

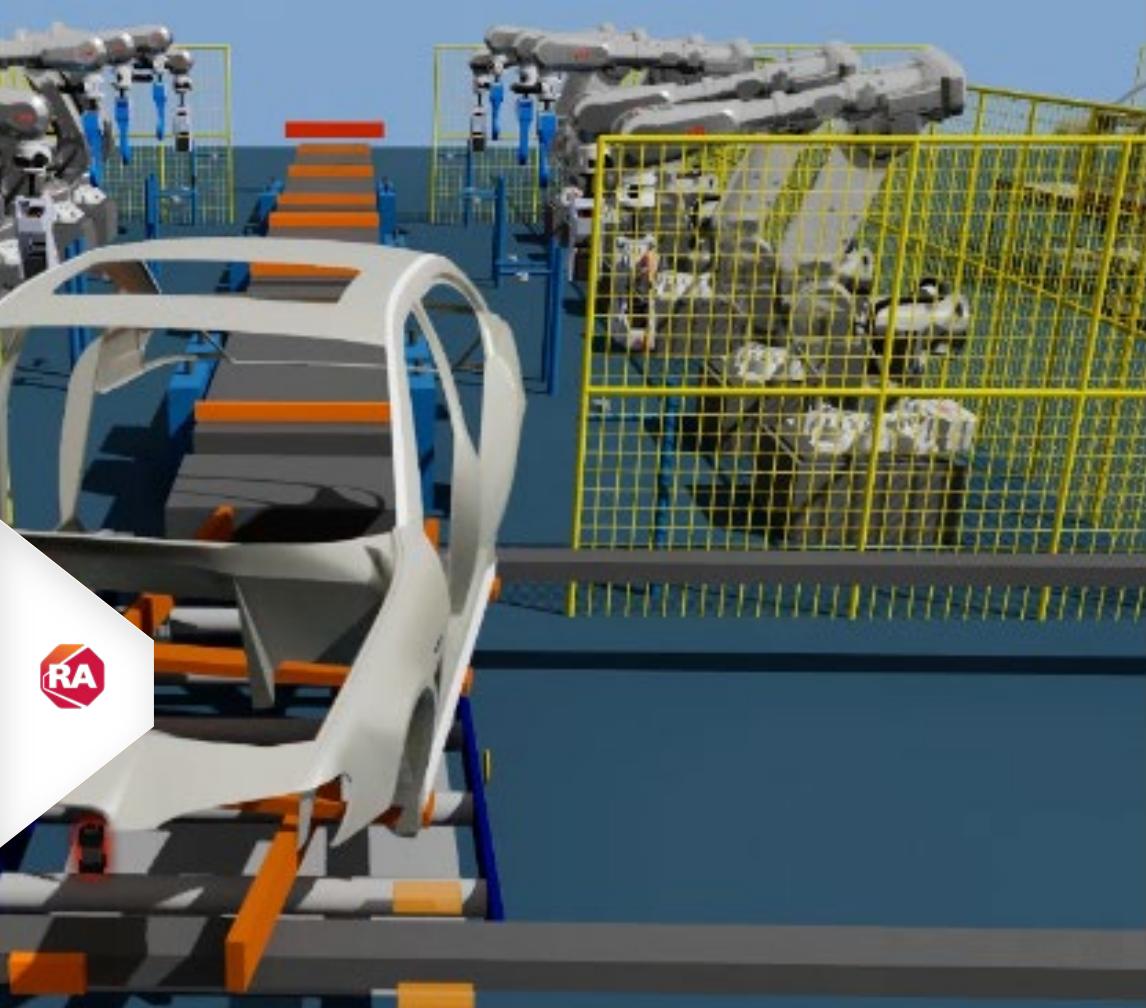
DIGITAL ENGINEERING SUMMIT

16-17 June 2025

Rome Marriott Park Hotel, Italy



Creating the Future of
INDUSTRIAL OPERATIONS



Test Runner

John Pritchard, Emulate3D Business Manager

Andrew Deeble, Emulate3D Product Manager

Emulate3D TestRunner Full Test Suite C:\Users\ADeeble\OneDrive - Rockwell Automation, Inc\Emulate3D\Projects\Emulate3D 2025 Feature Videos\Test Runner for Emulation

Zone 070

NEW TEST SCENARIO RUN ALL STOP

Name	Description	Configuration	Result
070PinStop1Retract	PinStop1 on 070 Fails to Extend	Action	Running
070PinStop2Retract	PinStop2 on 070 Fails to Extend	Action	Passed
070PinStop1Feedback	No AtExtended Feedback from PinStop1	Action	Passed
070PinStop2Feedback	No AtExtended Feedback from PinStop2	Action	Passed
D60to070CamLiftTableFailure	Incoming lift table fails to raise	Action	Passed
D70to080CamLiftTableFailure	Outgoing lift table fails to raise	Action	Passed
PX1Disconnect	No signal from PX1 prox	Action	Passed
PX2Disconnect	No signal from PX2 Prox	Action	Passed
PX3Disconnect	No signal from PX3	Action	Passed

Test Runner Aspect Viewer: PX1

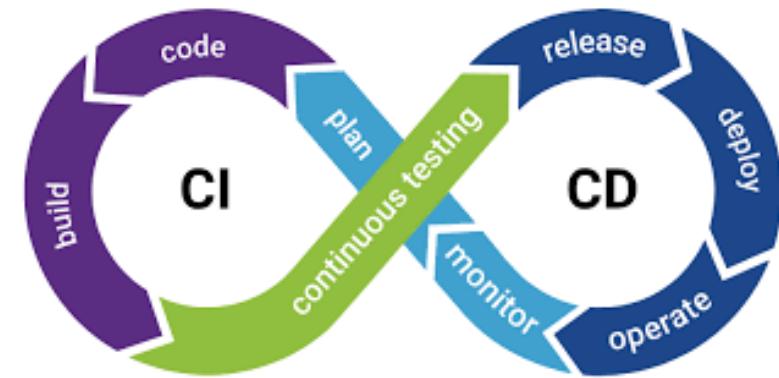
| What is Test Runner?

Built on top of NUnit Testing Framework

User Interface which generates NUnit Tests
...or create your own scripted tests

Application runs externally, launching
Emulate3D from the command line
...or use Test Runner inside Emulate3D

Enables powerful CI/CD workflows!



Why use Test Runner

Single test framework for Simulation, Controls Testing, and Catalog Development

Simulation

Compare models and layouts
against different scenarios

Metaheuristic Algorithms for
model optimization

Compatibility with existing
data collection methods

Experiments on steroids!

Emulation

Create and automatically run
emulation regression tests

Run test scenarios with each
controls code change

Perform fault simulation with
fault framework

CT Regression Test Dashboard!

Catalog Development

Create and automatically run
script and catalog unit tests

Run test scenarios for each
commit... more on that later!

Make full use of NUnit through
scripting tests

Facilitates CI/CD workflows!

