Using Systems Engineering to Identify and Develop Key Advancements in M&S

Presentation to

NDIA Systems Engineering Conference

October 2015

Mr. Scott Gallant, Systems Engineering
Mr. Derrick Franceschini, Software
Dr. Mark Riecken, Chief Engineer
Mr. John Rutledge, Program Manager

RESEARCH AND DEVELOPMENT OF DEPARTMENT OF DEFENSE MODELING AND SIMULATION NEXT GENERATION TRAINING ARCHITECTURE

Contract N61340-14-C-1010

Modeling & Simulation Coordination Office

STACKFRAME

Effective Applications

TRIDEUM

Analyze. Integrate. Innovate.

Contents

- Project Summary
- Systems Engineering Approach
- Osseus Prototype Description
- Lessons Learned and Future Effort

WebSTAR Project Summary

- ✓ **Project execution was only nine months (OCT 2014 - JUN 2015)**
- ✓ **Systems Engineering effort to develop technical requirements with traceability to community issues**
- ✓ **Software development to create a working Government-owned prototype – the *Osseus Platform***
- ✓ **Lessons Learned and Recommended Way Forward**

Project Objectives

1) *conduct and document applied research* so as to be able to further the community's understanding of using web technologies for distributed simulation, and

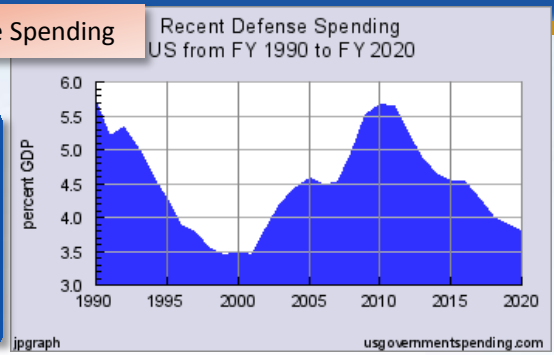
2) develop a *prototype software capability* suitable for limited experimentation in unclassified network environments and suitable for demonstrating key concepts of this approach

M&S Timeline

40 Year Timeline of M&S, Technology, and World Events

Significant changes have occurred since development of standing M&S architectures

Defense Spending



- 1968 – ArpaNet
- 1970 - NPT
- 1974 – EtherNet
- 1975 – MicroSoft / ABQ
- 1975 – Vietnam War ends

M&S

SIMNET 1987
ALSP 1990
1278 (1993) DIS
STOW-E
STOW '97 (1997) HLA Proven
1516 (2000) HLA
CTIA
TENA
DSEEP
WoW 10M Subscribers
Illustris*
Army STE*
Army Force 2025

1985-1990 | 1990-1995 | 1995-2000 | 2000-2005 | 2005-2010 | 2010-2015 | 2015-2020 | 2020-2025

Technology

HTML Berners-Lee
Google
1st DNS Attack
Fielding RESTful Ph.D.
Human Genome
1st Multi-Core Processor
Amazon EC2
iPhone
IPv6
Higgs
Watson Jeopardy
NoSQL "re-introduced"
HTML 5 Spec Finalized
1.4 Bn Facebook Users
26 Bn Devices on IoT (Gartner)
1 in 3 jobs: SW or robots (Gartner)
1 M commercial UAVs (Lux Populi)

World Events

4.8 Bn (World Pop.)
Berlin Wall falls
Desert Storm
Bosnian War ends
WTC
Snowden
ISIL
New START treaty expires (2011)
8.1 Bn (World Pop.)
China $17.6 T PPP
US $17.4 T PPP

