elements will be used in preliminary versions of a model, and replaced by reliability elements in later versions. The alternatives of using a simple Random Choice element or using a Reliability element are discussed further in section III.D below.

III.C Modeling Fault-trees in GoldSim

In a sense, the reliability elements themselves represent nodes in the total system's logic tree. Their internal requirements trees define the logical linkages to the next lower level in the total system tree, and their external requirements trees define the linkages to their peers in the total system tree.

Figure 3 shows the linkages for a single element, in this case a link to an "Environmental_Control_System" which is a peer, and an OR-gate for two child components, "Power_Train_1" and "Power_Train_2".



Requirements for successful operation

External Requirements
Environmental_Control_System (R-Tree)
Internal Requirements
OR
Power_Train_1 (R-Tree)
Power_Train_2 (R-Tree)

Fig. 3. Example of a Requirements-tree in GoldSim.

The GoldSim 'logic tree' is fully dynamic, as the model's configuration can change in response to the evolving state of the system, for example during repair or maintenance operations, or by switching to backup systems when a primary system fails. Another dynamic behavior is the possibility for the consequences of an event to be immediate or delayed, if the initial event(s) change the system's state such that other events occur subsequently.

III.D Component States

When an initiating event sets a train of pivotal events in motion, the outcome usually depends on the states of some physical components of the system. If a component is not operating when required, negative outcomes may ensue. All reliability analysis methodologies require probability distributions for the different possible states of each component.

In the simplest case, for a non-repairable component that has only operating and failed states, with a constant hazard (failure) rate, the probability that the component is operating when the event occurs simply equals its projected reliability. The reliability may be based on an exponential, Weibull, or other standard failure distribution. This approach is readily applied in the simulation approach, using the Random Choice element.

Another simple approach is useful for systems that are in a steady state, with failures being repaired with a known mean time to fail (MTTF) and mean time to repair (MTTR). For a constant hazard rate and a simple component whose only behavior is to be repaired if it fails, the probability that it is operating at any given time equals its expected availability, (MTTF / (MTTF + MTTR)). The Random Choice element can again be used for such cases.

However, if the hazard rate is not constant, or if the component has more complex failure, repair, switching, or maintenance behaviors, such simple approximations may not be adequate. For example, what if the component is normally repaired when it fails, but the necessary spare part may not be available? What if its aging rate depends on its operating environment? The flexibility of the simulation approach is apparent for cases such as these, as instead of using a simple Random Choice element the simulator can readily do a more realistic simulation of the component and its interactions.

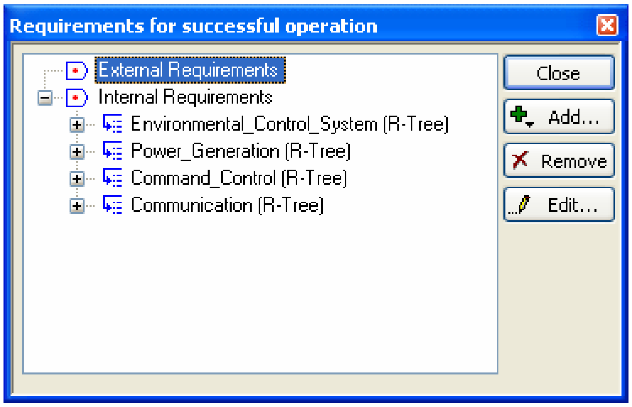
III.E Case Study Models

The two models built as part of the project are based on example models from Chapter 15 of the *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners* (Stamatelatos et al, 2002). The two examples are illustrative only, and do not represent complete PRA's. As described below, some additional assumptions were made for the two GoldSim models in order to illustrate specific simulation capabilities.

Lunar Colony Example

The first model is of a lunar colony over its 20 year mission. The purpose of the model is to determine the probability that the mission will successfully achieve all the mission goals, along with the probability of two undesirable end states (Loss of Mission and Loss of Crew).

The safe operation of the base is dependent on four major subsystems: Environmental Control, Power Generation, Command and Control, and Communication. If any of these systems fails, or if the structural integrity of the base is compromised, evacuation is required. The base's dependence on subsystems is modeled using a requirements tree (shown below in Figure 4), and its structural integrity is modeled as a failure mode that can be triggered by certain initiating events.