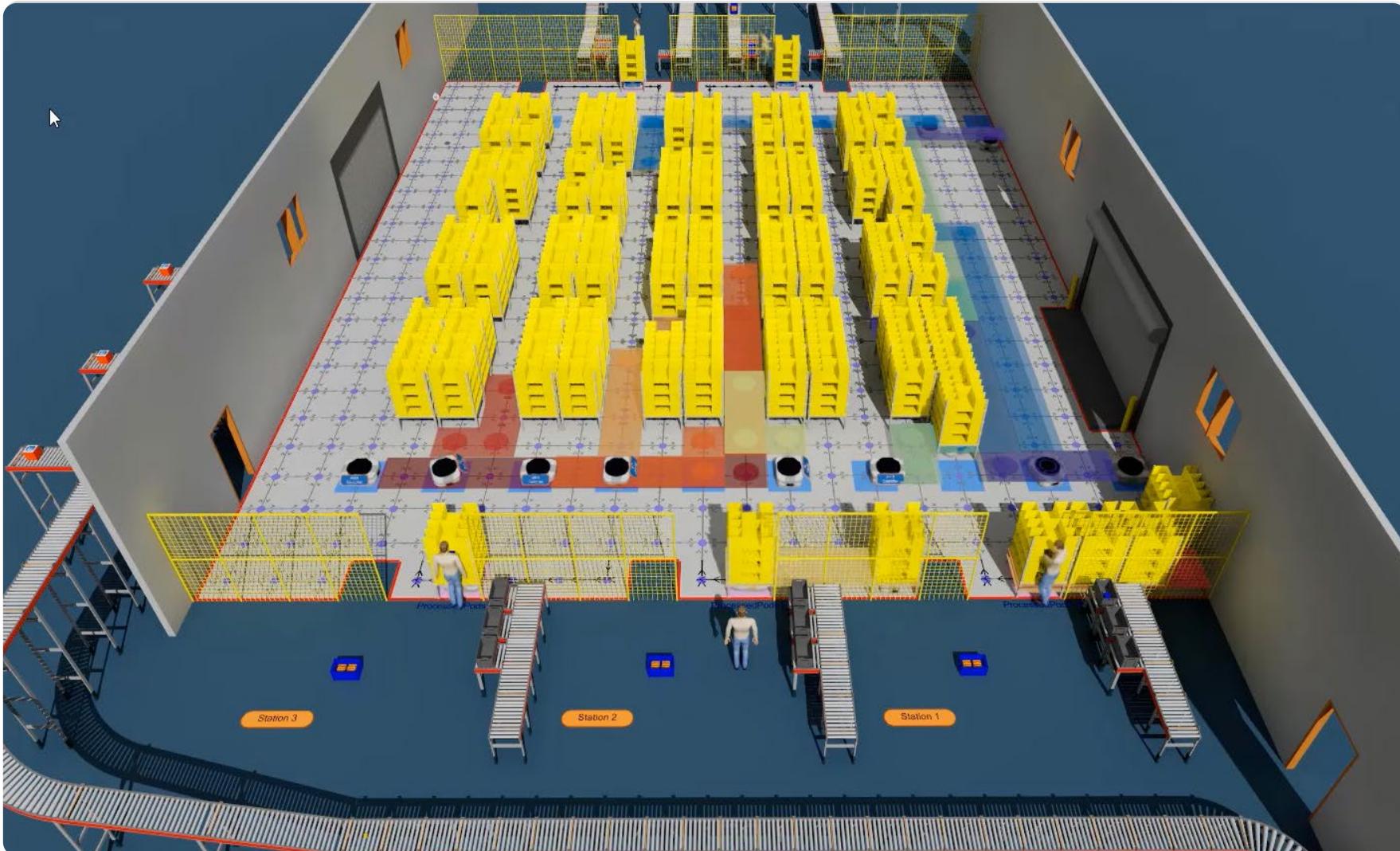


Example - Test Runner for Simulation

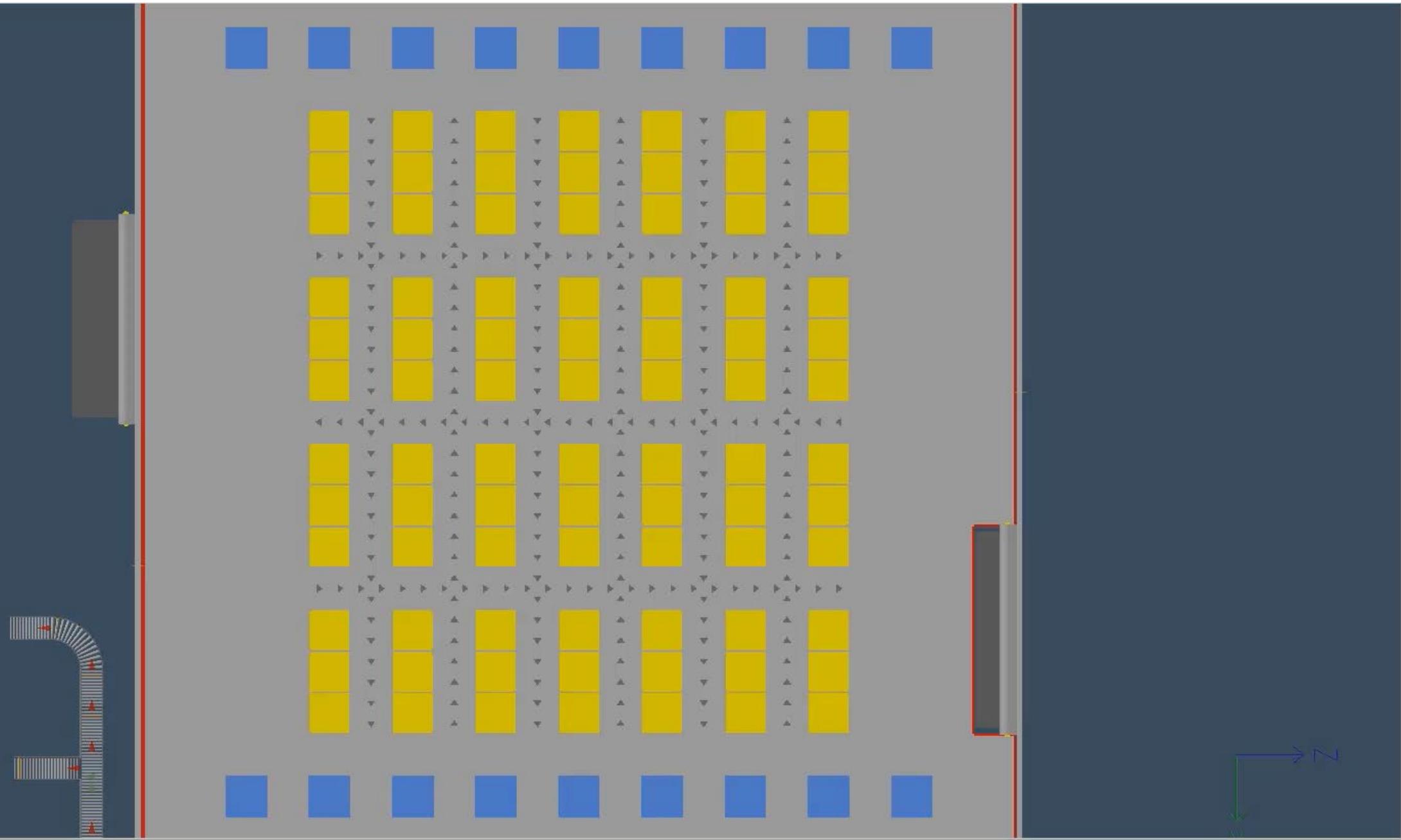
Test Runner as a successor to the Experiments feature

Use Test Runner to:

- Compare designs
such as different pod and aisle arrangements
- Run "What If" scenarios,
such as a Picking Station going down
- Vary model parameters
to optimize a system,
such as AMR fleet size