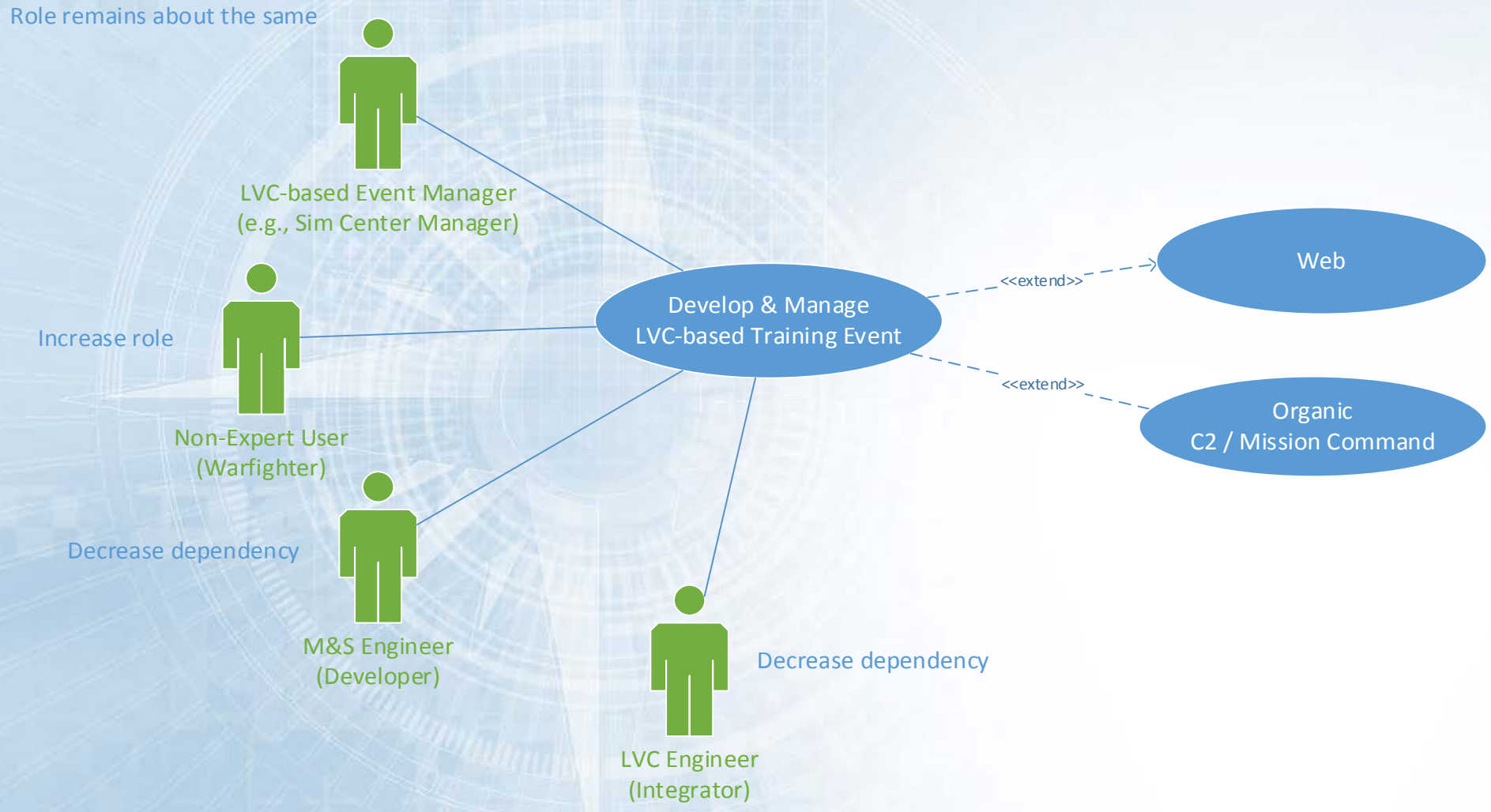
40 years

EC2 – Elastic Compute Cloud
Illustris = Simulation of the Universe (MIT, Harvard etc)
NPT = Nuclear Non Proliferation Treaty
PPP = Purchasing Power Parity

STE = Synthetic Training Environment
STOW-E = Synthetic Theater of War – Europe
WoW = World of Warcraft

DMSCO NGA Project - 2015

WebSTAR Project Use Case



Architecture Review

Table of Contents

Key Features (What's in it?)

