

Repeatable Modeling & Simulation



Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Copyright 2023 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

GOVERNMENT PURPOSE RIGHTS – Technical Data

Contract No.: FA8702-15-D-0002

Contractor Name: Carnegie Mellon University

Contractor Address: 4500 Fifth Avenue, Pittsburgh, PA 15213

The Government's rights to use, modify, reproduce, release, perform, display, or disclose these technical data are restricted by paragraph (b)(2) of the Rights in Technical Data—Noncommercial Items clause contained in the above identified contract. Any reproduction of technical data or portions thereof marked with this legend must also reproduce the markings.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM23-0143

Agenda

- Introduction
- History of our work
- How our tools work today
- Tool use cases
- Questions and prompts

Introduction

How do we make **mission-specific modeling** and **simulation** capabilities more **repeatable**?

- **Repeatability:** I want to be able to re-use rehearsal and/or learning materials in a way that is useful...
- **Mission-specific:** ... to my particular mission...
- **Modeling and simulation** capabilities: ... utilizing different tools and techniques.

Why Repeatability?

- Repetition is the key to successful learning
- Holds true for all varieties of rehearsals
 - Individual vs team
 - Assessment vs instruction
 - CPT vs CMT vs NMT
- More confidence in results

History of Repeatability in our Software

Simulation, Training and Exercise Platform (STEP)



- Monolithic (vs modular): Everything* in one system.
- Making updates to a scenario was inefficient
- Updating VMs takes time. Requires scripts to configure.

Improvements To Repeatability

Crucible et al.

- **Modular** (vs monolithic): Every tool has a discrete job.
- Improved scenario **portability** and editing
- Developed in **open-source** using **open standards**



TopoMojo

Caster: Topology Design & Deployment

- Utilizes Terraform to design and build topologies using infrastructure-as-code
- Allows for variety of platform providers (vsphere, proxmox, cloud, etc) for portability
- Code can be shared/copied/edited at will.
- Compare to previous model using VMs and baked-in configurations.



Steamfitter: Scenario-Event Design & Execution

- Scenario orchestration with tasks that execute on deployed machines
- Specific tasks can be copied to other events and machines and/or updated for subsequent iterations
- Utilizes open-source StackStorm automation engine to execute automatic workflows



Moodle

- Open-source learning management system
- Allows for sharing of curriculums and learning records
- Increased interoperability between platforms
- xAPI-capable learning records store for capture of student data
 - Uses “actor verb object” statements to track actions



- ASP.NET Core application for creating training content on the fly.
- “Topo” as in topology. “Mojo” as in the magic of deploying and accessing on demand.
- Allows content creators and developers to quickly create and deploy virtual environments to test software

Suggested Discussion Prompts

- How can I make assessments repeatable while maintaining integrity of the process?
- What are the components of a scenario to which I want to introduce variability?
- How much variability do I introduce into team, squad, and individual rehearsals?
- What are the types of data points I want to capture in my assessments/rehearsals?