Test Scenarios – Comparing Model Variants





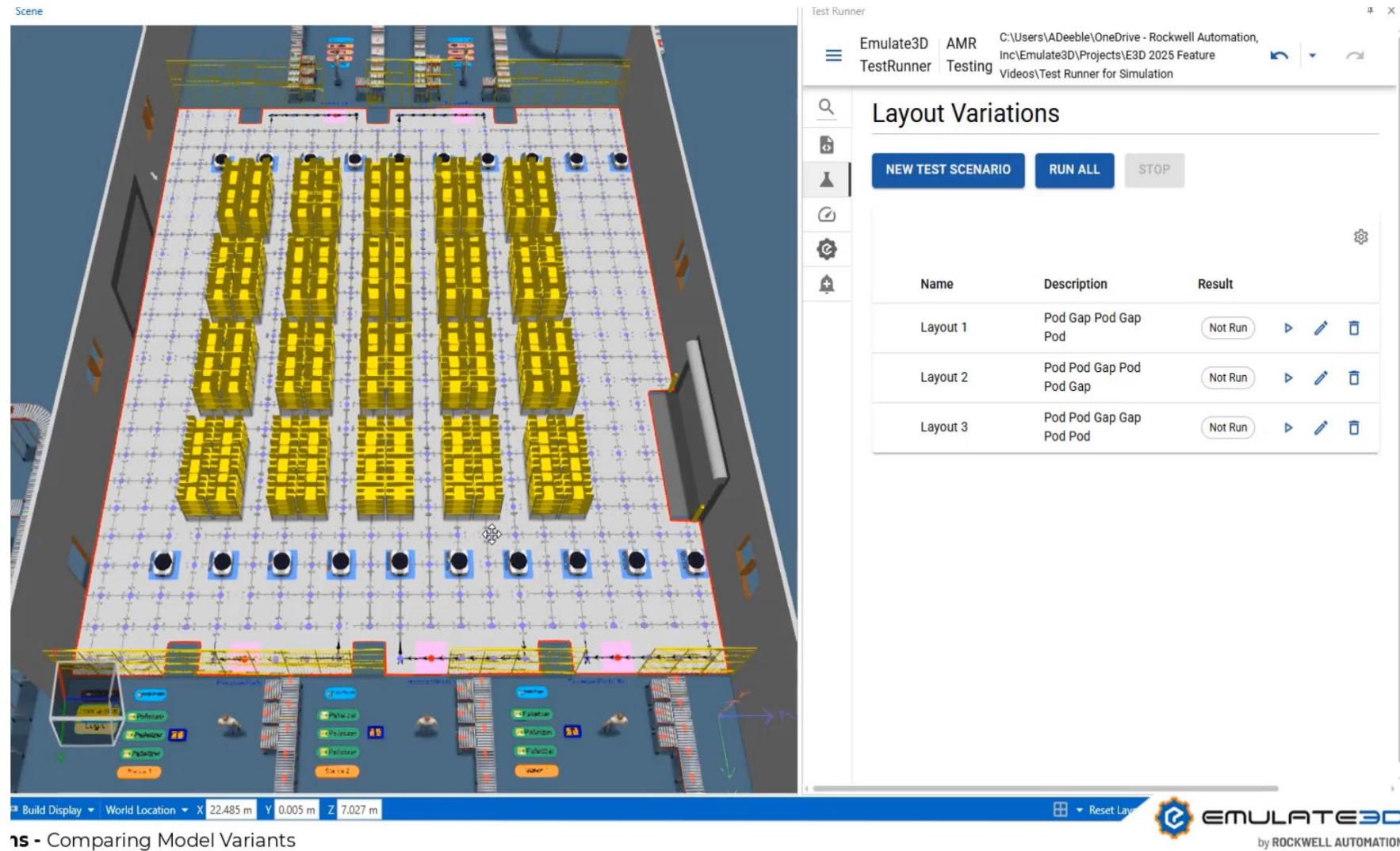
Creating Scenarios

Run a Model against Actions, and report on the results

Test Runner can be opened inside E3D, or as a separate application.

A Test Project can contain multiple models. These can be run against different scenarios.

Full data is collected in results folders. Summary KPIs can be seen inside Test Runner





What's Planned - Replications and Reporting

Test Runner is being updated in each minor release, with a [roadmap here](#).

Easily create replications for each scenario.

Run a test multiple times with different random number generation.

Confidence intervals presented for KPIs.

Show the pinned KPI in the dashboard view.

Improved summary data outputted as csv.

Filtering and sorting of test results.

The screenshot shows the 'Test Runner' application window. The title bar indicates the project is 'Emulate3D TestRunner' and the replication path is 'Replications C:\Users\RIzhar\Downloads'. The main area is titled 'Conveyor Throughput Tests'. It features a sidebar with icons for search, file, test, settings, and notifications. Below the title is a control bar with '+ NEW TEST SCENARIO' and 'STOP' buttons. A summary bar shows 4 successful (green checkmark), 0 failed (red X), and 1 warning (blue exclamation mark) scenarios. The main table lists test results for 'Scenario 1' across five seeds. Each row shows the seed name, result status (Passed), throughput value, and confidence interval. The last row for 'Seed 4' shows 'Running'.

Name	Result	Action
Scenario 1	4 Passed	Throughput: 215.75 ± 1.88
Seed 0	Passed	Throughput: 215
Seed 1	Passed	Throughput: 218
Seed 2	Passed	Throughput: 217
Seed 3	Passed	Throughput: 213
Seed 4	Running	



Creating Actions

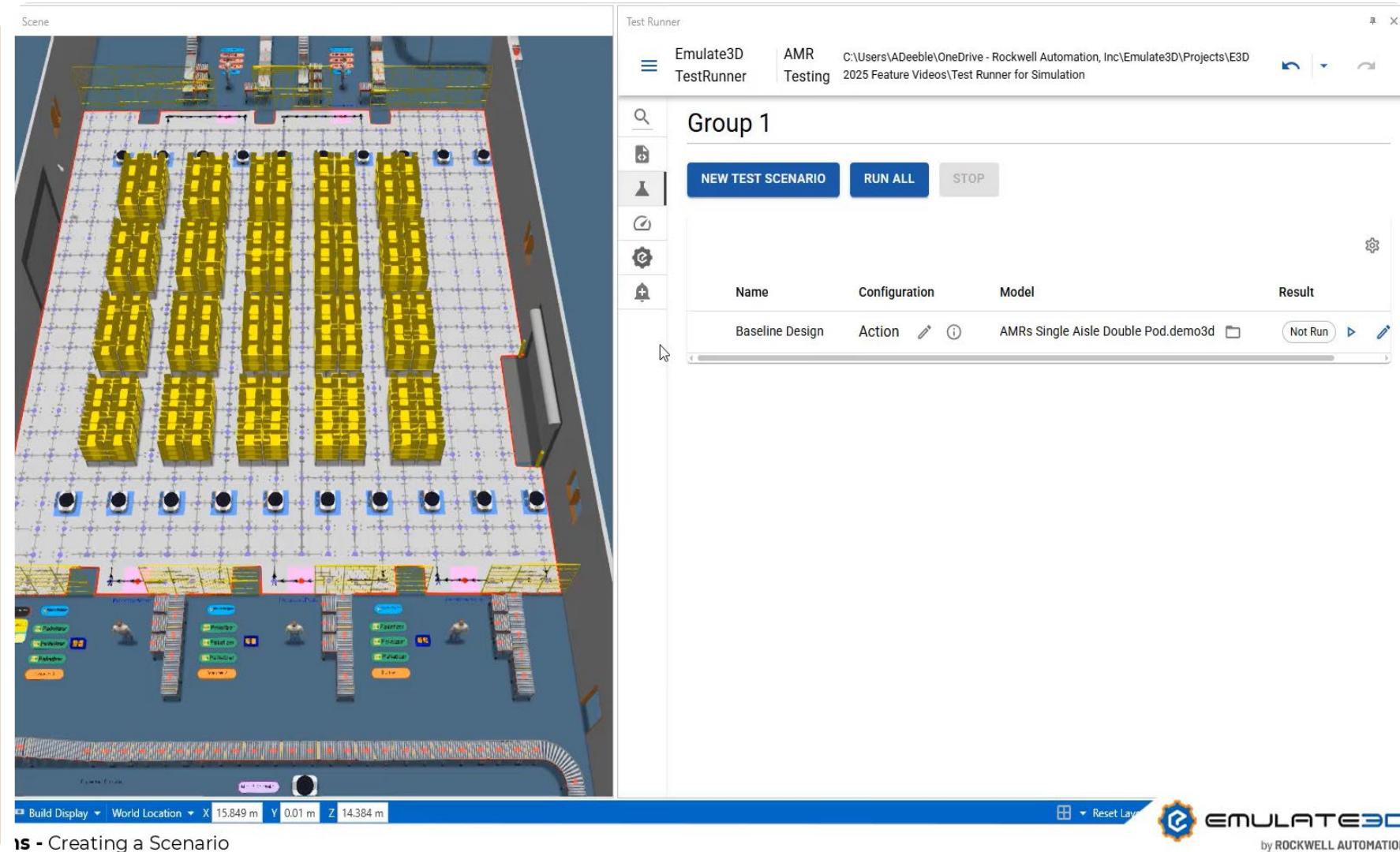
Create complex and flexible “What If” scenarios

Action schedules can be applied to a Scenario:

- Property setters change any property, aspect, or enable Fauts.