- Survey of current state of distributed M&S architecture
- High level issues (basis of technical approach)
- Discussion of emerging technologies

Takeaways (What does it say?)

- Current state of the practice: complex, highly technical environment
- M&S development are out of phase with system development
- Foundations of current state of the art developed in pre-internet era

Technical Approach Development

Key Features (What's in it?)

- Explanation of approach using High Level Issues, User Stories, Requirements
- Epic and Focused User Stores
- Technical Design Decisions & Rationale

Takeaways (What does it say?)

- Services, optimized data transports, filtered data sharing to optimize network traffic
- Graphical interface for behaviors & business logic
- Service registration, discovery, proxying, and relaying
- Web-based visualization
- Behaviors implemented using framework – no "affiliate" changes required

Technical Framework

Key Features (What's in it?)

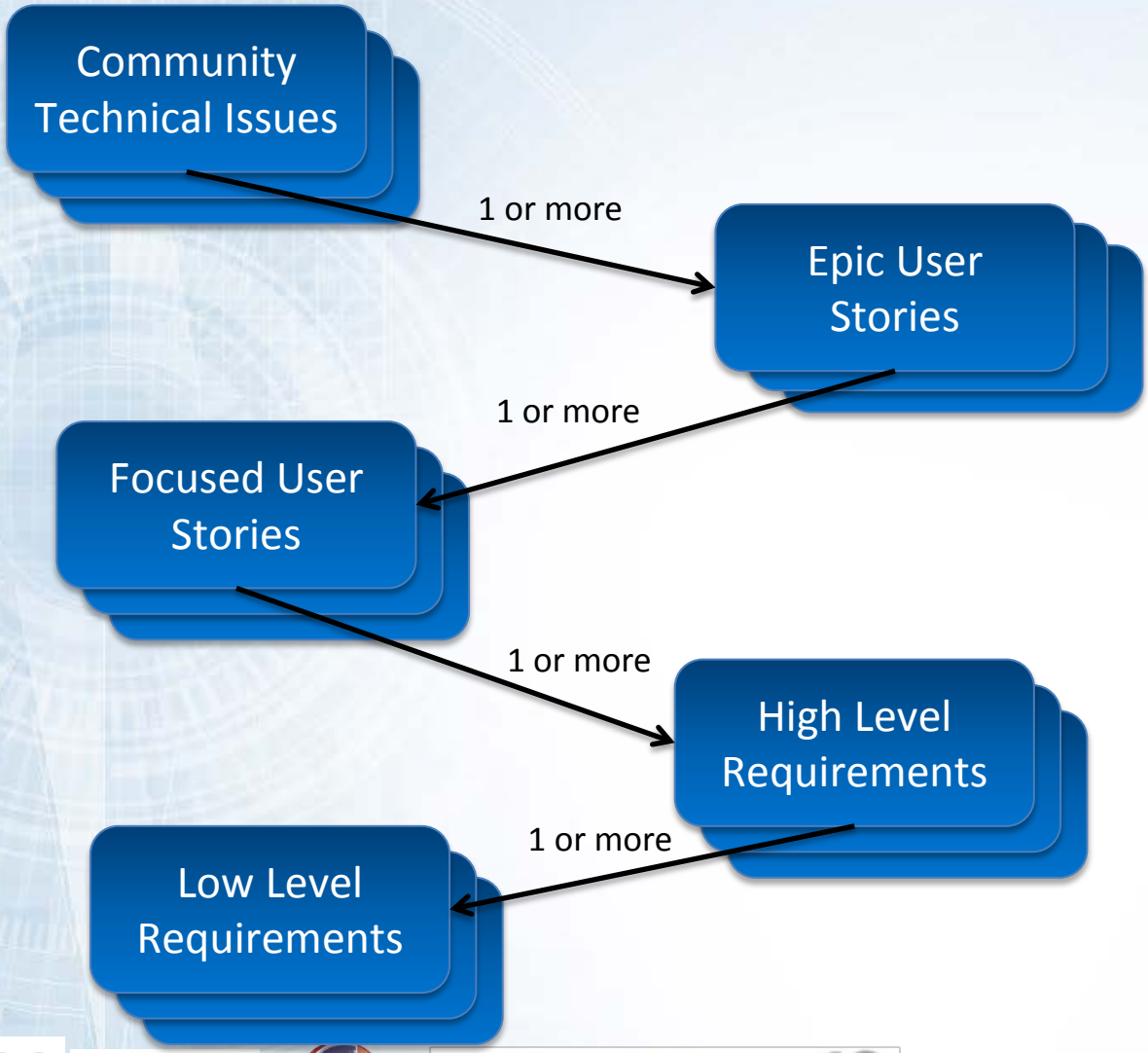
- Technical Framework (High Level Design)
- Platform, Storage, Transport, Services, Utilities, Behaviors
- Appendix of JSON HTTP and RESTful examples

