SYM-AM-25-426

Excerpt from the Proceedings

of the

Twenty-Second Annual Acquisition Research Symposium and Innovation Summit

Volume III

Identifying Downtime Drivers Using SIMLOX Simulations to Rapidly Develop Solutions Improving System and Mission Readiness

Published: May 5, 2025

The research presented in this report was supported by the Acquisition Research Program at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

Identifying Downtime Drivers Using SIMLOX Simulations to Rapidly Develop Solutions Improving System and Mission Readiness

Marissa McLaren—is an Analyst at Systecon North America working on lifecycle management and sustainment for complex technical systems. Utilizing Systecon's proprietary software for optimization and simulation modeling, she has worked with PEO IWS, NAVSUP, NAVSEA, NAVAIR, and JPO. She has an MS in Engineering and Public Policy from Carnegie Mellon University, and a BS in Mechanical Engineering from Georgia Institute of Technology. [marissa.mclaren@systecon.us]

Abstract

In this paper we present a simulation model that directly relates the system results to the individual parts and resources, otherwise known as downtime drivers. Understanding operational downtime drivers and whether they are item- or subsystem-specific is crucial to making strategic decisions for missions and operations. Identifying the drivers in the modeling phase allows for increased preparation and problem-solving to improve mission requirements. While working to solve the issues created by downtime drivers, industry and defense can work together to determine a reasonable solution to overcome the impact an item or subsystem can have on the overall system.

This case study describes a scenario where industry and defense have been able to identify downtime drivers for a complicated system and develop a set of reasonable alternatives to address these issues. OPUS10 identifies the initial spares purchase optimization for a given availability requirement. We can then utilize that recommendation in SIMLOX, a Monte Carlo–based simulation tool. Simulation results are often used to identify bottlenecks within the supply chain, spares, and support organization. With the recent software updates to SIMLOX, we can identify downtime drivers. Stakeholders can identify which subsystem(s) or items are causing a system to be down.

Keywords: Simulation, Downtime Drivers, System Readiness, Solution-Focused Modeling

Background

Systecon North America provides software support, training, and consulting for their proprietary software, OPUS Suite. OPUS Suite comprises three main software solutions. These include OPUS (optimization tool), SIMLOX (simulation tool) and CATLOC (cost control tool). Systecon has also developed EVO (tactical operation), INSIGHTS (business intelligence), and CONNECT (integration). Having a basic understanding of OPUS and SIMLOX will be beneficial to understanding the benefit of rapid downtime driver identification.

OPUS is a tool used for cost-effective spare parts steady-state optimization, balancing the spare part investment while also maintaining or increasing system readiness. Beyond just spares optimization, OPUS allows the user to evaluate support solutions, technical systems, and scenarios (Systecon Group, 2020*)*. OPUS is utilized across the Department of Defense as a tool for spares part purchases, such as the Navy Common Readiness Model, as well as for stress testing what-if scenarios for large programs such as the F-35 Program (Systecon North America*,* 2019).

SIMLOX is an event-driven simulation tool that allows detailed analysis of technical systems' performance over time, while factoring in varying operational and logistics support scenarios (Systecon Group*,* 2018). Since SIMLOX includes the time component, you can identify weaknesses in the support structure during a peak utilization period. The flexibility allows one to model any complex technical system. For example, their varied usage and utilization, varied support capabilities over time, complex functional block diagrams, performance targets, confidence intervals, and, most important for this paper, downtime drivers.

Introduction

Complex defense systems with intricate missions rely on cost optimization and simulation models to determine the best set of variables to meet mission and system readiness requirements. Linking system properties and logistics support to understand system performance is a multidimensional optimization problem over multi-indenture systems that requires an iterative approach. In a basic model, one could factor each variable to understand the overall system performance. However, for complex systems and support structures, an analytic solution becomes untenable. One cannot look at each item, system, mission, and resource in isolation; they all have relationships and interactions that impact each other and need to be accounted for.

With a multidimensional iterative model, it can be hard to intuitively determine, guess, or understand the downtime drivers. We know they are a subsystem or item that causes a technical system to be nonoperational, but truly understanding which systems and items those are, and if they change based on supply chain or utilization changes, proves tedious. SIMLOX allows the downtime driver results to be shown in both the results and reports. With the software update of 2024.1, any SIMLOX model in this and future versions will have the ability to showcase the downtime driver results.

