TECHNICAL ARTICLE

SIMULATION 73:5, 266-272

ISSN 0037-5497/99
Printed in the United States of America

Trailblazer: An Application of The High Level Architecture To Joint Experimentation

Anita A. Zabek
The MITRE Corporation
1820 Dolley Madison Boulevard
McLean, Virginia, USA
E-mail: anita@mitre.org

The Trailblazer project was created to examine the application of a federation of simulations to joint experimentation. The purpose of this paper is to describe the Trailblazer federation. The paper is organized by the steps of the Federation Development and Execution Process (FEDEP). The FEDEP is a systems engineering process for the development of federations of simulations using the High Level Architecture (HLA) and was used as the roadmap for development of the Trailblazer federation. Lessons learned are presented for each step of the FEDEP, and a summary presents overall observations on the usefulness of the FEDEP and the HLA in the joint experimentation domain.

Keywords: High Level Architecture, federation development and execution process, experimentation, analysis, Trailblazer

1. Introduction

The Trailblazer effort has its origins in 1998 in the Joint Warfighting Program, sponsored by the Defense Development Research and Engineering (DDR&E). The DDR&E objective was to examine how changes in doctrine, organization, and technology can enhance our ability to achieve and maintain information superiority in a hostile environment. DDR&E envisioned that the Joint Warfighting Program would design and conduct a series of experiments to examine these issues, and tasked the Defense Modeling and Simulation Office (DMSO) to provide technical support to those experiments involving modeling and simulation.

In an effort to better understand the requirements of modeling and simulation for experimentation, DMSO implemented the following strategy:

- Use the High Level Architecture to take advantage of existing simulation capabilities,
- Develop a prototype federation (Trailblazer) in support of experimentation to gain relevant experience.

The FEDEP [1] was used as a guide for federation development. The FEDEP is a five-step systems engineering process and is presented in a simplified form in Figure 1.

Define Federation Objectives	Develop Federation Conceptual Model	Design and Develop Federation	Integrate and Test Federation	Execute Federation and Analyze Results
Identify Sponsor Needs	Develop Scenario	Design Federation	Plan Execution	Execute Federation
Develop Objectives	Perform Conceptual Analysis	Develop Federation	Integrate Federation	Analyze Results
	Develop Federation Requirements		Test Federation	Prepare Feedback

Figure 1. Federation Development and Execution Process (FEDEP)

2. Define Federation Objectives

Step 1 of the FEDEP is ***Define Federation Objectives***. The FEDEP describes this step as: "The federation sponsor and federation development team define and agree on a set of objectives, and document what must be accomplished to achieve those objectives." [1]

For Trailblazer, the objectives included those laid out by DMSO, the objectives of the specific experiment, and the unique requirements of the experimentation domain.

2.1 DMSO Objectives and Guidance

The objectives of DMSO were to explore the use of HLA federations in the experimentation domain and to capture the lessons learned, both to improve the technology and to aid developers and users of future applications.

DMSO provided the following specific guidance to the federation systems engineer:

- Use the High Level Architecture,
- Distribute development at federate home sites,
- Conduct integration and execution at single site in Washington, DC, area,
- Operate at a classified, but no higher than collateral SECRET, level,
- Complete development within fixed budget and schedule (less than five months),
- Consider closed-loop, faster-than-real-time simulations.

2.2 Experiment Definition and Objectives

For purposes of this prototype effort, an experiment was adopted from a Joint Staff experiment plan [2]. The experiment hypothesis is that a proposed 2010 architecture for conducting a joint suppression of enemy air defenses mission is more effective than the baseline 2010 architecture. The proposed architecture introduced new technologies, doctrine, and command and control relationships.

The objective of the experiment, as described in the Joint Staff plan, is to prove or disprove the hypothesis. Two measures of effectiveness are identified. The first is the percentage of surface-to-air missile (SAM) batteries identified, engaged, and destroyed. The second is the time from SAM emission to weapon commit and time from weapon commit to weapon on target.

Discussions with the Joint Staff analysts identified an objective to explore the sensitivity of the experimental outcomes to variations in threat operating policy. Three variations were identified:

- Maximum attrition policy (long emission time and short hiding time),
- Maximum survival policy (short emission time and long hiding time),
- Balanced policy.