Requirements for successful operation

- External Requirements
- Internal Requirements
  - Environmental_Control_System (R-Tree)
  - Power_Generation (R-Tree)
  - Command_Control (R-Tree)
  - Communication (R-Tree)

Fig. 4. Lunar Base Top-Level Requirements-tree.

Evacuation is modeled by another reliability event triggered by the failure of the base, or the consequences of certain initiating events. The mission is always lost if evacuation is required, and if the evacuation fails it leads to Loss of Crew.

The scientific mission of the base is dependent on a number of instruments, some of which can be replaced during a resupply mission (once every 60 days), and others which cannot be replaced. If the irreplaceable scientific instruments are destroyed, it leads to loss of the mission.

In addition, there are four initiating events that can cause failure of the base: an energetic event; an atmospheric leak; an electrolyte leak; and a smoldering wire event.

The smoldering wire event sequence from the example model's PRA and its GoldSim counterpart are shown below in Figures 5 and 6. In this case, Decision elements referencing component states, and a Random Choice element, are used to model the system's response to the initiating event.

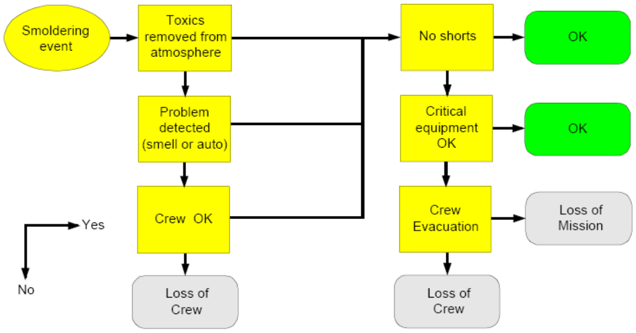


Fig. 5. Smoldering Wire Event Sequence Diagram [1]

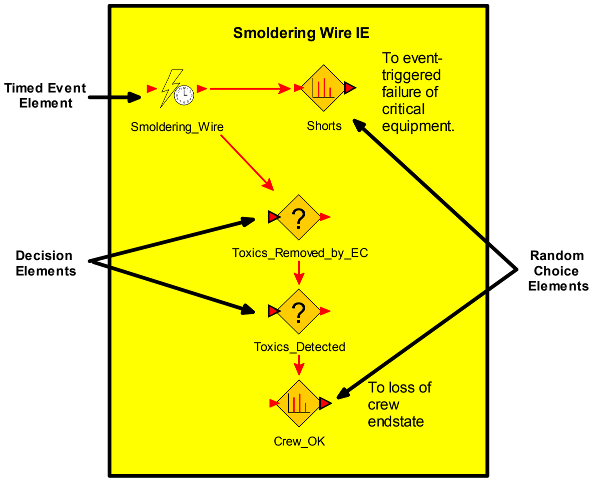


Fig. 6. Smoldering Wire Event Sequence Diagram Screen-shot from the GoldSim Model

The smoldering wire event occurs with a Poisson distribution with a frequency of 1/10 years. It has two direct consequences: release of toxic fumes into the base atmosphere, and the possible generation of electrical shorts.

In contrast to the ESD approach, in the simulation model both potential consequences are immediately triggered. With regard to the toxic fumes, the current status of the filtration system is queried (as described in III.D) using a Decision element that references reliability elements in the Environmental Control system. If the system is fully functional, the toxics will be removed. However, if the system is partially failed (in a compromised state but functional enough to safely support the astronauts), the toxics will not be removed, and GoldSim will proceed to the next Decision element, which queries a reliability element representing an automatic toxic detection system. If it is operating, the crew will be notified and will take action to remove the toxins. If not, GoldSim proceeds to a Random Choice element that determines whether the crew is able to detect the presence of toxics by smell. If they do, they can take action to deal with the problem. If they do not, it leads to the Loss of Crew end state.

The electrical shorts consequence uses a Random Choice element to determine whether or not electrical shorts occur. If they do, there is a probability that the shorts could trigger a failure of one or more major base systems. If failures cause the base's requirements tree to evaluate to false, the evacuation reliability element is automatically triggered (leading to one of the two undesirable end states).