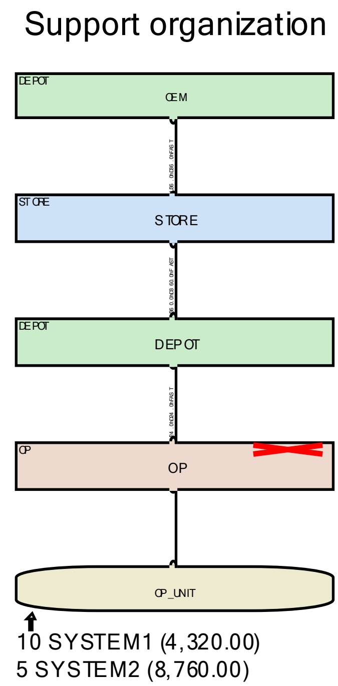
While the concept of downtime drivers is not new, the speed at which stakeholders can make decisions surrounding downtime drivers and modify optimization and simulation models to investigate the availability impacts from subsystems and items improves the collaborative and iterative decision-making process.

Model Scenario

The model for this use case is a pared-down radar system model. It contains two variants, 786 items, two depots, one store, and an operational location. There are a total of 15 systems deployed at the operational location.

- **Items:** Of the 786 items, 667 of them are universal to the radar (i.e., not variant specific). Every item is replaceable and repairable. There are 142 items that have a requirement of forced maintenance before the next mission. In addition, these items have a 90% probability for system effectiveness—meaning, 10% of the time, a failure on those items does not impact system availability or readiness—and a 90% probability of functional loss given a failure—meaning, 10% of the time there is not a functional loss when that item has a failure. There are 152 items that do not cause a functional loss and do not impact the system effectiveness. Prices for items range from $1 to $800,000, with the average being $11,727 and the mode and median being $1. Failure rates for these items have a range of 0.0008 to 1,000 failures per million operating hours. The mean is about 3 failures per million operating hours, the median is 0.3 failures per million operating hours, and the mode is 1 failure per million operating hours.
- **Maintenance:** Maintenance occurs at three levels. Level 1 is the removal and replacement of items from the system at the depot level. Level 2 is the complex removal and replacement of items at the depot level, as well as simple repairs of items at the depot level. The Level 3 is complex repairs at the OEM. More complex maintenance tasks take longer to complete.

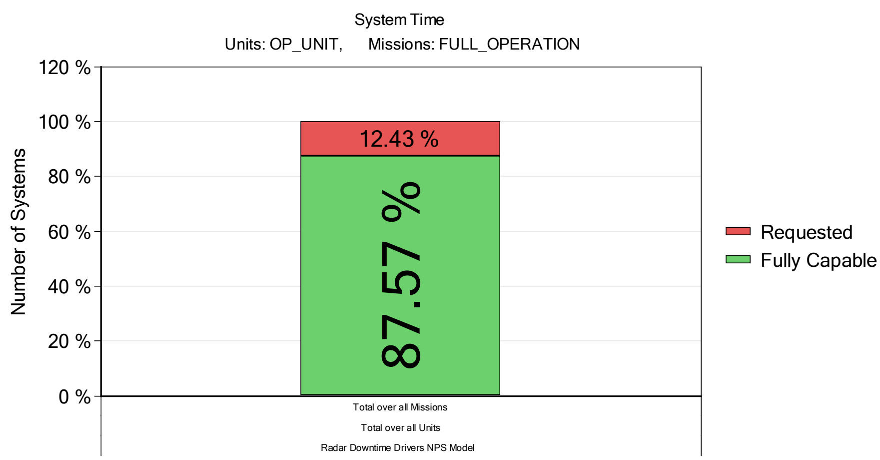
- **Support Structure:** This scenario has a basic support structure seen across many systems. There are two depots, which allow for storage and maintenance. The OEM is highest level, and it performs complex maintenance. In between the two depots, there is a store, which only allows storage of items. Items are reordered to both the OEM and the store. Below the store is the second depot, which is where intermediate maintenance occurs. Lastly, there is an operating station. At this station, maintenance and storage are not allowed. There is an operating unit assigned to this station and deployment of the systems assigned to it. Time to and from the operating station and the depot is 24 hours. Time to and from the depot and the store is 360 hours. Time to and from the store and the OEM is 36 hours.



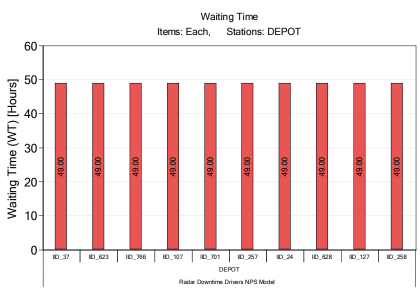
- **Missions and Deployment:** System 1, which is the variant 1, has 10 systems at the operational base. System 2 has five. Both systems are subject to a fixed 24-hour mission that requires one system. There are 64 of these missions within a year. That profile is repeated for each of the 3 years. The missions start at a randomized time and are distributed throughout the year randomly. Since the simulation is replicated 500 times, each replication has a different randomization; therefore, the overall results, which average utilization at a particular time, have factored in 500 possible randomized missions. Over the simulation period there are 192 missions that accumulate 4,608 system and mission hours.

