The Impact of Video Game Technology on Physical Protection System Modeling & Simulation

Paul Zahnle
M. Jordan Parks
Stephen Stromberg

Global Security Analysis & Simulation, Sandia National Laboratories

April 2023

SAND2023-03388C

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

ABSTRACT

Industry advancements in computing and graphics due to the emergence of video games and related technologies have paved the way for next generation modeling and simulation (mod/sim) tools. Using these advancements, next-generation mod sim tools will continue to be more accessible due to increasingly intuitive interfaces and graphics, readily accessible required hardware, and decreased required user specialization. The result is that more users can take advantage of the benefits of using these tools to assist in the secure design and safe operation of critical facilities.

Utilizing industry advancements Sandia National Laboratories (Sandia) has developed two next-generation security tools, Scribe3D© and PathTrace© which can be used both domestically and internationally at no cost for approved partners.

The use of next-generation mod/sim tools such as Scribe3D or PathTrace can provide many benefits to facility security personnel, operators, and regulators. Mod/sim can be used as a cost-effective way to evaluate security-by-design and the Design and Evaluation Process Outline (DEPO), evaluate the effectiveness of current physical protection systems (PPS), plan for force-on-force (FoF) exercises, conduct tabletop exercises with results recorded for future trainings, model an unlimited set of “what if” scenarios, conduct virtual tours, analyze adversary attack plans, study adversary attack paths, find the critical detection point (CDP) in an adversary pathway, and simulate combat engagements with scientific and verified data.

This paper is not intended to be a study of all applications and variations of modern or historical mod/sim tools. It will concentrate on the advent of computer modeling for use in combat simulations and vulnerability analysis of critical facilities.

CONTENTS

LIST OF FIGURES

1. MODELING AND SIMULATION BEGINNINGS

1.1. Mod/Sim Definition

"Modeling and Simulation is the use of a physical or logical representation of a given system to generate data and help determine decisions or make predictions about the system. (Modeling and Simulation) is widely used in the social and physical sciences, engineering, manufacturing, and product development, among many other areas."[1]

1.2. Physical Systems Modeling

In 1941, Director of the U.S. Army Corps of Engineers Eugene Reybold built the Mississippi River Basin Model, a large-scale hydraulic model that represented an area of 200 acres at a 1:100 vertical and 1:2000 horizontal scale. Considered one of the first models of a critical infrastructure system, this model enabled engineers to simulate weather conditions needed to evaluate the effect of flood control measures on the entire system, rather than on one element or area at a time.

"The first real success of the project came in April 1952, when the model was used to predict that year's flood. Based on the results from the test of the model, the government was advised where levees needed to be raised and which areas needed to be evacuated. That year the Mississippi River Basin Model prevented an estimated $65 million in damages. For the next two decades until computer modelling software became available, the Mississippi River Basin Model was used extensively to determine flood control strategies from Montana to Louisiana. The model was last used in 1973 to test the effects of opening the Morganza Spillway."[2]

1.3. Computer-Based Modeling

Corresponding to the end-of-life of physical models such as the Mississippi River Basin Model in 1973, in the early 1970s there existed a complex web of computer technologies available for writing advanced software to simulate effects on physical systems. Honeywell had a computer system with 9-Bit Bytes. Cray-designed CDC computers utilized 60 Bit words. The CDC 7600 was the Seymour Cray-designed successor to the CDC 6600. Significant scientific computing in the early 1970's was done on these CDC computers and was written in the Fortran computer language. If you wanted to run advanced mod/sim codes, you had to buy time on one of the two CDC computers available at the time. The Weapons Lab at Kirtland AFB had one, as well as another that was located at a U.S. government facility in Northern Virginia. These 6600's were very expensive to operate and cost hundreds of thousands of dollars to purchase. Between 1965 and 1978, IBM 360 and 365 computers were also starting to be used for scientific computing, but they also had

---

[1] TechTarget Contributor. (September 2017). modeling and simulation (M&S). TechTarget. https://www.techtarget.com/whatis/definition/modeling-and-simulation-MS.

[2] Kaushik Patowary. (September 2015). The Mississippi River Basin. https://www.amusingplanet.com/2015/09/the-mississippi-river-basin-model.html.

bit/byte limitations. These early IBM computers had a limitation of 65,000 words in an array. These were just some of the major hardware limitations for the early coders of advanced Mod/Sim programs.