The atmospheric leak and energetic event ESDs are represented in GoldSim using similar techniques. The electrolyte leakage event is modeled somewhat differently, because crew reactions and behavior are taken into account. This portion of the model is discussed in more detail in Section V.

The GoldSim model contains 100 elements, and took approximately four days to build. It takes approximately 79s to run the model with 1000 Monte Carlo realizations on an Intel Pentium 4 1.6 Ghz laptop with 1 Gb of RAM. The model file (including results) is approximately 13Mb in size.

In addition to providing information on the failures of each component of the base, the model also provides the probability of crew loss and the probability of mission loss (with confidence bounds on each). These values are:

- Loss of Crew:
  Probability: 0.083
  5/95% Confidence Bounds 0.069 / 0.097
- Loss of Mission Before 20 Year Goal
  Probability: 0.478
  5/95% Confidence Bounds 0.452 / 0.504

Additional Monte Carlo realizations could be performed, if desired, to narrow the confidence bounds.

The model also provides standard reliability statistics and their values over the course of the simulation. An example is the plot of reliability versus simulation time, shown below in Figure 7. This figure shows an initial high-reliability "honeymoon" period when the base first starts operating. However, after the first year of service, the base begins to exhibit more steady-state behavior.

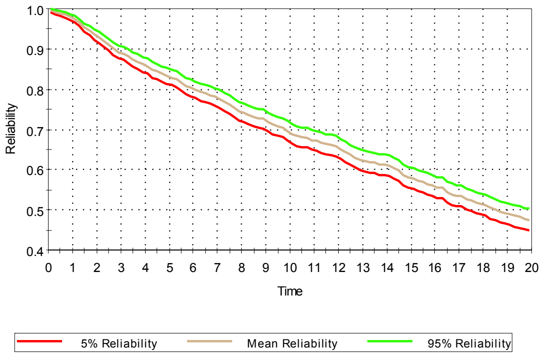


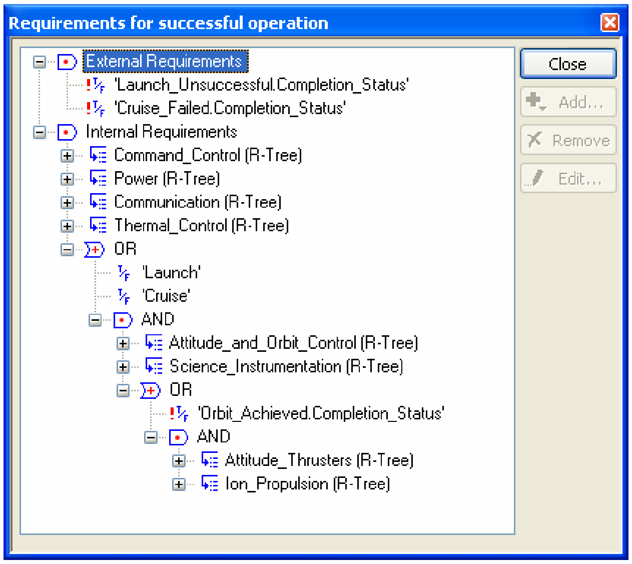
Fig. 7. Plot of Lunar Base Reliability Over the 20 Year Mission

Unmanned Exploration Mission Example

The second model is of an unmanned scientific mission to another planet. There are a number of phases (launch, cruise, orbital insertion, lander descent and scientific mission), and the requirements for successful operation change as the mission evolves. The key outputs from the model are the proportions of missions where minimal and full scientific goals are achieved.

To successfully launch the spacecraft (consisting of an orbiter and a lander), the rocket engines (solid rocket boosters, main engines and upper stage), must function when required during the launch sequence. They must also successfully separate at the appropriate time. Any failures lead to loss of vehicle.

If the spacecraft is launched successfully, the cruise phase begins. During this phase, only certain spacecraft systems are required for successful operation (e.g., the orbit control system and scientific equipment are only required when in orbit). The orbiter thus has a "dynamic logic tree" (shown in Figure 8) with a branch that becomes active once the Orbiter has reached the target planet, and another that becomes active once orbit has been achieved.