2.3 Unique Requirements of Experimentation

Four requirements are unique to the domain of experimentation. These are:

- **Validity**—Individual selected federates must be credible, and federation implementation must maintain that credibility.
- **Repeatability**—Execution of the federation with the same inputs should lead to the same outputs.
- **Extractability of Results—**The analyst must be able to collect and correlate data about critical events that occur across federates.
- **Speed—**Requirements for multiple excursions, multiple runs per excursion and limited time imply faster-than-real-time simulation rates.

2.4 Lessons Learned

Because, by definition, an experiment is designed to investigate new things, the experiment designers had to specify the proposed changes to technology, organizations, or doctrine and expected outcomes. Furthermore, all excursions had to be defined.

The experiment designer also specified the measures of effectiveness that were used on the outcome of experiments, because those measures drove the level of fidelity needed in the federation.

Finally, those requirements evolved as the federation was developed. The very process of eliciting this information from the experiment designer enabled evolution of the experiment design.

3. Develop Conceptual Model

Step 2 of the FEDEP is ***Develop Conceptual Model***. The FEDEP describes this step as: "A representation of the real-world domain of interest is developed and described in terms of a set of required objects and interactions." [1]

3.1 The Trailblazer Conceptual Model

The Trailblazer conceptual model first identified the key events in the experiment. These are depicted in Figure 2.

The conceptual model further identified the objects and activities that comprise each event. Additionally, the experiment designer was interviewed to identify the key factors associated with each activity that would affect experiment outcomes. Objects, activities, and factors are shown for a single event, "Tasking of strike assets," in Figure 3.

The Trailblazer conceptual model also includes detailed process models that described the behavioral rules and command and control relationships.

Detection of SAM Emission
Detection of SAM Launches
Tasking of Sensors
Monitoring and Tracking of SAMs
Imagery Collection
Tasking of Strike Assets
Strike Asset Attack

Figure 2. Key events

Event:	***Tasking of Strike Assets***
Objects:	Joint Air Operations Center Air Assets Ground Assets Naval Assets
Activity:	JAOC tasks strike asset
Factors:	Rules of employment Geography Aircraft speed Weapon range Asset vulnerability

Figure 3. Objects, activities and factors

Finally, the Trailblazer conceptual model defines the scenario for the experiment, including geographic region, specific order of battle, operations orders (OPORDs), and initial locations.

3.2 Lessons Learned

The approach described above has several benefits. First, identification of the key events allowed the federation designers to focus detailed conceptual modeling and, later, federate selection, on those key elements. Similarly, asking the experiment designer, who was an expert in the operational domain, to identify the critical factors also helped the designers to scope the problem. And finally, documentation of this information greatly enhanced communication between the experiment designer and federation designer.

Because there were no standard field or technical manuals that the federation designer could consult for specification of future or hypothetical systems and forces, the experiment designer had to provide these descriptions. In fact, depending on their representations, some simulations required more detailed descriptions than the experiment designer had defined.

A final lesson learned is that it is important to document the changes in the conceptual model from the base case to the alternative, and from excursion to excursion.

4. Design and Develop Federation

Step 3 of the FEDEP is ***Design and Develop Federation***. The FEDEP describes this step as: "Federation participants are determined (if not previously identified), and a federation object model (FOM) is developed to explicitly document information exchange requirements and responsibilities." [1]

For Trailblazer, this involved eight substeps:

- Federate survey
- Federate selection and functional allocation
- Federation policies
- Federation functional design
- Data collection strategy
- FOM design
- Time management design
- Database development

4.1 Federate Survey

First the federates were selected. The key criterion was to select federates that would represent the conceptual model, with a focus on the ability of the federate to model key events and to account for those critical factors identified by the experiment designer.

Service-sponsored simulations were favored both to engender buy-in and to ensure that the basic representations of supporting events were credible. For example, launching of an aircraft could be considered a supporting event in the Trailblazer conceptual model.