Many of these early simulation codes were based on answering the questions surrounding the possible use of tactical nuclear weapons in Europe. The existence of the Warsaw Pact and the threat of nuclear warfare in Europe helped drive ongoing U.S. investments in computer-based modeling and simulation development. In the 1970s and 1980s. Other countries let the U.S. lead the very expensive effort of creating Mod/Sim codes. NATO used a Mod/Sim program called COMO in Brussels in an early example of a U.S. developed simulation being used in Europe to study the threat of nuclear warfare. These early simulation interfaces were all text-based for data input and results output.

More recently, the concept of Digital Twins (DTs) has been used to visualize critical infrastructure facilities such as nuclear power plants and manufacturing facilities. These digital twins can now be created in advanced 3D modeling programs, imported into Mod/Sim tools, and can serve multiple purposes. Threat modeling, Probability Risk Analysis (PRA), critical systems modeling, and Security by Design (SeD) methodologies can now all be supported in single or multiple related models. As proven with the very first Mississippi River Basin Model in 1941, modeling complex systems in a simulated environment is an accurate and cost-effective way to study complex system interactions and better prepare for future events.

1.4. U.S. Government use of Mod Sim

In the late 1980s, computer simulations of FoF exercises were introduced. The U.S. Air Force began using Lawrence Livermore National Laboratory's Security Exercise Evaluation System (SEES) application. SEES was a variant of the Urban Combat Computer Assisted Training System (UCCATS) – Army Simulation. In the 1990s, new tools emerged and were combined. The Department of Energy and National Nuclear Security Agency (DOE/NNSA) began to use SEES for Physical Protection System (PPS) analysis to augment FoF analysis. SEES/UCCATS were merged into a new Semi Automated Forces (SAF) called the Joint Tactical Simulation (JTS) in 1994. DOE policy by 1996 included JTS as one of the approved Vulnerability Analysis (VA) tools, basing the results on the Air Force's verification, validation, and accreditation project of JTS. During the next development phase of JTS, it was revised into the JCATS application in 1997. Since 1997, JCATS has been the approved/accepted means of PPS Security throughout DOE NNSA, DOD, NRC, NATO, and DOS critical infrastructure. JCATS is used by 3 national laboratories, multiple DoD sites worldwide, NATO, and 30 foreign allies.

1.5. Video Games and Game Engines

Over the past 50 years video games have evolved from simple two-dimensional games like pong to three-dimensional games with photorealistic surroundings and massive complexity and scale. As video games have advanced, so have the game engines that power them. "A game engine is a software framework primarily designed for the development of video games and generally includes relevant libraries and support programs."[3] Many of these game engines were created by a specific company to support the development of their own games or series. However, a relatively recent change has occurred in the gaming industry in which companies are developing, releasing, and maintaining game engines as products to be used by the public. This has substantially lowered the barrier to entry for developers to tap into the advancements in computer graphics achieved by the gaming industry. Two modern mainstream examples of these game engines include Unity and Unreal Engine. Descriptions of the two engines are given below:

"Unity is a cross-platform game engine developed by Unity Technologies, first announced and released in June 2005 at Apple Worldwide Developers Conference as a Mac OS X game engine. The engine has since been gradually extended to support a variety of desktop, mobile, console and virtual reality platforms. It is particularly popular for iOS and Android mobile game development, is considered easy to use for beginner developers, and is popular for indie game development."[4]

"Unreal Engine (UE) is a 3D computer graphics game engine developed by Epic Games, first showcased in the 1998 first-person shooter game Unreal. Initially developed for PC first-person shooters, it has since been used in a variety of genres of games and has seen adoption by other industries, most notably the film and television industry."[5] It cannot be understated how UE has literally re-defined 3D development across many different industries. An example is the Disney series called The Mandalorian, which uses Unreal engine on its massive screen-stages to render backdrops in real-time.

---

[3] Valencia-Garcia, Rafael, et all. (2016, November 23). Technologies and Innovation: Second International Conference. ISBN 9783319480244. Retrieved 2021-07-22. https://en.wikipedia.org/wiki/Game_engine#cite_note-1

[4] Wikipedia contributors. (Retrieved April 2023), Unity (game engine). Wikipedia.
Dealessandri, Marie (January 16, 2020). "What is the best game engine: is Unity right for you?". GamesIndustry.biz. Gamer Network. https://www.gamesindustry.biz/what-is-the-best-game-engine-is-unity-the-right-game-engine-for-you

[5] (Wikipedia, 2023) Wikipedia contributors. (Retrieved April 2023), Unreal engine. Wikipedia. https://en.wikipedia.org/wiki/Unreal_Engine

2. THE CONVERGENCE: MOD/SIM AND VIDEO GAME TECHNOLOGY

Some of the first programs that were used for conducting vulnerability analysis on critical infrastructure facilities, such as the U.S. government-funded programs ASSESS, ATLAS, and MP VEASI, were based on Microsoft Excel spreadsheets or simplified GUI programming that did not render a full scene in 3D. Although these programs used complex and previously vetted algorithms to produce accurate results, they still required specialized training and knowledge to produce results when applied to a vulnerability analysis study of critical infrastructure.