- External Requirements
  - 'Launch_Unsuccessful.Completion_Status'
  - 'Cruise_Failed.Completion_Status'
- Internal Requirements
  - Command_Control (R-Tree)
  - Power (R-Tree)
  - Communication (R-Tree)
  - Thermal_Control (R-Tree)
  - OR
    - 'Launch'
    - 'Cruise'
    - AND
      - Attitude_and_Orbit_Control (R-Tree)
      - Science_Instrumentation (R-Tree)
      - OR
        - 'Orbit_Achieved.Completion_Status'
        - AND
          - Attitude_Thrusters (R-Tree)
          - Ion_Propulsion (R-Tree)

Fig. 8. Orbiter Requirements-tree Showing Dynamic Branch

During the cruise phase, the path of the spacecraft can be disturbed due to a number of causes (e.g. micrometeoroid debris, solar winds). These "nuances" will require the spacecraft to perform a correction, with its attitude thruster and ion engine consuming a certain amount of their propellant reserves.

If the cruise phase is successful, the spacecraft wakes up its remaining systems and begins insertion into orbit. Consumption of propellant during the orbital insertion is modeled, and this methodology is described in more detail in Section VII.

If the spacecraft achieves orbit, the lander portion of the spacecraft will attempt to separate. If it successfully separates, the lander's engine will be started briefly to deorbit the spacecraft. After a few minutes delay, the engine will be restarted to slow the lander's descent, and the lander's parachute will be deployed. If the lander engine successfully completes its burn schedule and the parachute deploys successfully, the landing is a success and the scientific mission can begin.

Once on the surface, the scientific mission is intended to last three years. The lander has two key pieces of scientific equipment on board: a mass spectrometer and a rock abrasion tool. Successful completion of minimal mission requirements requires that either the mass spectrometer or rock abrasion tool be active for the first year on the surface. Completion of all mission goals requires the rock abrasion tool to be active for three years on the surface. Again, this is modeled using a dynamic fault-tree (similar to that used for the orbiter).

The GoldSim model contains 133 elements, and took approximately five days to build. It takes approximately 60 seconds to run the model with 1000 Monte Carlo realizations on an Intel Pentium 4 1.6 GHz laptop with 1 Gb of RAM. The model file (including results) is approximately 6Mb in size.

In addition to the failure modes and reliability information calculated by each reliability element, the model also predicts the probability of mission success, partial success, or failure. These probabilities are:

- Full Mission Success
Probability: 0.533
5/95% Confidence Bounds: 0.507 / 0.559

- Partial Mission Success (Minimum Goals Met)
Probability: 0.175
5/95% Confidence Bounds: 0.155 / 0.195
- Mission Failure
Probability: 0.292
5/95% Confidence Bounds: 0.268 / 0.316

Again, the model also provides standard reliability statistics and their values over the course of the simulation. Figure 9 below shows the reliability of the orbiter over the duration of the simulation. It shows that in some (~4%) of the realizations the orbiter is destroyed during launch. Relatively steady-state behavior is shown until 1.5 years (the end of the cruise phase) where there is a drop in reliability as the spacecraft is inserted into orbit and previously unused systems are started. After this point, the orbiter exhibits an approximately steady-state failure rate. During this final phase the orbiter is actually less reliable than the lander.

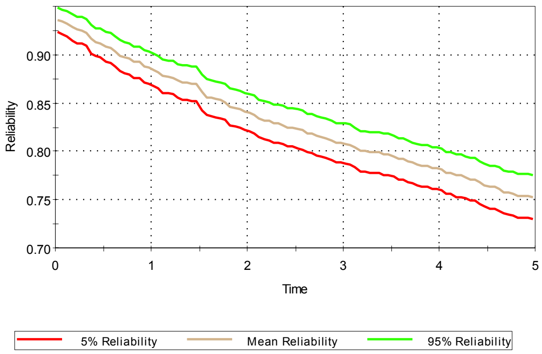


Fig.9. Orbiter Reliability over the Duration of the Simulation

IV. UNCERTAINTY ANALYSIS

While there are many ways to characterize uncertainty, it is conventional to subdivide the sources of uncertainty into two broad categories [5]:

- Aleatory uncertainty refers to random processes, such as random failures of components or random events like solar flares. Aleatory uncertainty is also referred to as variability, as one random sample will differ from another.
- Epistemic uncertainty refers to a state of limited knowledge. This would apply, for example, to the failure rate for a type of component that had only limited data available, or to a process that is poorly understood. Epistemic uncertainty is also referred to as true uncertainty.

