Use Cases of Repairable Systems Simulation Platform

Peng Liu, JMP Principal Research Statistician Developer, SAS
Leo Wright, JMP Principal Product Manager, SAS

Abstract

Repairable Systems Simulation (RSS) is a new addition to JMP capability in reliability engineering. RSS is designed to solve simple to very sophisticated problems. The innovative graphical user interface of the platform offers engineers great freedom in expressing complex maintenance strategies. The presenter will first demonstrate the basics of using the platform to help the audience familiar with the tool and its concept. The presentation will walk through several use cases to illustrate the capability of the platform. Use cases will demonstrate the usefulness in a variety of applications, including identifying critical component to the system reliability, computing reliability of large non-repairable systems, comparing costs of different maintenance strategies, and changing component state to mimic environment impact to system reliability, etc.

Introduction

Repairable Systems Simulation (RSS) platform is an engineering tool to study repairable reliability systems using discrete event simulation. A repairable reliability system consists of a reliability block diagram (RBD) and maintenance arrangements.

RBD is a graphical representation of the reliability function of the system. RBD is an acyclic directional diagram. The vertices of the diagram usually denote individual components of the system. Vertices are called blocks in the RBD terminology. The edges denote the dependence among the blocks. Software implementations usually add a "Start" block and an "End" block to the diagram to define reliability. A system is called reliability if there is a path from the "Start" block to the "End" block. If such a path does not exist, the system is called unreliability. Sometimes, people use system states exchangeably to describe reliable and unreliable. For example, "system is on" means "system is reliable.", and "system is down" or "system fails" means

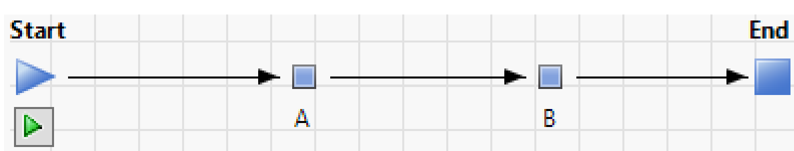

Figure 1 Series System

"system is unreliable". Figure 1 is an example of series systems. A series system is on if and only if all the components are on. For example, if component "A" is down, it is equivalent to remove "A" from the diagram, such that the path from "Start" to "End" is broken, and the system is down. RBD is sufficient for describing non-repairable systems. The failure of a component is equivalent to removing the component from the diagram. And the system fails when we cannot find a path from "Start" to "End".

Repairable systems involve system maintenance, such as replacing components, repair components, adjust settings, etc. In addition, maintenance is operated based on certain arrangements. For example, a component may be replaced after it fails. Or it can be replaced before it fails, which is known as preventive maintenance. In RSS, maintenance is called "action", and the timing of a maintenance is prescribed by a "event". Both "action" and "event" are represented by additional "nodes" to RBD. In addition, multiple maintenance tasks can be operated upon a single event, either in a sequence, or simultaneously. That is achieved by connecting "action" nodes and "event" nodes" with edges. The "action" nodes, "event" nodes, and edges among them form an event-action sub-diagram. Figure 2 is an example of repairable reliability systems with

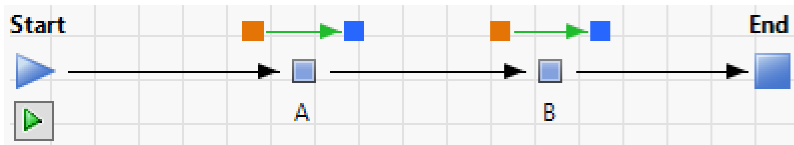

Figure 2 RSS Example

maintenance arrangements. The orange nodes denote failure events of the two components. And the dark blue nodes denote maintenance actions that need to be performed upon the occurrences of corresponding events. The green edges establish the correspondence between appropriate events and actions. The details of events and actions are revealed in the interface during the interaction with the software.

This paper presents several use cases of the platform.