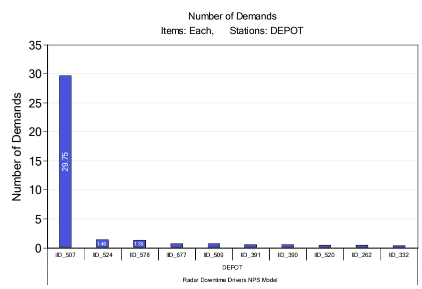
Model Results

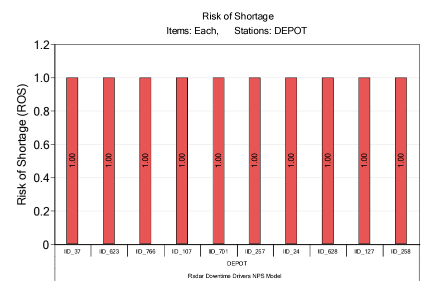
Overall, the simulation results show mission results—or how well the mission profile was carried out—to be 87.53% fully capable and 12.47% requested.



Historically, with SIMLOX results, we have looked at the item-specific information to try and identify downtime drivers. Graphs that include the number of demands, risk of shortage, and waiting time are a few that showcase what items are contributing to downtime of the system.







It is hard to tell from the item results graph exactly which items are downtime drivers and how much downtime they are causing.

Downtime Driver Results

The downtime driver results are subsystems or items-specific and are either time per a time period or accumulated time.

Table 1. System Downtime Caused by an Individual Item During an Interval of Time

	UNID	SID	OPMID	IID	STINT	ETINT	SYCDT
	Unit identifier	System identifier	Operational mode identifier	Item identifier	Start Time interval [Hours]	End Time interval [Hours]	System capability downtime [Hours]
1	OP_UNIT	SYSTEM1	<DEFAULT>	IID_507	0.00	24.00	0.0059
2	OP_UNIT	SYSTEM1	<DEFAULT>	IID_524	0.00	24.00	0.0013
3	OP_UNIT	SYSTEM1	<DEFAULT>	IID_611	0.00	24.00	0.00059
4	OP_UNIT	SYSTEM1	<DEFAULT>	IID_727	0.00	24.00	0.00089
5	OP_UNIT	SYSTEM1	<DEFAULT>	IID_305	24.00	48.00	0.00091
6	OP_UNIT	SYSTEM1	<DEFAULT>	IID_327	24.00	48.00	0.00038
7	OP_UNIT	SYSTEM1	<DEFAULT>	IID_484	24.00	48.00	0.0012
8	OP_UNIT	SYSTEM1	<DEFAULT>	IID_507	24.00	48.00	0.028
9	OP_UNIT	SYSTEM1	<DEFAULT>	IID_509	24.00	48.00	0.00049
10	OP_UNIT	SYSTEM1	<DEFAULT>	IID_524	24.00	48.00	0.0020
11	OP_UNIT	SYSTEM1	<DEFAULT>	IID_542	24.00	48.00	0.00038
12	OP_UNIT	SYSTEM1	<DEFAULT>	IID_553	24.00	48.00	0.0018

Table 2. Accumulated System Downtime Caused by an Individual Item During an Interval of Time

	UNID	SID	OPMID	IID	STINT	ETINT	SYCDT
	Unit identifier	System identifier	Operational mode identifier	Item identifier	Start Time interval [Hours]	End Time interval [Hours]	System capability downtime [Hours]
1	OP_UNIT	SYSTEM1	<DEFAULT>	IID_507	0.00	24.00	0.14
2	OP_UNIT	SYSTEM1	<DEFAULT>	IID_524	0.00	24.00	0.031
3	OP_UNIT	SYSTEM1	<DEFAULT>	IID_611	0.00	24.00	0.014
4	OP_UNIT	SYSTEM1	<DEFAULT>	IID_727	0.00	24.00	0.021
5	OP_UNIT	SYSTEM1	<DEFAULT>	IID_305	24.00	48.00	0.022
6	OP_UNIT	SYSTEM1	<DEFAULT>	IID_327	24.00	48.00	0.0091
7	OP_UNIT	SYSTEM1	<DEFAULT>	IID_484	24.00	48.00	0.030
8	OP_UNIT	SYSTEM1	<DEFAULT>	IID_507	24.00	48.00	0.82
9	OP_UNIT	SYSTEM1	<DEFAULT>	IID_509	24.00	48.00	0.012
10	OP_UNIT	SYSTEM1	<DEFAULT>	IID_524	24.00	48.00	0.079
11	OP_UNIT	SYSTEM1	<DEFAULT>	IID_542	24.00	48.00	0.0092
12	OP_UNIT	SYSTEM1	<DEFAULT>	IID_553	24.00	48.00	0.044

Note. The SYCDT adds up all time intervals prior and includes the current time interval.