A key conceptual difference between aleatory uncertainty and epistemic uncertainty is that in principle epistemic uncertainty can be reduced, through additional testing or analysis, while aleatory uncertainty cannot be reduced.

GoldSim can carry out three distinct types of uncertainty analysis. The first type is **variability analysis**, where multiple system realizations are made using fixed input data, with only random (aleatory) processes affecting the results. The outcome of this type of analysis is frequency distributions for the different possible system outcomes, as indicated in Figure 10 below, assuming perfectly-known input data.

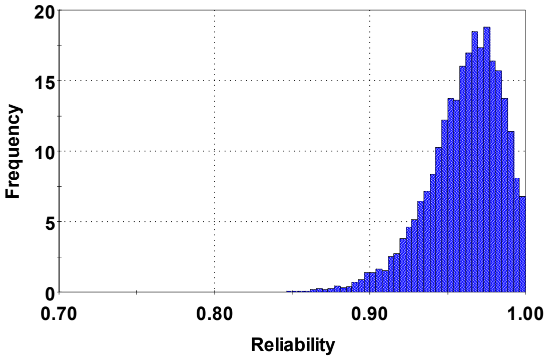


Fig. 10. Example of a Frequency Distribution of a Random (Aleatory) Process

Variability analysis is the approach that would typically be used to assess the reliability of a complex system.

In the second approach, known as **mixed variability and uncertainty analysis**, the model also randomly samples any uncertain (epistemic) input parameters at each realization of the system, and the resulting probability distributions then reflect both the uncertainty in the random performance of the system and also the uncertainty in the input data. For example, the mean life of a component may be uncertain, and would be sampled from a probability distribution. This might be termed the 'insurance view' of the system, as it represents the best estimate of the likelihood of the different outcomes. However, by combining the two fundamental types of uncertainty into a single analysis, some important information may be lost.

As part of the current project a third type of uncertainty analysis, **separated uncertainty and variability analysis**, was added to GoldSim. In this analysis approach the effects of the epistemic and aleatory uncertainties are clearly distinguished. This is done by embedding an inner 'variability' Monte Carlo model within an outer 'uncertainty' model. A typical result is shown below in Figure 11, representing the uncertainty in a statistical measure of the system's performance, the probability of a total failure of the planetary exploration mission. This analysis is based on the following assumptions for five key uncertain inputs to the model:

1. The frequency of cruise nuances is lognormally distributed with a mean of 3 events per year and a standard deviation of 1 event per year.
2. The frequency of major solar events is assumed to be uniformly distributed between 0.1 and 0.75/yr
3. The probability of the lander successfully deploying its parachute is normally distributed with mean 0.99 and standard deviation 0.003.
4. The rate of propellant consumption by the ion thruster while in orbit is normally distributed with mean 5 kg/yr and a standard deviation of 1 kg/yr.
5. The rate of propellant consumption by the attitude thrusters in orbit is normally distributed with mean 5 kg/yr and a standard deviation of 1 kg/yr

This might be termed the 'decision analysis' view of the system, as it illuminates whether there is sufficient confidence in the input data to support a decision to go ahead with the mission. If there is not sufficient confidence, possible alternative decisions would be to:

a) Get more or better data,
b) Switch to better understood components (i.e. with lower uncertainty about their performance), or
c) Switch to components with higher safety margins.

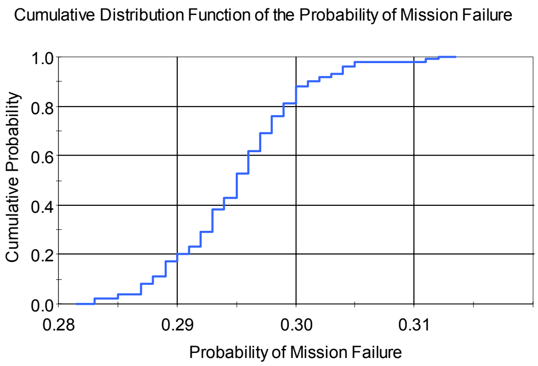


Fig. 11. Probability of Mission Loss From a Separated Uncertainty and Variability Analysis

This type of randomness vs. uncertainty analysis is also supported by typical PRA software, which also uses an outer Monte Carlo loop to sample epistemic uncertainties for an inner fault-tree model.