Takeaways (What does it say?)

- Provides details of Osseus prototype framework (architecture)
- Lists Low Level Requirements (LLR) that are being implemented in this phase (through June 11) of prototype development – derived from requirements in CDRL A002

Traceability

1. Discover & Capture
2. Organization
3. Targeting
4. Requirements & Design
5. Specification



Epic User Stories to LLR

EUS01: As an LVC training system provider, I want to adjust the technical implementation to satisfy a training event's objectives so that I can provide a relevant solution rather than forcing the training event to adjust to the existing technical implementation. Technical Issue References: TI01, TI03, TI04, TI20, TI21 and TI24.

EUS02: As an LVC training system provider, I want to allow communication and interoperability among existing DoD simulations that already work in HLA environments. Technical Issue References: TI13, TI23 and TI24.

EUS03: As an LVC training system provider, I want to reduce the amount of time and errors for bringing in a new system into an existing distributed M&S environment. Technical Issue References: TI05, TI15 and TI16.

Epic User Stories

US01 - As an LVC System I want to be able to discover available services upon starting and joining the SoS so that I can use distributed services for common tasks and concentrate on my functionality. Epic User Story Reference: EUS02.

HLR001 – Osseus shall provide discovery services for modeling capabilities, services, and current object state.

US05 - As a simulation engineer, I want to join the Alliance [1]at any time and be able to obtain the latest data state without burdening or relying on other Affiliates[2]. Epic User Story Reference: EUS07.

HLR006 - Osseus shall allow for late joining Affiliates to get the latest state of objects without burdening all other Affiliates.

HLR010 - Osseus shall provide a single server instance to be the central connection point from all the distributed client applications.

Focused User Stories & High Level Requirements (HLR)

Low Level Requirements (LLR)

LLR001 - Osseus shall include a server with a RESTful API for Affiliate registration and service advertisements. [HLR001]

LLR002 – Osseus shall maintain a registry of available services. [HLR001]

LLR003 – Osseus shall provide a RESTful query for available services, the ability for a service to start and register itself, and the ability for a service to shut down and become unavailable. [HLR001]

LLR004 - Osseus shall provide Adaption services to translate between the organic data model and Affiliate data exchange models including HLA, DIS, and TENA. [HLR002]

LLR005 - Osseus shall provide a behavior representation data format that is based on object's data model and semantics. [HLR003]

Lessons Learned and Future Effort

Key Features (What's in it?)

- Lessons Learned
- Significant barriers to technical progress
- Future Effort recommendations
- Annotated bibliography

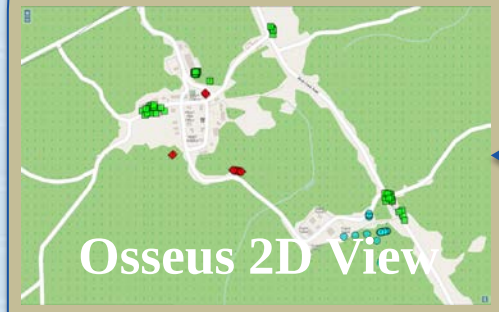
Takeaways (What does it say?)