- Property recorders capture headline KPIs, alongside data collectors

- Assertions check model behaviour, and pass/fail.





Tips and Tricks – Generating Many Tests

Test Runner is being updated in each minor release, with a [roadmap here](#).

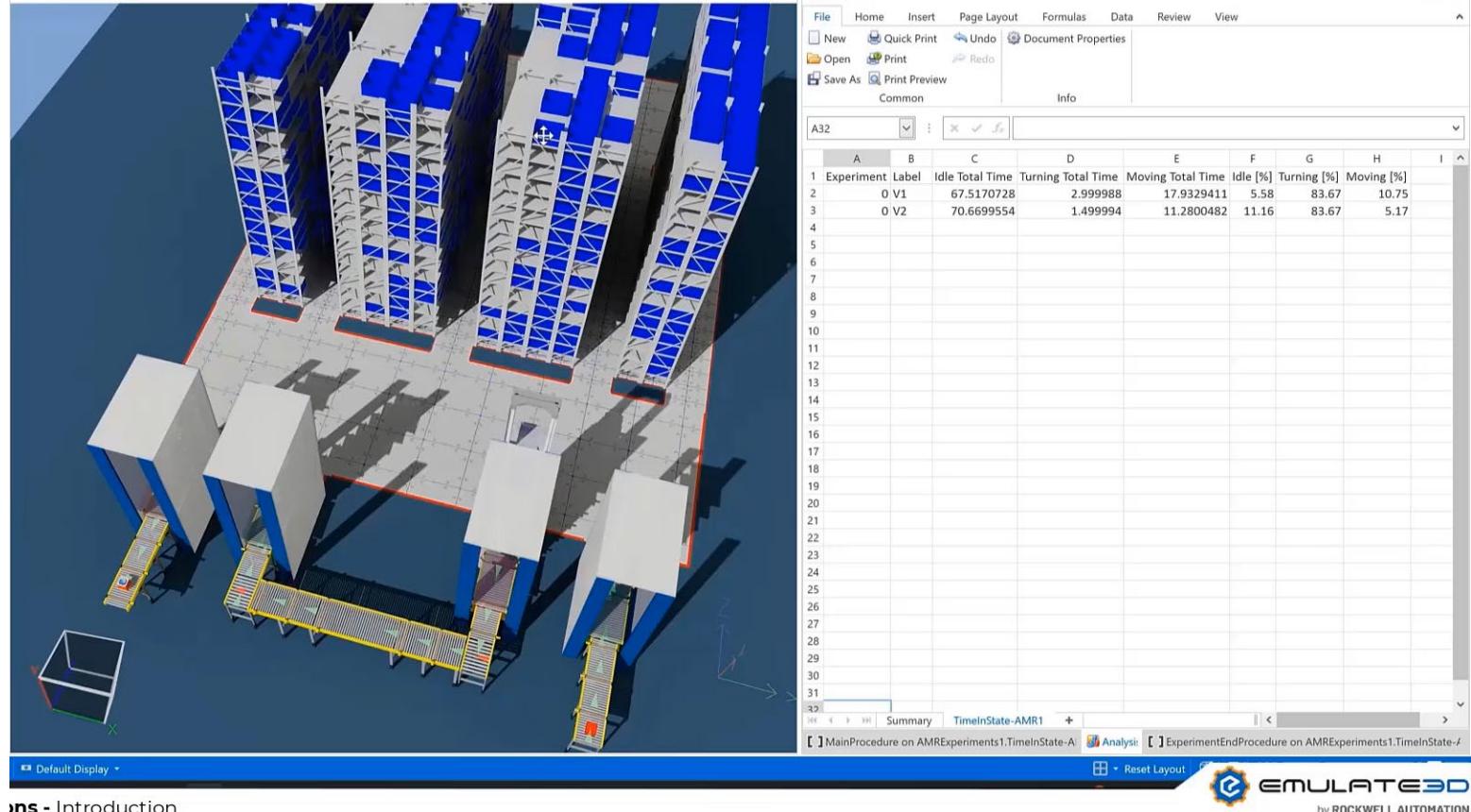
For example, sizing a fleet of AMRs. How can we run a test for 20, 21, 22, 23..... 39, 40 AMRs?

Note that the AMR framework has been updated for improved experimentation!

Use the Optimizers. The Brute Force optimizer is ideal for ranges of tests!

Use “Based On” to make one set of Actions inherit from other Actions.
Coming soon - actions triggered by events / delays from preceding actions

Customise Actions – all editable JSON
Coming soon - import actions from CSV



Model Optimization

Tools to automatically vary parameters to maximize a chosen goal

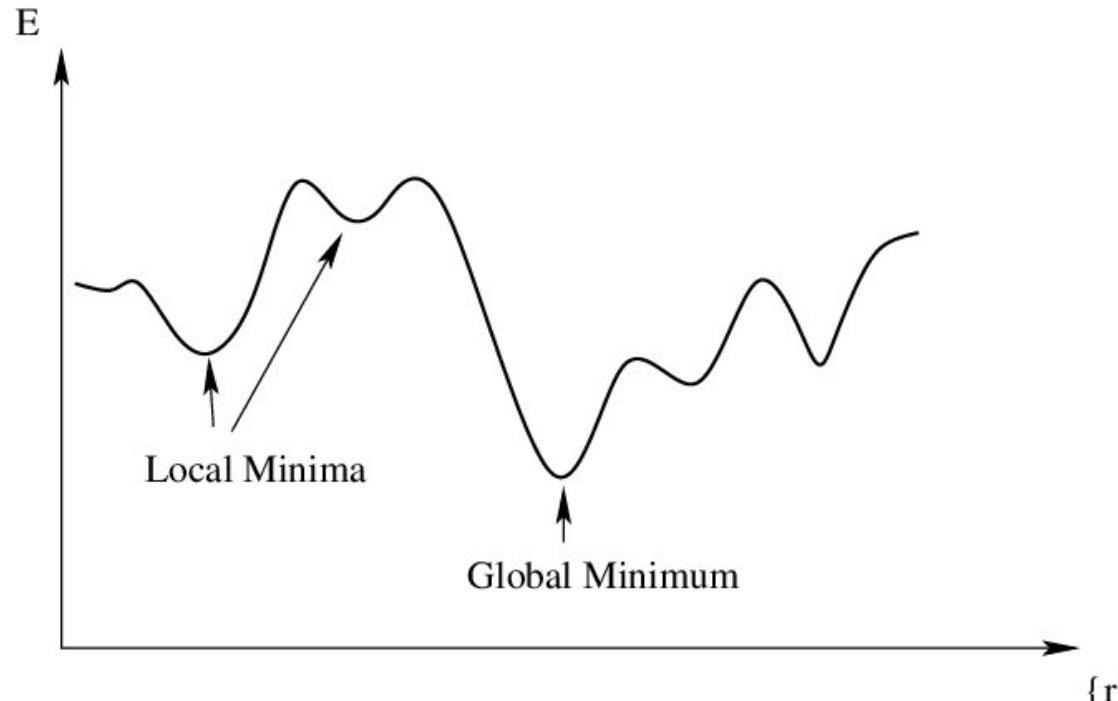
Find combination of parameters that maximizes some goal(s).

Parameters are user-specified:

- Number of vehicles, size of buffers or storage...
- Any numeric property (continuous or discrete)

Goals are user-specified

- Throughput, Utilization, Operating cost
- Any function that returns a numeric value



The gradient based methods you learned at school are not enough...