V. SIMULATING HUMAN BEHAVIOR

The simulation approach allows for a straightforward representation of the effects of human behavior on the system. This was illustrated using the lunar base example, by adding a scenario where crew action may be required in order to react to and repair a rupture of one of the two electrical storage battery systems.

In this scenario, the crew is required to notice and respond to the failure, taking appropriate action to mitigate the effects of any electrolyte release before irreparable damage is done to the scientific instruments or to the other battery system- either of which would result in loss of the mission. However, there is a possibility of the crew responding in an inappropriate way, so that the damage occurs even though the response was timely. To summarize the scenario:

- Electrolyte leakage from a battery train occurs according to an exponential distribution once every 80 years.
- After failure of one of the systems, the crew has to respond to mitigate any damage due to released electrolyte. Until the crew responds there is an increasing probability of irreparable damage (to the other power train and to scientific equipment). Damage to the scientific equipment is defined by P(damage) = 1 – $e^{-t/25 \text{ min}}$, where damage to the other power train is defined by P(damage) = 1 – $e^{-t/20 \text{ min}}$, where t is the delay time before the response occurs.

- There is a variable amount of time between the initial battery failure and the crew noticing it – they may be asleep, or distracted by other activities. This time was defined by a lognormal probability distribution with a mean of 5 minutes and a standard deviation of 2 minutes.
- Once the crew notices the failure, they may respond promptly or, if they are unsure of the correct response, either refer to their procedural guides (mean delay time is 10 minutes with a standard deviation of 3 minutes) or request advice from the earth base (mean delay time is 12 minutes with a standard deviation of 2 minutes). These three possibilities were given equal probabilities.
- If the crew responds promptly and correctly there is a reduced likelihood of irreparable damage due to the failure. However, a prompt crew response has a significant likelihood (25%) of being wrong. If the crew takes the wrong action, it takes a mean of 15 minutes for them to realize that their action was incorrect, and then they must contact the earth base for advice before taking further action.
- If the crew refers to procedural guides or contacts the earth base station before taking action, the correct response is assumed, but there is an increased likelihood of irreparable damage due to the additional time involved.
- In all cases, successful remediation of the electrolyte leak requires 10 minutes, with a standard deviation of 2 minutes.

A cumulative probability plot of crew response times in the model is shown below in Figure 12.

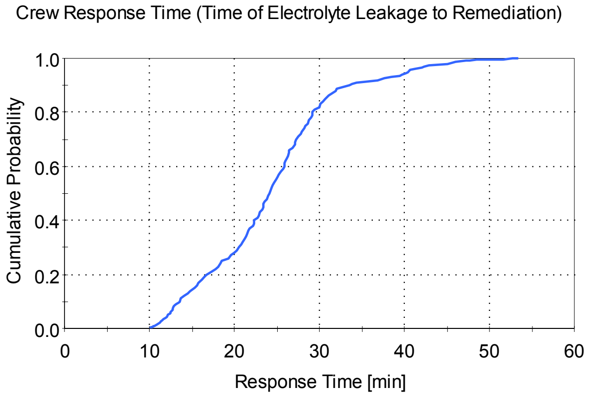


Fig. 12. Cumulative Probability Plot of Crew Response Times Following Electrolyte Leakage Event

VI. OPTIMIZATION

The GoldSim software also supports optimization, and this was demonstrated in the lunar base model, by adding logic that optimizes the number of spare air filters that should be on hand at the base and the age at which unfailed filters are replaced. This optimization had to balance the costs of shipping extra spares to the base, and storing them there, against the risk of losing the base, and potentially the crew, due to being unable to effect repairs when necessary. This is done by combining all three considerations into an objective function, which is minimized by the optimizer.