A Gold Miner's Mission

This example is based on a hypothetical gold mining situation. The miners need to keep one processing equipment running uninterrupted as long and often as possible. This example demonstrates how to build the system and its maintenance arrangement from scratch. The example will navigate through all

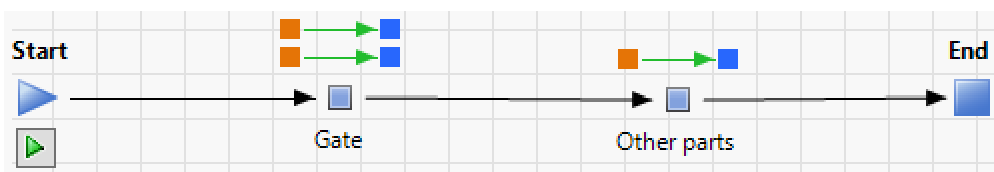

Figure 3 A Gold Miner's Mission

the graphical user interface elements, describe the simulation result format, and briefly present the default analysis report of the simulation.

The first step of creating an RSS diagram is to create an RBD which represents the system. In this example, a piece of big gold mining machine is represented by a series diagram of two blocks. One block represents a part which requires replacement from time to time. And the other block represents the rest of the machine, which is much less frequent to fail.

Several maintenance arrangement scenarios will show how effective a preventive maintenance can be. In terms of availability, the system can be largely improved, such that the gold miner can improve the productivity.

A few words on simulation results. A simulation result is saved in a JMP data table, as a collection of events and actions. The data table has six columns. In another word, every record has six fields. Every

Sim ID	Time	Subject	Predicate	State	Note
1	337.39856494	Other parts	Machine Failure	Down	
1	337.39856494	System	Turn Down System	Down	Unintended
1	337.39856494	Other parts	Minimal Repair ...	Removed	
1	337.39856494	Other parts	Minimal Repair ...	Start	
1	457.39856494	Other parts	Minimal Repair ...	On	
1	457.39856494	System	Turn On System	On	Automatic
1	457.39856494	Other parts	Minimal Repair ...	Finish	
1	1560	Gate	Inservice Based		

Figure 4 Simulation Result

record has a simulation ID. Every event or action occurs at a specific time. Every event must be initiated by a specific subject or the system. The name of event and action acts as a predicate. State change may be recorded in the fifth field. The last field records miscellaneous comment.

The data table has a built-in script which will summarize the simulation result with an emphasis on system availability. The data table holds all the information for inspections from all angles.

An Expensive Refrigerator

In this example, an expensive helium refrigerator is the subject of interest. The fridge consists of many common off-the-shelf parts and two imported heat exchangers which must be imported. This example represents a typical use case of an availability study using the platform. The default report of a simulation is mainly designed for the availability study.

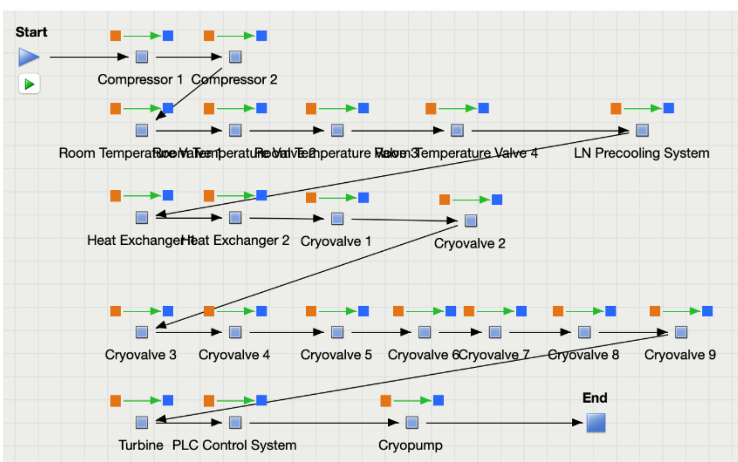

Figure 5 An Expensive Refrigerator

In this example, an expensive refrigerator is modelled by a series diagram. Simple corrective maintenance plan is applied. In another word, parts are replaced upon failures. However, purchasing and replacing off-the-shelf components require little time, purchasing and replacing a heat exchanger require much longer time. Results in the final report reflect the situation. The example will show how to interpret the results in the report.

A Chemical Plant Investment

The default report on the simulation result focuses on the system availability. This example demonstrates an advanced use of the default report, which calculates quantities that are based on the availability statistics. The quantity of interest can be different from use case to use case, which is not general, and cannot be easily foreseen by the software.