Metaheuristics are algorithmic frameworks that help find solutions to complex optimization problems that are difficult to solve with traditional methods

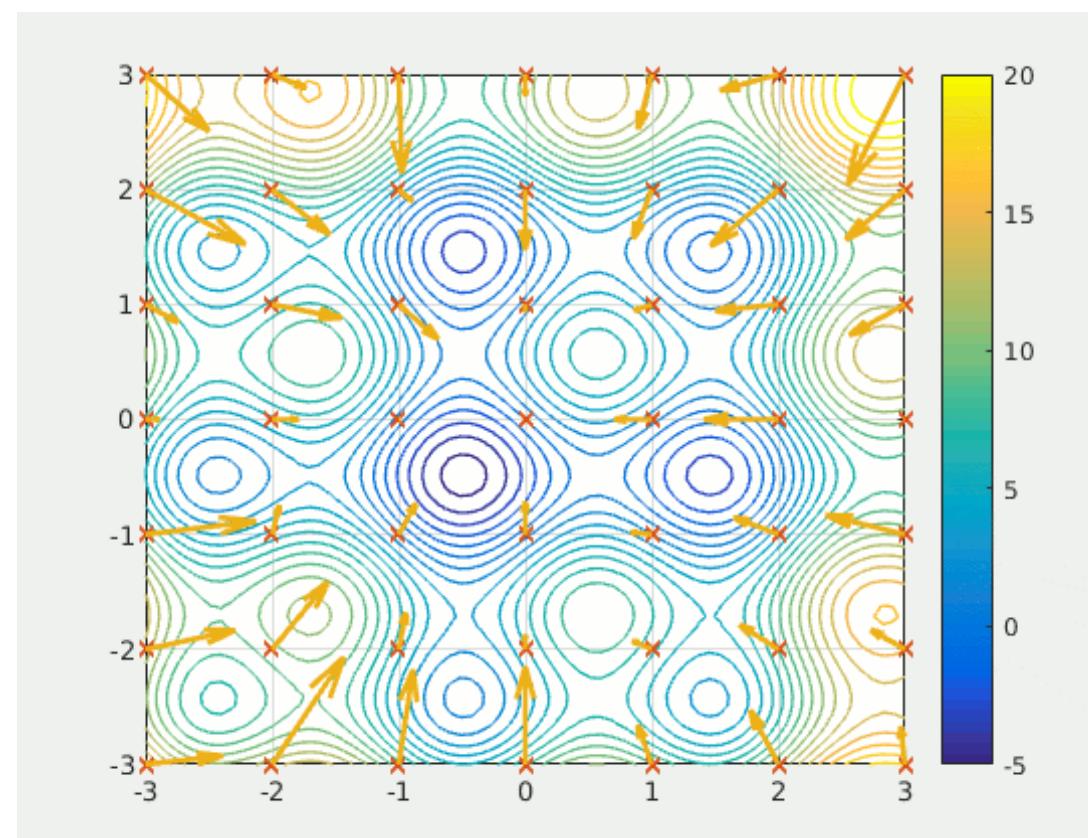
Major advantages for Emulate3D:

- Less likely to get trapped in a local minima
- Can be stopped at any time
- Can be easily parallelized

A balance between **Exploration** and **Exploitation**

Exploration: Venture into new areas of the search space

Exploitation: Refine promising solutions



Metaheuristic Algorithm Options in Test Runner

Many options offered out of the box, suitable for different applications

Algorithm	Description	Advantage
Particle Swarm Optimization	Particles gravitate towards their historic best position and the global best position	Strong exploitation to refine best solution
Genetic Algorithm	Evolutionary approach, the best parents produce children with random mutations	Strong exploration to find novel solutions
<i>What's Planned - Tabu Search</i>	<i>Intelligently avoids searching previously explored regions of the search space</i>	<i>Reaches a conclusion with fewer tests</i>



Optimizers

Automatically generate tests, learn from the results, and repeat to find an optimum solution

Specify parameters to vary and their range:
AMRs, 10 – 50,
Merge Delay, 0 – 0.4s

Specify objective function, the target goal e.g Throughput

Generates and runs tests
Reads results, generates new batch of tests
Reports results the user via dashboard

Coming Soon - live data reporting as tests run



Stages of Virtual Commissioning

Can we connect?

Simply connecting to the model finds issues:

Map IO to virtual equipment

Connect the HMIs

Initialize and reset alarms

Check safety feedback

Can we run in manual?

Begin testing system by forcing values:

Verify sensor feedback

Dry run motors

Test manual functions

Step through the sequence

Can we run in auto?

Create product and see how our system responds:

Run idealized cycle on auto

Reposition field sensors

Verify predicted throughput

Test safety, stops, restarts

Ready to go on site?

Wait there is more!

...Maximising Value from Virtual Commissioning

Can we connect?

Can we run in manual?

Can we run in auto?

Advanced testing

Operator Training

System Upgrades

**Testing in the virtual world
is easier than in the real:**

Inject device faults

Check alarms &
diagnostics

Run varied load schedules

Stress test the system

**Familiarize and train on an
accurate model:**

Familiarize with HMIs

Run training scenarios

Train on device failures

Grade operator responses

**Reuse the model to test
potential changes:**

Regression test code
changes

Optimize performance

Test hypothetical scenarios

Reproduce issues virtually



Using Test Runner for Emulations

Automated testing of your controls code, enabling CI/CD workflows

Create a suite of test scenarios:

- Inject simulated device faults
- Run different product variations
- Simulate operator interactions
- Stress test the system

Define expected behaviour with Assertions:

- Add to the model or to the scenario
- Define severity level, prerequisite conditions

**Run test suite from Test Runner every time
you make a PLC code or design change**

See a dashboard of test results

The screenshot shows a software interface for managing test scenarios. At the top, there's a search bar and a back arrow. The main title is 'EStop on Turntable'. Below it, it says 'Based on None'. On the left, there's a vertical toolbar with icons for search, file, project, and other functions. A large blue button labeled 'ADD ACTION' is centered. The main area is a table with the following data:

	Enabled	Time	Description	Source	Severity	Actions
1	<input checked="" type="checkbox"/>	17	Set ESTOP1.EStopPressed to true			edit copy delete
2	<input checked="" type="checkbox"/>	20	Assert that STA4.TRN.State equals false		High	edit copy delete
3	<input checked="" type="checkbox"/>	20	Assert that STA4.PR.B.State equals false		High	edit copy delete
4	<input checked="" type="checkbox"/>	30	Set Restart1.ButtonPressed to true			edit copy delete
5	<input checked="" type="checkbox"/>	33	Assert that STA4.TRN.State equals true		High	edit copy delete
6	<input checked="" type="checkbox"/>	33	Assert that STA4.PR.B.State equals false		High	edit copy delete
7	<input checked="" type="checkbox"/>	36	Assert that STA4.PR.B.State equals false		High	edit copy delete