The objective function was arbitrarily set equal to the total costs for the additional filters, plus $5E4 for each maintenance event, plus a penalty term to discourage actions that could impact safety. The penalty term was equal to $5E6 for each filter failure, plus $1E6 for each time there were no spare filters on hand, plus $1E5 for each preventative maintenance event (as this takes one of the redundant air filters offline for a period of time). (Note that the implied relative rankings of these factors are purely fictional).

The failure distribution assumed for the filters was a Weibull distribution, with a mean life of 120 days and a slope factor of 10.

The replacement and ordering rules are:

- If replacement filters are available, failed filters are immediately replaced.
- If a filter is unfailed the filter is replaced after an in-service interval which was determined by the optimizer.
- Replacement filters must be ordered 20 days in advance of the arrival of the supply ship, and the maximum filter storage capacity cannot be exceeded.

GoldSim uses Box's complex method to optimize the solution, running the simulation a number of times. The optimal result (after 82 outer loop simulations) is as follows:

Objective Function Value:	$3.55 E7
Replacement Age:	67.5 days
Number of Spare Filters:	4

VII. SIMULATING CONTINUOUS PROCESSES

The simulation approach can readily incorporate continuous processes in addition to the discrete events involved in failures and repairs. This was illustrated in the interplanetary exploration model, where the amount of propellant remaining in each of the three spacecraft thrusting systems (attitude control system, ion propulsion system, and primary chemical rocket system) was tracked.

During the cruise phase of the mission, correcting nuances caused by micrometeoroid hits or other events consumed a stochastic amount of ion engine and attitude thruster propellant reserves. This was modeled using discrete withdrawals from those reserves.

The model of fuel consumption during the orbital insertion is more complex. Instead of discrete withdrawals from the propellant reserves, the change in speed and the amount of fuel consumed is modeled. When the vehicle begins the orbital insertion burn, GoldSim samples its approach speed, and the velocity change required to enter orbit is computed. The chemical engine is started, and the momentum change it provides is calculated (a fixed propellant flow and relative exhaust velocity is assumed). The change in speed this imparts is calculated by dividing by the mass of the spacecraft (which decreases as propellant is consumed).

The chemical engine is jettisoned when the spacecraft achieves orbit or if the propellant is exhausted. In the latter case the spacecraft will then continue to attempt to achieve orbit using the ion engine. Both the ion engine and attitude thrusters are required while the orbiter is at the target planet. Thus, the frequency of nuances during the cruise phase, along with the potential need to use the ion engine to enter orbit, can abbreviate the mission length. The time history probability plot in Figure 13 shows the consumption of ion engine propellant.

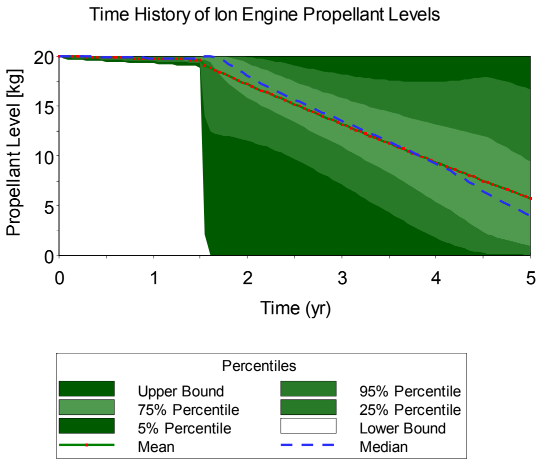


Fig. 13. Time History of Ion Engine Propellant Levels

Figure 14 shows a CDF of the amount of ion engine propellant remaining at the end of the mission. Note the small probability (~1%) of completely consuming the propellant, which results in termination of the mission.

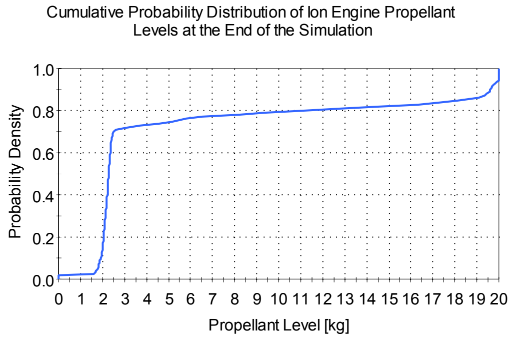


Fig. 14. Cumulative Probability Distribution of Ion Engine Propellant Remaining at the End of the Simulation

VIII. ACCURACY

The simulation approach can produce approximate results quite quickly, with relatively few Monte Carlo realizations. However, achieving highly precise results can require large numbers of realizations (accuracy typically increases with the square root of the number of realizations). As a result, the simulation approach may not be practicable for studying the performance of extremely reliable systems, unless some form of importance sampling is used.

