Executable Scenario Definition Using Datalog to Describe Simulation Capabilities

US Army Research Laboratory (ARL)
Human Research and Engineering Directorate (HRED)
Simulation & Training Technology Center (STTC)
• The Next Generation Architectures (NGA) for Modeling & Simulation (M&S) research project refines and demonstrates advances in computer science that support the development of M&S architectures required for future training, experimentation and acquisition decisions.

• This presentation discusses an initial effort under this project area to identify scenario primitives that can link analytical data elements to simulation execution objects in a way that supports automated model and simulation selection based on the content of the scenario.
The Situation

• Simulation models with fixed interfaces
• Explicit integration
• Failing events due to models and simulation not meeting expectations
I provide Stress Levels

Stress-By-Detonations Model

I need Stress Levels and Munition Detonations

Detonation Model

I provide Munitions Detonations

Simulation models with fixed interfaces
What we have

I provide Stress Levels

Stress-By-Detonations Model

I'll take Munitions Detonations And request Stress Levels

Detonation Model

I provide Munitions Detonations

Explicit integration

*SFC Paul Ray Smith Simulation & Training Technology Center

*Unclassified – For Public Release*
What we have

I provide Stress Levels

I provide Munitions Detonations

Stress-By-Detonations Model

Detonation Model

Invalid Stress Levels

I’ll take Munitions Detonations And request Stress Levels

Failing events

Detonation Event(s)
What we want

- Fluid, discoverable simulation models
- Model sandboxing
- Integration via composition
What we want

Fluid, discoverable simulation models

I need Stress Levels and Munitions Detonations
What we want

OK.
I'll check

I need Stress Levels and Munitions Detonations

Fluid, discoverable simulation models
What we want

Anyone provide Stress and Detonations?

Stress-By-Detonations Model

Broker

Detonation Model

Fluid, discoverable simulation models
What we want

- I provide Stress Levels
- Stress-By-Detonations Model
- I provide Munitions Detonations
- Detonation Model

Broker

Fluid, discoverable simulation models

*Unclassified – For Public Release*
What we want

Do these Work for You?

Stress-By-Detonations Model

Broker

Detonation Model

Fluid, discoverable simulation models
What we want

I don’t know. I need a finer-grained Stress Level

Fluid, discoverable simulation models
What we want

OK. Why don’t you send me a few tests

Stress-By-Detonations Model

Detonation Model

Broker

Model sandboxing
What we want

Stress-By-Detonations Model

Test detonations

Stress requests

Broker

Detonation Model

Model sandboxing
What we want

Stress-By-Detonations Model

Test detonations

Stress levels

Detonation Model

Stress requests

Broker

Model sandboxing

Stress levels

*Unclassified – For Public Release*
What we want

Stress-By-Detonations Model

Detonation Model

That doesn’t look right. I need casualty stress too.

Model sandboxing
What we want

I know about another one, give me a second

Stress-By-Detonations Model

Detonation Model

Broker

Integration via mix-ins

*Unclassified – For Public Release*
What we want

Integration via mix-ins

Start Stress-By-Casualties Model

Broker

Stress-By-Detonations Model

Detonation Model
What we want

Stress-By-Detonations Model

Stress-By-Casualty Model

Detonation Model

Can you two compose?*
*Always a tricky question!

Broker

Integration via mix-ins
What we want

Stress-By-Detonations Model

Well, try this out.

Test detonations
Stress requests

Stress levels

Stress levels

Detonation Model

Stress-By-Casualty Model

Broker

Model sandboxing

*Unclassified – For Public Release*
What we want

Stress-By-Detonations Model

Stress-By-Casualty Model

Detonation Model

How did that look?