Operational Criteria	Technical Criteria	Logistical Criteria
• Ability to support the conceptual model • Ability to support MOEs and MOPs • Credibility • Changeable unit and system behaviors • Explicit representations of command and control	• Number of objects the federate can represent • Duration of run supported by the federate • Previous interfaces to other simulations, including human and hardware in the loop • HLA compliance	• Scenario availability • Ability to run closed loop and faster than real time • Portability • Ability to distribute • Ability to operate at collateral SECRET security level • Availability of developer

Figure 4. Federate selection criteria

If an Air Force-sponsored simulation were to provide that representation, the user of the federation would have confidence that the representation was adequate. Other criteria are depicted in Figure 4.

4.2 Federate Selection and Functional Allocation

Three simulations were selected as the core federates: the Extended Air Defense Simulation (EADSIM), Eagle, and the Naval Simulation System (NSS). A federation management tool was used to monitor and control the federation. The federation is depicted in Figure 5.

Once the federates were selected, the Trailblazer team was formed. The Trailblazer team consisted of the following government and contractor members:

- DMSO and The MITRE Corporation (Washington) as lead and systems engineer
- USA Training and Doctrine Command Analysis Center (TRAC) and SAIC for Eagle
- USAF Electronic Systems Command (ESC), Teledyne Brown Engineering, and The MITRE Corporation (Bedford) for EADSIM
- USN SPAWAR/Metron for NSS

Functionality described in the conceptual model was allocated to the federates based on what service owned a particular system in the real world, the fidelity of the individual federate representations, and federation performance considerations. The functional allocation was:

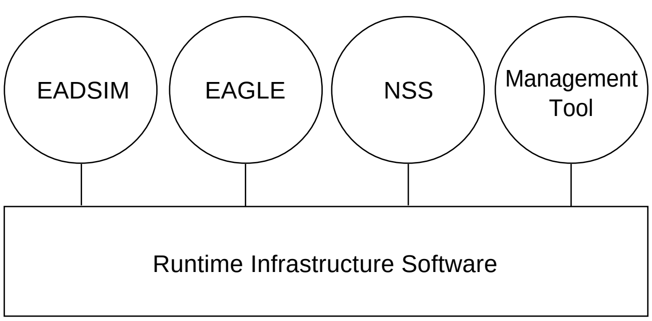

Figure 5. Trailblazer federation

- **Eagle—**red ground clutter; red tactical SAMs; Corps Tactical Operations Center (CTOC); and ATACMS (artillery system).
- **EADSIM—**red strategic SAMs; Air Force strike aircraft; Joint Surveillance Target Attack Radar Systems (JSTARS) (MTI and SAR platform); AWACS (Command and Control and Early Warning aircraft); and Joint Air Operations Center (JAOC).
- **NSS—**aggregate Electronic Intelligence (ELINT); infrared launch detection; and Navy strike aircraft.

4.3 Federation Policies

The federation policies were the agreements reached between the federation teams beyond the specific data exchange defined in the federation object model.

Examples of federation policies for Trailblazer are shown in Figure 6.

4.4 Federation Functional Design

Federate developers documented and designed, as required, their representation of allocated events, objects, activities, and factors defined in the conceptual model. The federate developers documented the key algorithms in pseudo-code. The algorithms were then reviewed by both the experiment designers for suitability to the experiment, and by other federate developers for cross-federation consistency and to tune data exchange requirements.

4.5 Data Collection Strategy

There were three options for collecting data in the Trailblazer federation:

- Put data in the federation object model and collect data with a logger federate,
- Have federates log data locally, or
- Use a combined strategy.

The second option, local logging, was selected.

A data collection plan was developed. For each measure of effectiveness identified by the experiment

- **HLA specification**
  - **Version 1.3**
- **Time management**
  - **All federates (except FMT) constrained and regulated**
- **Units of measure (e.g., distance in kilometers)**
- **Modeling policies**
  - **Which federate computes attrition (owner of killer or victim object)?**
  - **Which federate computes communication delays (sender or receiver)?**

Figure 6. Example federation policies

designer, the plan specified the following:

- Textual description, explanation, implications, and assumptions,
- Algorithms required for reduction,
- Description of each variable in the algorithm,
- Logging responsibility and format for each variable.

The plan also specified file-naming conventions. A manual data correlation and reduction approach was decided upon, both because the scope of the prototype was small and because of budget and schedule constraints.