In the 1990s, private corporations and some government entities began actively developing mod/sim programs that tapped into the advanced capabilities of 3D rendering technologies. However, at the time, the availability of readily acquirable game engines was limited, so many of these programs involved the development of bespoke game engines leveraging low level graphics libraries, such as OpenGL. This placed the burden of maintaining both the engine and the applications on the shoulder of the developers.

Today, with the mainstream release of game engines, mod/sim tools can be developed using technologies that are continuously improved as part of the gaming industry, and do not have to be maintained by the developer. Two examples of mod/sim tools that leverage commercial gaming technology are Scribe3D and PathTrace, which were created by Sandia National Laboratories using the Unity game engine. They were created using industry technologies for visualization and other modern game engine advancements, while staying grounded in scientifically backed analytics. These tools are designed to be simple and easy to use for an average-level computer user. These tools were created to provide a basic, accessible, and free analytical capability to international and domestic partners and agencies where accessibility inhibits the possible utilization of mod/sim tools. Below, a more in-depth description of these tools is presented in order to showcase examples of modern tools leveraging game technologies.

2.1. PathTrace: Path Analysis Leveraging Industry

Created using the Unity game engine, PathTrace provides an intuitive 2D software solution for a user to perform an analysis of adversary pathways on any facility. Using PathTrace, pathway analysis can be completed for any facility in three stages. It serves as the next step in making path analysis more user friendly, visually compelling, and accurate.

Figure 1. PathTrace user interface

Previous generations of tools such as Systematic Analysis of Vulnerability to Intrusion (SAVI) and Multipath Very-Simplified Estimate of Adversary Sequence Interruption (MPVEASI) used simplified adversary sequence diagrams which produced abstract, mathematical models of facilities. PathTrace is based on the same algorithms of these predecessor tools, but features a 2D site layout, which makes building the model, presenting results, and training on the tool much more accessible without sacrificing analytical validity.

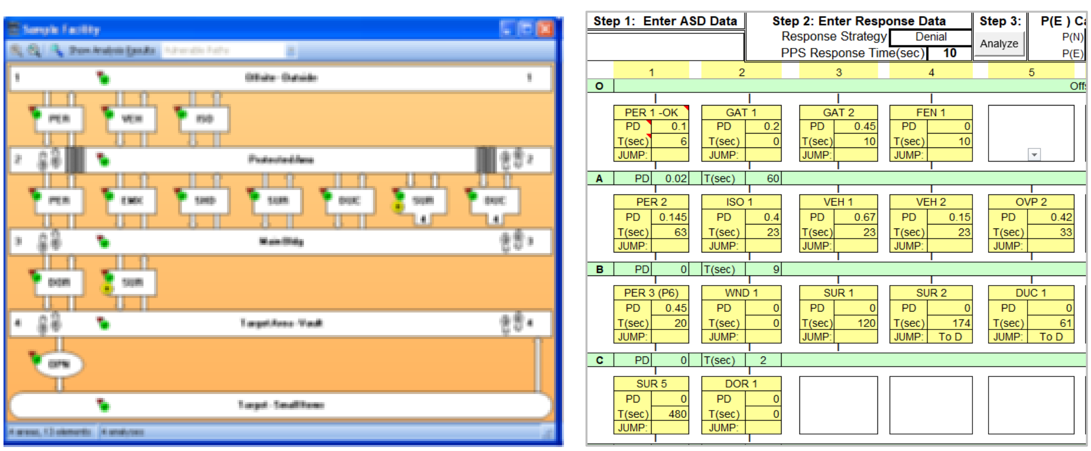

Figure 2: SAVI user interface (left); MP VEASI user interface (right)

2.1.1. Model Building

PathTrace allows the user to represent a facility utilizing only an aerial image taken from a diagram, blueprint, floorplan, or even Google Maps imagery. Once the image is identified, the user can then draw delay barriers such as walls, doors, fences, gates, and windows that would stop the progress of the attacker. The user can then define areas of detection such as sensor volumes, patrolled zones, and camera coverage that would result in the detection of the attacker. At this stage, the user is able to specify and credit all of the physical protection system elements that would contribute to both the delay and detection of an attacker.