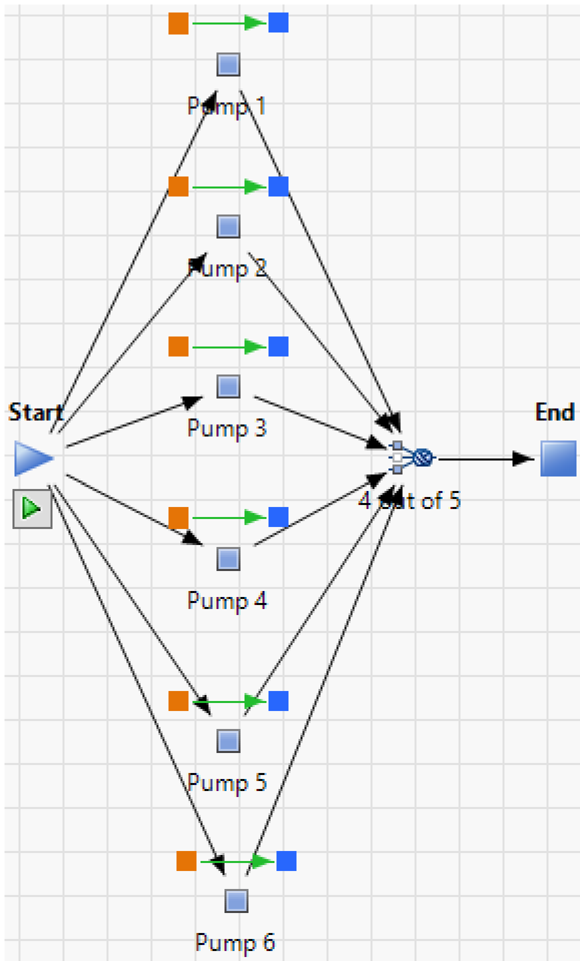

Figure 6 A Chemical Plant Investment

In this example, an investor needs to decide how to invest in a recently purchased chemical plant. The chemical plant has multiple water pumps, which are required to operate simultaneously to meet certain safety requirement. If there are not enough working pumps, the plant must shut down. The investor must calculate the initial cost of the investment, the maintenance cost and the revenue loss, to see the net const over the period of many years. Some quantities are related to the system availability, some are related to component availabilities, and the calculation of investment is independent of the repairable systems simulation.

Simulate Seasonal Effect

There are features that do not existing in competing software. This example illustrates one of such features. In competing software, the simulation is limited to a certain number of common tasks, such as corrective maintenance and preventive maintenance. The corrective maintenance means that a certain type of maintenance must be done upon component failure. The preventive maintenance means that a type of maintenance must be done according to predetermined schedules. The maintenance options are usually limited to one of the two types: replace with new and minimum repair. The design is simple and effective but there are so many real-world scenarios that are impossible to simulate under the limitation.

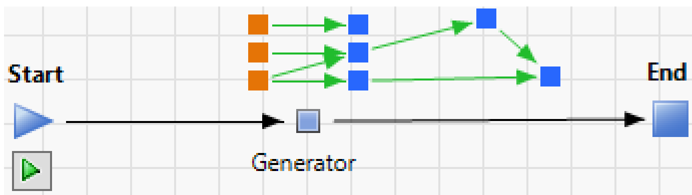

Figure 7 Simulate Seasonal Effect

The JMP Repairable Systems Simulation provides flexible ways of customizing the system design and maintenance choices. This example illustrates a trivial one-component system with a sophisticated seasonal behavior. The system is a power generator. Because of seasonal consumption patterns, it runs under high stress for half of the year, and under low stress for the other half of the year. The generator has a higher likelihood to fail during the high stress season than the lower stress season. The collection of events and actions in the software and the event-action sub-diagram feature allow engineers to create sophisticated logics to model the real system and maintenance arrangement.

Conclusion

The Repairable Systems Simulation is a flexible tool for simulating the behavior of a repairable reliability system. It easily answers the standard questions regarding availability studies. It is powerful and flexible to address many other problems in more sophisticated settings. Its patent pending design is innovative, but also leaves room for future improvement, when the tool meets applications. We encourage users to explore the potential of the tool and provide us feedbacks, so we can further improve the software.