Table 3. Accumulated System Downtime Caused by a Subsystem

	UNID	SID	OPMID	BDEID	STINT	ETINT	SYCDT	SYCDTA
	Unit identifier	System identifier	Operational mode identifier	Subsystem breakdown element identifier	Start Time interval [Hours]	End Time interval [Hours]	System capability downtime [Hours]	System capability downtime breakdown accumulated [Hours]
1	OP_UNIT	SYSTEM1	<DEFAULT>	RADAR_VARIANT_1	0.00	24.00	0.0087	0.0087
2	OP_UNIT	SYSTEM2	<DEFAULT>	RADAR_VARIANT_2	0.00	24.00	0.00	0.00
3	OP_UNIT	SYSTEM1	<DEFAULT>	RADAR_VARIANT_1	24.00	48.00	0.042	0.042
4	OP_UNIT	SYSTEM2	<DEFAULT>	RADAR_VARIANT_2	24.00	48.00	0.00	0.00
5	OP_UNIT	SYSTEM1	<DEFAULT>	RADAR_VARIANT_1	48.00	72.00	0.076	0.076
6	OP_UNIT	SYSTEM2	<DEFAULT>	RADAR_VARIANT_2	48.00	72.00	0.00	0.00
7	OP_UNIT	SYSTEM1	<DEFAULT>	RADAR_VARIANT_1	72.00	96.00	0.10	0.10
8	OP_UNIT	SYSTEM2	<DEFAULT>	RADAR_VARIANT_2	72.00	96.00	0.00	0.00

Note. In this case, the only subsystem in the model are the variants.

Since the only subsystems in this model are the variants, the focus for this use case is the items. Utilizing the information from these tables, you can identify what items are causing the most system downtime.

Modeling Impact

Showcasing the direct link between items and system downtime connects the dots between item decisions and their impacts. Assuming the input data for the simulation is accurate, and the maintenance tasks and support organization cannot be changed or modified, then you would be done with the downtime driver identification here. Deciding on what to do with this information will fall to decision-makers deciding if they want to invest in improving those items' failure rates or updating the support organization to improve the repair times.

If there is any uncertainty with the input data, or if you are modeling a system still in the design phase, then once the downtime drivers are identified, there needs to be a verification and inquiry as to why those might be the downtime drivers. In the example above, Item 507 causes an accumulation of over 1,400 hours of downtime. While this looks like a problem, upon further investigation, this item has a quantity of 500 in the systems, so the accumulated downtime for that item makes sense in this case. This item would then be counted as an outlier and not included in further decision-making addressing downtime drivers.

When there is room to adjust the support organization, maintenance capabilities, resource usage, or redesign items, then understanding and knowing the downtime drivers allows one to address those issues and find solutions. It is also important to see how varied usage of the support organization or resources, or maintenance capabilities can impact the downtime drivers. Identifying the overarching drivers amongst the model excursions leads one to believe that to improve the downtime for those items, an improvement on the item itself needs to be redesigned.

Quickly understanding how excursions or changes in the models impact or change the downtime drivers allows for better decision-making when supply chain issues arise or global events impact system readiness.

Data are crucial to the validity of a model or simulation; it is just as important that the results be interpreted and understood well so that decisions can be made quickly and effectively. Having explicit downtime driver results from SIMLOX reduces the risk of a poor decision being made by misinterpreted results. In addition, the ability to quickly visualize these results allows decision-makers to test possible solutions to overcome downtime and find the best option(s).

References

Systecon Group. (2018). *SIMLOX—Ensuring performance*. https://www.systecongroup.com/us/software/simlox-ensuring-performance

Systecon Group. (2020). *OPUS10—More than spare parts optimization*. https://www.systecongroup.com/us/software/opus10-more-spare-parts-optimization

Systecon North America. (2019, October 11). *US Navy common readiness model powered by Opus Suite* [Video]. YouTube. https://www.youtube.com/watch?v=hKDIJ9J_Rec

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net