Broker

Integration via mix-ins
What we want

Stress-By-Detonations Model
Stress-By-Casualty Model
Detonation Model

Integration via mix-ins

Looks great! Let’s begin.
What we want

Stress-By-Detonations Model

Stress-By-Casualty Model

Stress requests

Detonations

Detonation Model

Stress levels

Broker

Detonation Event(s)

Stress levels

Successful events
What we want

• Fluid, discoverable simulation models
  – Models need to be able to describe what they “think” that they can do

• Model sandboxing
  – Consumers and models need a way to test their interactions in a safe way to determine viability

• Integration via composition
  – Model composability needs to be expressible
We need a “primitive” language

- To express wants and needs
- To express capabilities
- To express composition of capabilities
- That can “host” existing systems
We need a primitive “primitive” language

- Simple
- Syntactically composable
- Syntactically inferential

Simple solution: Datalog
Hi, I’m a tuple!

[<entity> <attribute> <value>]

*Unclassified – For Public Release*
The entity part is a referent (think identity)

\[<\text{entity}> \ <\text{attribute}> \ <\text{value}>\]
The attribute part is a tagged property

[<entity> <attribute> <value>]

*Unclassified – For Public Release*
The value part is a property’s value for an entity

[<entity> <attribute> <value>]

Datalog
[Stress-By-Detonations Model provides Stress]
Stress-By-Detonations Model

[Stress-By-Detonations Model :provides Stress]

[User :needs Stress]
Stress-By-Detonations Model

Stress-By-Detonations Model :provides Stress

Stress-By-Detonations Model :stress/component Detonations

Stress-By-Casualty Model

Stress-By-Casualty Model

Stress-By-Casualty Model :provides Stress

Stress-By-Casualty Model :stress/component Casualty
• Allows inference and querying
• Hosts primitive “model” AND Object Models
• Simple
• Self-describing / self contained
Executable scenarios

This is a Scenario
(has “desired capabilities”)

Simulation Functions
(has “provided capabilities”)

Win!
Red and Blue start with an obstruction between them. They move forward until they see each other and engage.
System overview

Broker

Peer

Peer

OneSAF

Movement Surrogate

Weapons Surrogate
<table>
<thead>
<tr>
<th>Capability</th>
<th>Primitives</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run-time infrastructure</td>
<td>n/a</td>
<td>This is a special category of capability</td>
</tr>
<tr>
<td>Plan View Display</td>
<td>n/a</td>
<td>This is a special category of capability</td>
</tr>
<tr>
<td>Creation</td>
<td>Time, Location, Heading, Units, Platforms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Side, Orientation</td>
<td></td>
</tr>
<tr>
<td>Movement</td>
<td>Latitude, Longitude, Velocity, Orientation</td>
<td></td>
</tr>
<tr>
<td>Damage</td>
<td>Health</td>
<td></td>
</tr>
<tr>
<td>Fires</td>
<td>Weapons Control</td>
<td></td>
</tr>
<tr>
<td>Vision</td>
<td>Sensation, Orientation</td>
<td></td>
</tr>
</tbody>
</table>
• Persistent Peer architecture
• Inferential database
• Browser-based user interface
• Hyper Text Transfer Protocol (HTTP)-based system-to-system interface
  – Messages containing collections of tuples
    • I.E. [<entity> <attribute> <value>]

*Unclassified – For Public Release*
Peer architecture

EASE: Executable Architecture Systems Engineering – contact authors for more information
Peer architecture

- Persistent connections
- Capabilities declaration
- Launch logic
Lessons learned

• We can use the Broker/Peer technologies now, and in the future
  – Coordinator (persistent connections, launch)
  – EASE (meta-model and inference)
• We need to think harder about simulation primitives
• We need to talk to some analysts
• We need to explore sandboxing in depth
• We need to explore auto-extraction of desired capabilities from scenarios
Authors

Chris Gaughan
Christopher J. Metevier
Simulation and Training Technology Center,
Human Research and Engineering Directorate,
Army Research Laboratory,
Orlando, Florida

Mike Fogus
Joseph S. McDonnell, Ph.D.
presenter – joe.mcdonnell@d-a-s.com
703-474-7038
Dynamic Animation Systems, Inc.
Fairfax, Virginia

Scott Gallant
Effective Applications Corp.,
Orlando, Florida

*Unclassified – For Public Release*
Questions/Comments?

ARL-HRED-STTC
Public Affairs Office
(407) 384-5227
Additional Reference