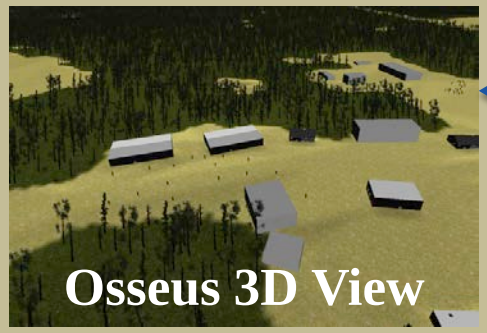
- Detailed discussion of lessons learned (results / findings)
- Work Breakdown Structure (WBS) approach to future effort

The Osseus Platform (Objective)

An open source development and execution environment facilitating interoperability of multiple architectures

Osseus Platform





Osseus Server

Osseus Adapter

Osseus Adapter

Osseus Affiliate (Existing Application)

Osseus Affiliate (Existing Federation)

Osseus Affiliate (Individual Model)

Osseus Affiliate (Individual Service)

Osseus Behavior Designer

Osseus Metrics Dashboard

Osseus Alliance

Osseus Lessons Learned

- Adaptation platform eases interoperability barriers
- Need for Technology Refresh to Keep Pace with Commercial Advancements
- Behavior (Content) Authoring Can be accomplished by Non-Expert Users
- Adaptation and Filtering Services Can Relieve System Bottlenecks and Reduce Code Development Time
- Measures can and should be “Built In” to the Architecture
- Focusing on Automating Interoperability Could Provide Significant Dividends

Thank you

- **Questions?**

- **Contact Information**
  - Project questions: John Rutledge (jrutledge@trideum.com)
  - M&S Application: Dr. Mark Riecken (mriecken@trideum.com)
  - Systems engineering: Scott Gallant (Scott@EffectiveApplications.com)
  - Software Development: Derrick Franceschini (derrick@stackframe.com)

Need for Technology Refresh to Keep Pace with Commercial Advancements

Technology	Impact
NoSQL and Time Series Databases	NoSQL facilitates a broader acceptance of data models and data structures into a working distributed M&S LVC architecture. TSDB can be used to support metrics collection and data analysis. database technology.
REST and Web Sockets	Using these two connection mechanisms together provides the flexibility to use the most appropriate mechanism depending on the Affiliates and the type of data being transferred. This is an upgrade over traditional M&S middleware infrastructures where only one type of connection is possible.
Internet of Things (IoT)	Exponential growth in "things" that may need to be incorporated into an event. May become a significant driver in the need to automate aspects of interoperability.
HTML5	HTML5 is rich enough to provide very useful user experiences through a web browser. This is significant because it further supports a web-based approach to user interfaces enabling more of a distributed computing paradigm.
Artificial Intelligence	AI can be expected to contribute multiple areas in distributed M&S. A few examples follow. 1) Providing assistance to the user in enhanced UIs through related technologies such as natural language processing (NLP); 2) in conjunction with modern database and Big Data technologies, AI can assist directly in implementing automated aids to interoperability; and 3) AI can increase the fidelity and richness of simulated entity behaviors.
Data Distribution Service (DDS)	DDS is a more modern data sharing mechanism that is being used commercially and has developer momentum. Using an approach popular within the commercial sector allows M&S COP to take advantage of a large community of developers making advances.
Algorithm Research	Algorithm research focused on automating many if not all interoperability engineering tasks is potentially one of the most promising technology frontiers of all. Interoperability is based on data models and protocols that are generally published, well understood by both humans and machines.
BRASS (a DARPA project)	Changing M&S software, standards, protocols, and data models are one set of examples of the types of software that could benefit from research efforts like BRASS (and others). BRASS technologies could, if successful, directly contribute to the automation of interoperability mechanisms, especially when considering the asynchronous nature of updates to M&S software and related data.

Future Effort WBS