Test Runner for Emulation

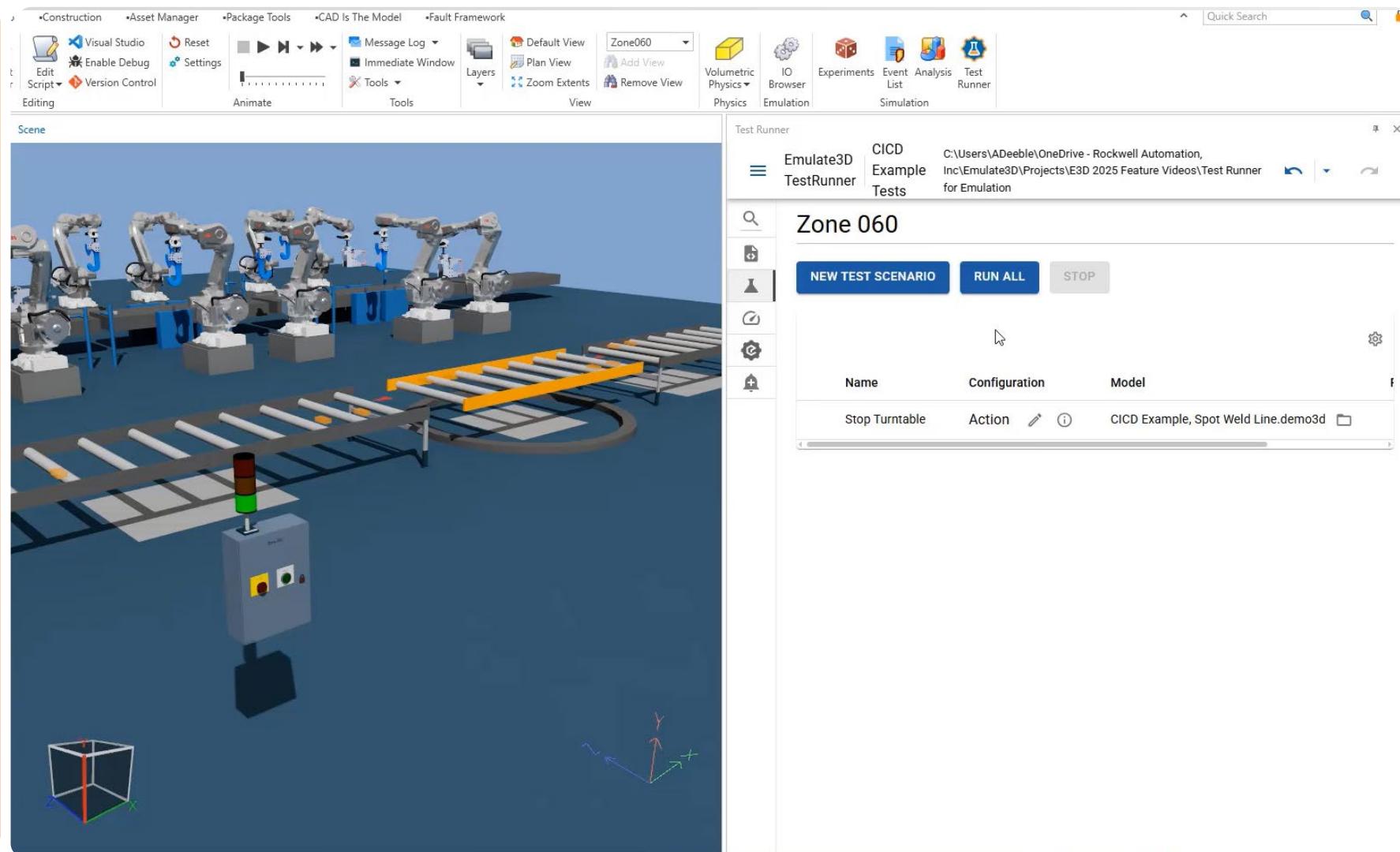
Create regression tests for your controls code

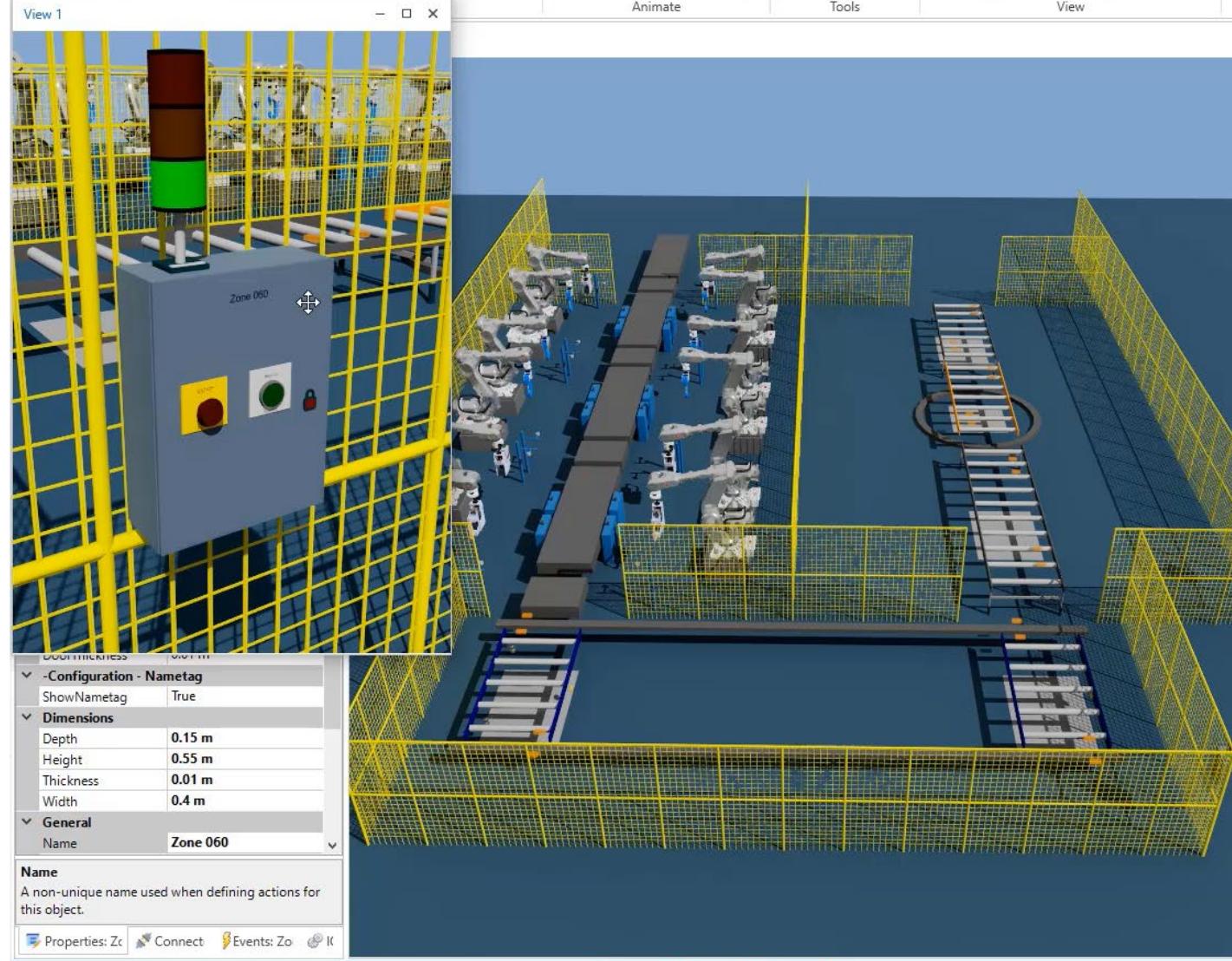
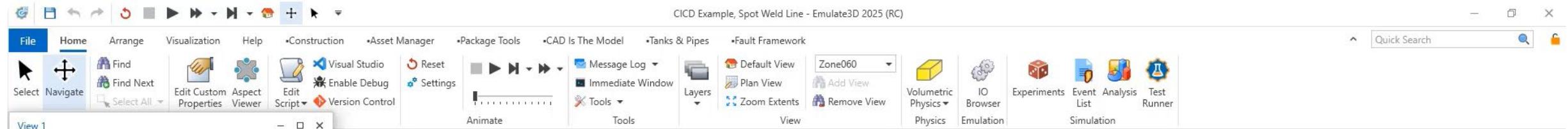
Create a set of repeatable actions which can be run against your PLC code.

Define Assertions in the test, or in the model.

Run the test suite, report on Pass/Fail/Warnings.