2.1.2. Response vs. Adversary

After defining the model, the use then enters information to define response timelines and adversary capabilities. It is in this stage where the user will be able to specify the response team's reaction to a detection event at the facility. Specifically, after an adversary is detected, the user will specify a response timeline from the moment the sensor alarms to response force arrival and interruption. The adversary capabilities involve the tools that the adversaries will use in their attack. The user is able to consider different kinds of adversary teams with different toolkits and can represent how those adversaries will interact with that facility.

2.1.3. Pathway Analysis

The final stage handles the analysis of a facility. PathTrace takes the representation of the facility, the reaction of the responding forces, and the toolkit of the adversary and generates possible pathways of an adversary attack. At this stage, the user can quickly generate the quickest, stealthiest, and most vulnerable pathway to a target or target set. Each pathway is visually represented and reports the probability that a response team will be able to interdict or halt the actions of the adversary.

Users can consider any number of facility upgrades and "what if" scenarios quickly and measure effects on adversary pathways and subsequent Probabilities of Interruption (Pi). The outputs of PathTrace can be used to verify the effectiveness of an existing design and justify upgrades before they are implemented. PathTrace is defined by its strength in producing cost effective actionable feedback and quick investigation on the feasibility of a facility design or upgrade.

Figure 3. Notional PathTrace model (left); Adversary pathway (right)

2.2. Scribe3D: Tabletops Leveraging Industry

Scribe3D has many advantages over conventional tabletop exercise methods. Next Generation tabletop tools such as Scribe3D allow operators to quickly and inexpensively model and record adversary and protective force actions within a physical protection system. Operators can study an unlimited number of "what if" scenarios, adjust and analyze variables in detection, delay, and response to facilitate a greater understanding of facility security.

Scribe3D uses science-based data to compute Probability of Hit/Probability of Kill (P_H/P_K) and Probability of Neutralization (P_N) for analysis. Operators can quickly identify scenarios of concern and test mitigation strategies. You can then save and record results for further presentation and training.

Additionally, operators can:

- Create table-top exercises for attack, theft, sabotage, or natural disasters.
- Train operational staff and conduct virtual tours.
- Use the Terrain Creation features to download real-world terrain data and create new Scribe3D models on the fly.
- Run Monte Carlo simulations to analyze large datasets.

Figure 4. Scribe3D user interface (left); Hypothetical models (right)

2.2.1. Tabletop Analysis

Scribe3D operates in the following modes:

- **Standard:** Create, save, and modify tabletop scenarios.
- **Battleboards:** Quick and simple analysis without a timeline.
- **Multiplayer:** Conduct tabletop exercises where all perceptions are controlled by the application and actions are approved/denied by a moderator.

- **Analysis:** Run hundreds or thousands of simulations in a Monte Carlo algorithm to study large datasets to compare different scenario variables.
- **Terrain Generator:** Create terrain from real-world data for your facility.

With Scribe3D in Analysis Mode you can run hundreds or thousands of simulations in a Monti Carlo algorithm to study large datasets of different scenario variables. You can create and test different “what if” scenarios to modify and compare variables in detection, delay, and response values.

You can also create table-top exercises for attack, theft, sabotage, or natural disasters and create realistic scenarios using more than 100 additional props (objects), including office equipment, sensors, barriers, and natural elements such as trees and rocks.

Modeling a critical infrastructure facility such as a nuclear power plant or research reactor in a Modeling and Simulation (Mod/Sim) tool such as Scribe3D or PathTrace can provide many benefits to facility security personnel, operators, and regulators. Mod/sim can be used as a cost-effective way to evaluate facilities in a Security-by-Design and a Design and Evaluation Process Outline to facilitate better and more cost-effective facility security anywhere in the world.

3. CONCLUSION

As advancements in visualization have progressed based on the rise of video games and related technologies so too have mod/sim tools benefitted. These benefits have manifested in many ways including the advancement of graphical capabilities and reducing the complexity of development and operations of the tools. Two examples of these tools, built on leveraging industry gaming technologies are PathTrace and Scribe3D. These tools showcase these benefits by being built on top of commercial game engines in order to achieve user interfaces similar in quality to modern video games. In turn these applications have increased accessibility enabling non-technical users with physical security backgrounds to conduct their own analysis. Looking towards the future, the continued integration of physical security concepts with game industry technologies will continue to push the envelope of sophisticated mod/sim tools.