4.6 Federation Object Model Design

An initial FOM was developed by the systems engineer based on the conceptual model and the allocation of functionality to the federates. The FOM was refined iteratively to meet the federate data exchange requirements. Because a local logging approach was selected for data collection, the FOM only included data required by federates to allow the objects to interact. As a result, not all objects and interactions in the conceptual model were in the FOM.

4.7 Time Management Design

Time management was approached from a federation perspective to ensure temporal consistency. First, a unit of time appropriate for both the experiment and federates was selected; Trailblazer simulated a three-hour scenario using a one-second time unit.

Next, an evaluation was made to determine the frequency that subscribing federates needed updates on reflected objects. In the case of federates that had selectable update rates, the lowest frequency consistent with all subscribers' requirements was selected.

Finally, the federate lookahead values were set so as to increase the degree of parallel federate computation without sacrificing validity.

4.8 Database Development

Databases were developed for each federate that included specific objects, numbers, and locations based on the scenario, object scripts based on experiment operational orders, and parameterized object behaviors and physical characteristics based on the conceptual model. Key features of the databases, such as roads and threat laydowns, were matched.

4.9 Lessons Learned

For experimentation, federates must have explicit and flexible representations of systems, humans, and command and control organizations. Flexibility can be enhanced by interfaces to other simulations, systems, or federates.

Selected federates may require software and/or database changes to accomplish the experiment. While this can be true in any domain, it is particularly likely for joint experimentation because no existing simulation is likely to represent future or hypothetical systems or doctrine.

The federate algorithms must produce the data required to compute measures of effectiveness. It is important for the experiment designer to review these key algorithms to ensure they are appropriate for the experiment and also to ensure that the experiment designer understands how the data he provides will be used in the algorithms.

Development of a federation time management policy should be an explicit part of federation design and should consider both performance and functionality.

Finally, implementation of a scenario in a federation may be the longest lead time item of the entire FEDEP.

5. Federation Integration and Test

Step 4 of the FEDEP is ***Federation Integration and Test***. The FEDEP describes this step as: "All necessary federation implementation activities are performed, and testing is conducted." [1] For Trailblazer, this step included facility planning and setup, federate testing, integration testing, and compliance testing.

5.1 Facility Planning and Setup

The first activity was to identify a facility for integration and execution. Next, federate facility requirements were identified, including requirements for computer makes and models, memory, disk, OS and version, language(s), and compilers. Next all hardware and software were installed, including the DMSO-provided HLA software and tools: runtime infrastructure software (RTI), the object model development tool (OMDT), and the federation execution planner's workbook (FEPW) editor.

As a last step in setting up the facility, the FEDEP security overlay was consulted and facility security requirements and procedures were reviewed.

5.2 Federate and Integration Testing

Prior to bringing all federates together for an integration event, each federate was individually checked to ensure it could perform basic HLA tasks such as join, resign, and time management.

Next, a series of integration events was scheduled with code/re-code time between them. Integration testing was incremental. The first event focused on join, synchronize, object declaration management, and resign. The next event added the advancement of time, and a subset of the object publications, reflections and interactions. New objects and interactions were brought in on a predetermined schedule.

Integration was also end-to-end; it included experiment data collection and reduction. User review of each excursion was an integral part of testing.

Testing aids included the conceptual model, functional design, the data collection plan, the FOM, the FEPW and a test plan derived from the above.

5.3 Compliance Testing

The Trailblazer federate developers chose to be individually tested for HLA compliance. This allowed compliance testing to piggyback on scheduled integration events, and also, by allowing each federate to be stimulated by the rest of the federation, removed the requirement for federate test harnesses.

Because the only existing Trailblazer federation databases were classified, compliance testing occurred in the secure Trailblazer laboratory. Testing was relatively quick: four federates were tested in the span of three mornings.

5.4 Lessons Learned

It is prudent to conduct individual federate testing for each federate prior to integration testing regardless of previous experience with HLA.

Standard operating procedures for executions, such as output file archival and initialization procedures, should be developed and tested as part of integration testing.

It is also important to coordinate federate development plans to facilitate incremental integration testing.