Tip - The Emulation State component can help each test automatically connect





Test Runner

Emulate3D CICD Example Tests C:\Users\ADeeble\OneDrive - Rockwell Automation, Inc\Emulate3D\Projects\E3D 2025 Feature Videos\Test Runner for Emulation

Zone 060

NEW TEST SCENARIO RUN ALL STOP

Name	Configuration	Result
Stop Turntable	Action	Not Run
Stop After Turntable	Action	Not Run
Stop on transfer to 070	Action	Not Run
Stop on cam lift table	Action	Not Run



Test Runner with Fault Framework

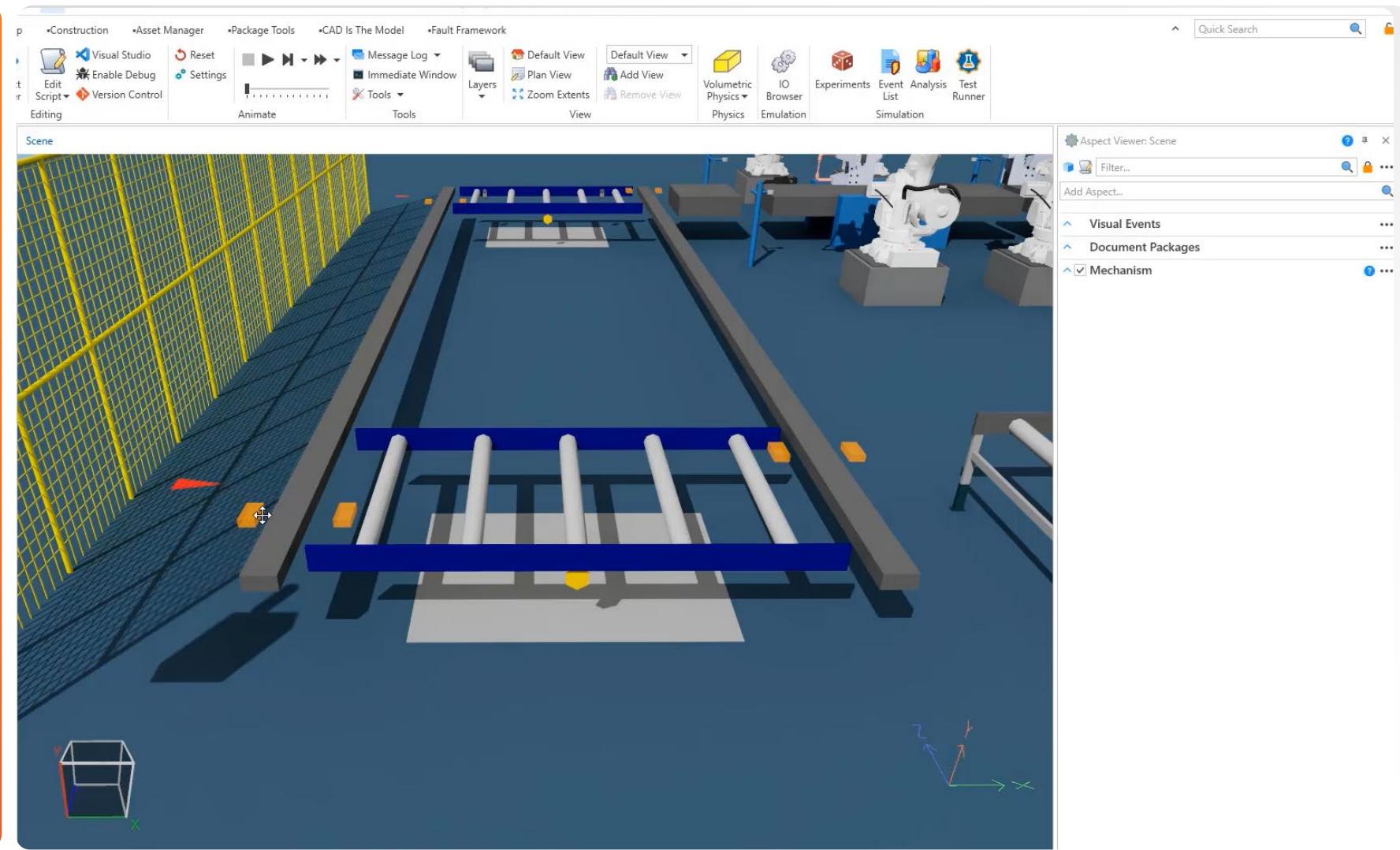
Simulate device faults using the Fault Framework aspects, from within Test Runner

Easily trigger faults from the Test Runner.

Tests adhere to Fault Framework assertions, or to test actions.

Also applicable to Operator Training

Coming Soon - Dedicated Fault action with autofill





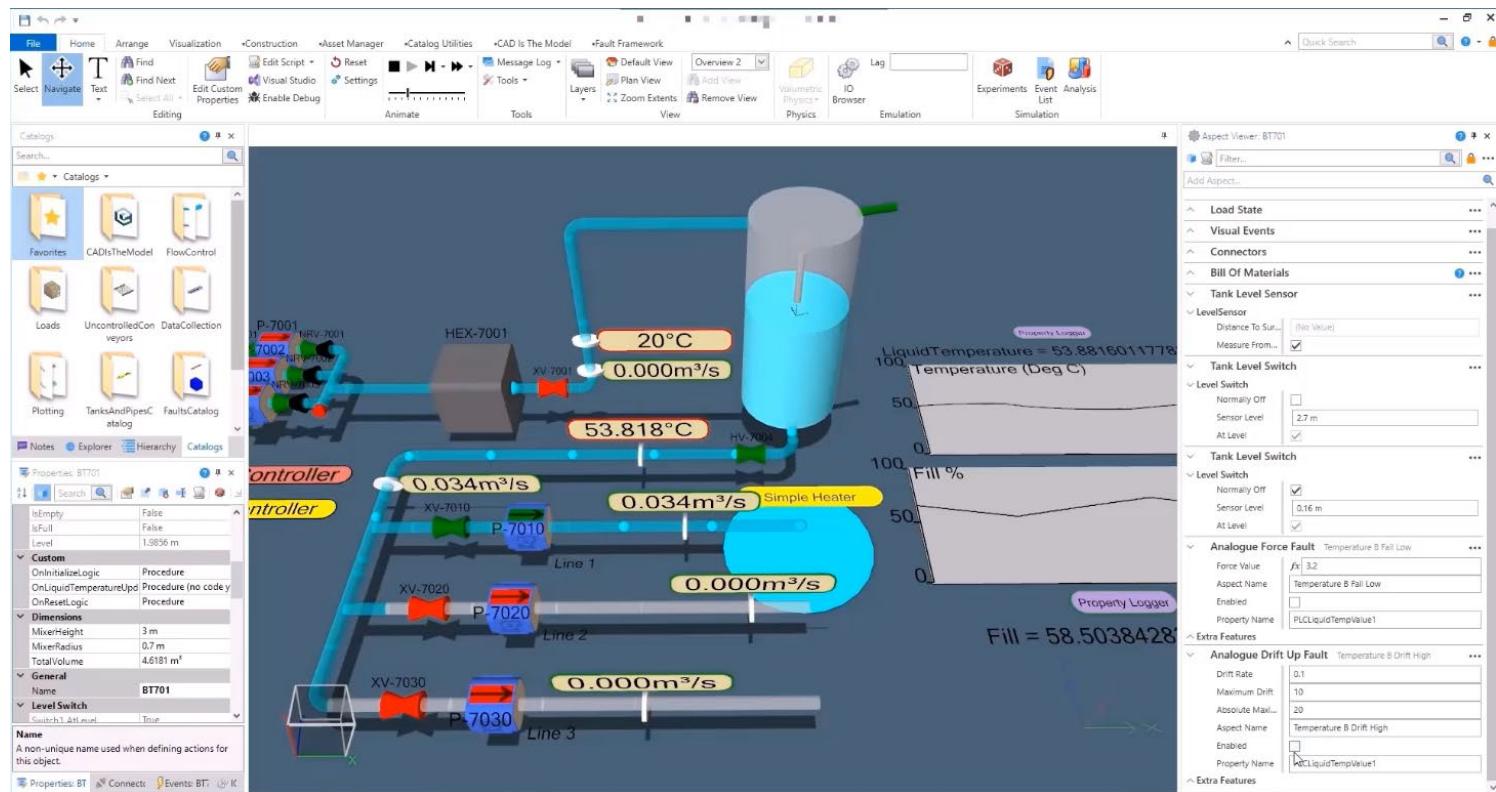
Operator Training

Fault Framework and Test Runner enables Operator Training workflows

Create operator tests with actions and device faults for operators to diagnose

Use assertions to see if the operator correctly resolved the challenge

Automatic log of all interactions, ability for operator to log notes



Test Runner is also for Catalog Development...

... but you'll hear about that in another presentation!