Note that the number of realizations required is a function of the **system's** reliability, not the reliability of its components. A system with a thousand non-redundant modeled components, each having a reliability of 0.999 999, could have an overall reliability in the order of 10^{-3} and be suitable for Monte Carlo modeling using in the order of 10^{4} realizations.

To assist in evaluating the accuracy of its results, the GoldSim software can automatically show confidence bounds for its results, which indicate whether sufficient Monte Carlo realizations were carried out.

The conventional PRA approach is not subject to computational approximations, which is a clear advantage. However, the price of this precision is the necessity to make numerous simplifying assumptions when setting up the model- a different source of error with effects which may be hard to quantify. Also, if there is significant uncertainty about the system's parameters, a highly precise result may not be required.

IX. EFFICIENCY

The simulation approach can be demanding in terms of computer resources. Table II shows representative run times for the cases considered in this paper. The Lunar Base and Unmanned Mission to Another Planet simulations were carried out using a Dell Latitude computer with a 1.6 GHz Intel Pentium 4 processor, while the Double Monte Carlo and Optimization case were carried out a Dell Poweredge Server with a 2 GHz Intel Xeon processor.

TABLE I. Run Times for Various Cases

Case	Number of Realizations	Run Time
Lunar Base	1000	79 seconds
Unmanned Mission to Another Planet	1000	60 seconds
Separate Uncertainty and Variability Analysis	100 outer loops each with 1000 inner loops	2 hours 27 minutes
Optimization	82 outer loops each with 1000 inner loops	2 hours 59 minutes

Larger, more realistic models will demand significantly more computing resources than these simple examples. In the event that the run-times become inconveniently long, it is possible to connect a cluster of computers over a local area network so as to share the burden of Monte Carlo computations. This approach is scalable, and can reduce run-times for large models by orders of magnitude.

X. CONCLUSIONS

The dynamic simulation approach used for nuclear-waste performance assessments has been adapted for application to mission risk and reliability analysis. Two illustrative NASA examples used for conventional PRA analyses were represented using the simulation approach.

It was found that the two examples were readily represented using the simulation approach. The resulting models ran quickly (in the order of one minute each), and produced both quantitative assessment of the mission risk and reliability and also insight into failure mechanisms and sensitivity to inputs.

Most of the logical content of the two NASA PRA example models could be incorporated into the simulation model using simple element types, such as the random event generator and the random choice element. The reliability element types provided an effective mechanism to represent the dependencies between system components. Some of the more powerful capabilities of the simulation approach were added to the models, to illustrate such processes as simulating continuously-varying system parameters (propellant levels), human behavior and its effects, and optimization of spares levels.

The simulation approach to mission risk and reliability analysis was found to be a practical for the two case studies considered, and may provide a valuable complement to the PRA approach. It is a flexible and transparent approach which represents highly dynamic and nonlinear system behavior in a natural way.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of NASA, and the technical advice provided by Dr. Steven Cornford of JPL

REFERENCES

1. M. STAMATELATOS et al., “Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners”, NASA, 2002.
2. W. VESELY et al., “Fault Tree Handbook with Aerospace Applications”, NASA, 2002.
3. G. APOSTOLAKIS, “How Useful is Quantitative Risk Assessment?”, Risk Analysis, Vol. 24, No. 3, 2004.
4. P. LABEAU. et al. “Dynamic Reliability: Towards an Integrated Platform for Probabilistic Risk Assessment”, Reliability Engineering & System Safety 68 (2000) pp. 219 – 254.
5. J. C. HELTON, “Uncertainty and sensitivity analysis in performance assessment for the Waste Isolation Pilot Plant”, Computer Physics Communications, vol. 117, Issue 1-2 (1999), pp.156-180.