Use actions and assertions for simple tests

NUnit support in custom scripting enables advanced testing

CI/CD for custom catalog development:

- Version control with integrated GIT UI
- Run automated tests with Test Runner
- Deploy to custom package feed

Let's first see GIT version control in detail!

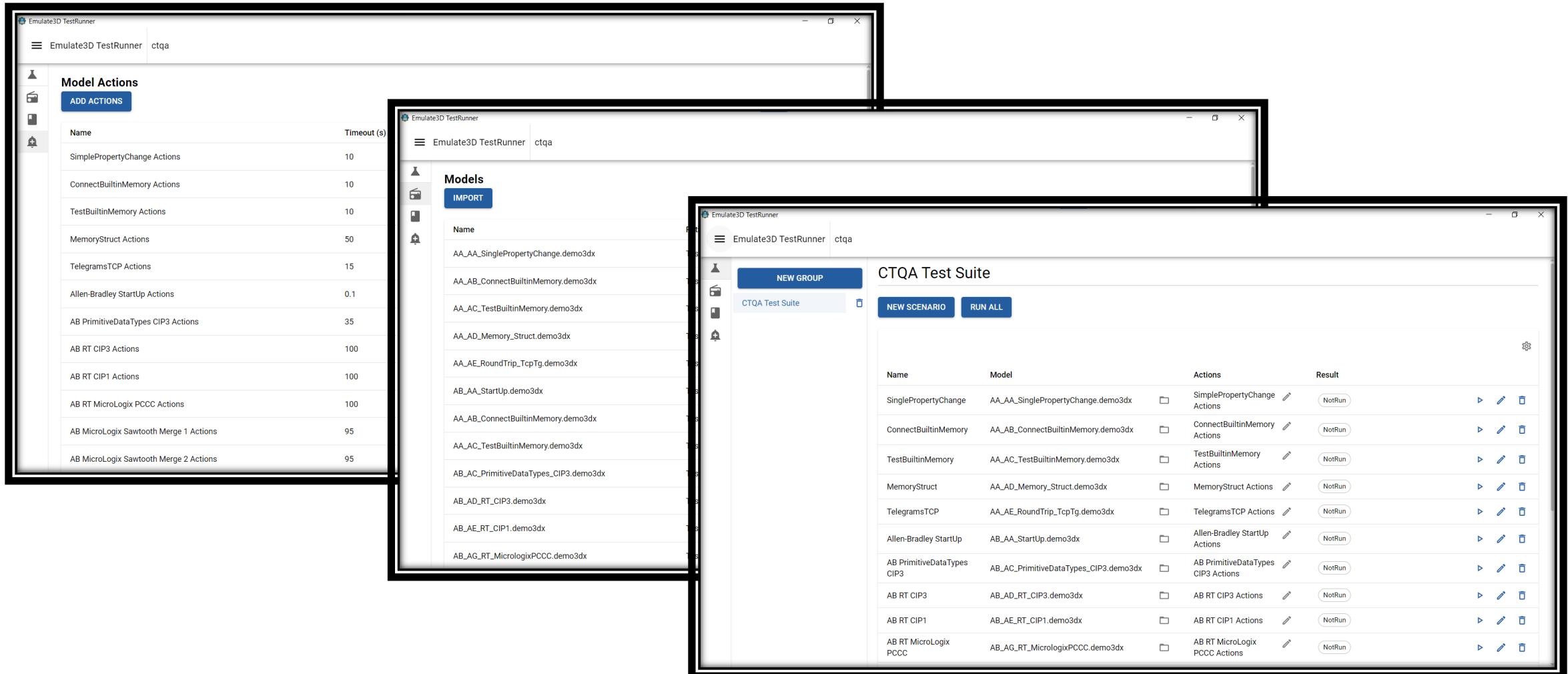
```
[TestFixture]
public class MyTests : Emulate3DTestFixture {
    [Emulate3DTest]
    public async Task Pass() {
        await Task.Delay(2000);
        Assert.Pass();
    }

    [Emulate3DTest]
    public async Task Fail() {
        await Task.Delay(2000);
        Assert.Fail();
    }

    [Emulate3DTest]
    public async Task ConveyerTest() {
        var conv = App.Document.FindVisual("SB1");
        Assert.IsNotNull(conv);
        var lc = conv.FindChild("LoadCreator1");
        Assert.IsNotNull(lc);
        if (lc.Props is not LoadCreatorProperties props) {
            Assert.Fail("Unexpected type");
            return;
        }
        props.DelayRate = 20;
        props.CongestionZone = false;
        App.Invoke(App.Reset);
        RunActionAtTime(5, () => {
            var loadCount = App.Document.PhysicsEngine.CountLoads();
            Assert.That(loadCount, Is.GreaterThanOrEqualTo(5),
                "Not enough loads in the scene, is the LoadCreator1 release rate too low?");
            Assert.That(loadCount, Is.LessThanOrEqualTo(10),
                "Too many loads in the scene, is the LoadCreator1 release rate too high?");
        });
        await RunModelForSecondsAsync(5.1, 2);
    }
}
```

Test Runner is used by the Emulate3D Developers...

...for catalog, regression, communications, and performance testing!



The image displays three separate windows of the Emulate3D TestRunner application, each showing a different aspect of the tool's functionality:

- Model Actions:** This window shows a list of various actions categorized under "Model Actions". Each entry includes a "Name" (e.g., "SimplePropertyChange Actions", "ConnectBuiltinMemory Actions", "TestBuiltinMemory Actions", etc.) and a "Timeout (s)" value (e.g., 10, 10, 10, 50, 15, 0.1, 35, 100, 100, 100, 95, 95). A "ADD ACTIONS" button is visible at the top.
- Models:** This window shows a list of imported models. The "IMPORT" button is visible at the top. The list includes: AA_AA_SinglePropertyChange.demo3dx, AA_AB_ConnectBuiltinMemory.demo3dx, AA_AC_TestBuiltinMemory.demo3dx, AA_AD_Memory_Struct.demo3dx, AA_AE_RoundTrip_TcpTg.demo3dx, AB_AA_StartUp.demo3dx, AA_AB_ConnectBuiltinMemory.demo3dx, AA_AC_TestBuiltinMemory.demo3dx, AB_AC_PrimitiveDataTypes_CIP3.demo3dx, AB_AD_RT_CIP3.demo3dx, AB_AE_RT_CIP1.demo3dx, and AB_AG_RT_MicrologixPCCC.demo3dx.
- CTQA Test Suite:** This window shows a "CTQA Test Suite" with a "NEW GROUP" button at the top. It lists 13 test cases with columns for "Name", "Model", "Actions", and "Result". All test cases are currently marked as "NotRun".

Name	Model	Actions	Result
SinglePropertyChange	AA_AA_SinglePropertyChange.demo3dx	SimplePropertyChange Actions	NotRun
ConnectBuiltinMemory	AA_AB_ConnectBuiltinMemory.demo3dx	ConnectBuiltinMemory Actions	NotRun
TestBuiltinMemory	AA_AC_TestBuiltinMemory.demo3dx	TestBuiltinMemory Actions	NotRun
MemoryStruct	AA_AD_Memory_Struct.demo3dx	MemoryStruct Actions	NotRun
TelegramsTCP	AA_AE_RoundTrip_TcpTg.demo3dx	TelegramsTCP Actions	NotRun
Allen-Bradley StartUp	AB_AA_StartUp.demo3dx	Allen-Bradley StartUp Actions	NotRun
AB PrimitiveDataTypes CIP3	AB_AC_PrimitiveDataTypes_CIP3.demo3dx	AB PrimitiveDataTypes CIP3 Actions	NotRun
AB RT CIP3	AB_AD_RT_CIP3.demo3dx	AB RT CIP3 Actions	NotRun
AB RT CIP1	AB_AE_RT_CIP1.demo3dx	AB RT CIP1 Actions	NotRun
AB RT MicroLogix PCCC	AB_AG_RT_MicrologixPCCC.demo3dx	AB RT MicroLogix PCCC Actions	NotRun



Thank you

www.rockwellautomation.com