6. Federation Execution

Step 5 of the FEDEP is ***Federation Execution***. The FEDEP describes this step as: "The federation is executed, outputs analyzed, and feedback provided to the federation sponsor." [1]

6.1 Run Execution, Data Collection, and Reduction

An analysis plan was developed that specified the number of runs required for the base case and the alternative, and for each experiment excursion. These runs were executed, and the data was collected and reduced manually using an electronic spreadsheet tool.

6.2 Lessons Learned

The boundary between testing and analysis of execution runs can be indeterminate; testing is not complete until all data is incorporated, all cases are executed, all measures of effectiveness are generated, and the results from each case are analyzed. Therefore, analysis should be performed in near-real time, and it is preferable to maintain team composition from testing to execution.

Interpretation of the results can be facilitated by good documentation of key algorithms and data, and also by participation of the experiment designer and the federate scenario experts.

Trailblazer used a data collection approach that relied on local logging by each federate, but the extra testing required for custom logging, as well as the data correlation issues that arose from multiple log files, caused too many potential points of failure. Future versions of the federation will use FOM-based data collection.

Also, while data reduction was performed manually, it is evident that automated data reduction is needed for large data sets.

7. Summary

7.1 HLA and the FEDEP

In summary, the FEDEP provides a useful framework for federation solutions. The FEDEP should not be interpreted as a linear process because FEDEP activities may be sequential, cyclic, or concurrent. Use of the FEDEP should be tailored to meet specific federation applications.

From a technical perspective, building a federation is not difficult. The existing HLA software, tools, and technical support (help desk, object model template tools, object model library, object model data dictionary system, FEPW editor, FMT) are helpful. Additional tools are available for federation verification and data collection.

The most challenging aspect of federation development is interpretation of requirements and translation of those requirements into a good design. The HLA has made the technical aspects faster, simpler, and easier so that the focus can be on this fundamental aspect of the problem.

7.2 Lessons Learned for Experimentation

The HLA provides the common architecture, tools, and processes to meet new requirements for joint experimentation in a fast and cost-effective manner. Several steps can be taken to further speed development time:

- Select simulations that are HLA-compliant,
- Ensure user participation throughout the FEDEP,
- Provide secure access to classified networks for transfer of documents and conduct of meetings.

The longest lead-time activity may be implementation of the scenario in the simulations. The Trailblazer federation itself has potential as a "persistent" federation that can be augmented, sub-setted, or modified to meet other experimentation requirements.

8. Acknowledgements

The author wishes to acknowledge the contributions of her teammates on the Trailblazer systems engineering team: Mr. James Calpin, Ms. Laura Feinerman, Mr. David Prochnow, and Mr. David Seidel. Additionally, the author acknowledges the other organizations that participated in the Trailblazer project: USA TRAC, SAIC, USAF ESC, Teledyne Brown Engineering, The MITRE Corporation, USN SPAWAR, and METRON.

9. References

[1] DMSO. "Department of Defense High Level Architecture Federation Development and Execution Process (FEDEP) Model." Version 1.3, DRAFT, Dec. 9, 1998.

[2] Joint Staff (J-6). "JV2010: Information Superiority Implementation and Assessment Plan (U)." 30 September 1997 (SECRET).

10. Additional Reading

Feinerman, L., Zabek, A. "Trailblazer: A Federation For Experimentation." Briefing, The MITRE Corporation, McLean, VA, February 1999.

Zabek, A., Feinerman, L., Seidel, D. "Joint Warfighting Program Information Superiority Experiment Trailblazer." *Proceedings of the Fall 1998 Simulation Interoperability Workshop*, Orlando, FL, September 1998.

Feinerman, L., et al. "Design and Execution of a Federation for Experimentation." *Proceedings of the Spring 1999 Simulation Interoperability Workshop*, Orlando, FL, March 1999.

Anita Adams Zabek is a Program Area Manager in the Information Systems and Technology Division at The MITRE Corporation. She is the task leader for the Trailblazer (now Pegasus) Project for DMSO. She is also lead for DMSO Cadre support to the Air Force Distributed Mission Training Program. She has previously supported the JSIMS Program Office and was the task leader for MITRE systems engineering support for the ALSP-based Joint Training Confederation. She was project leader for Corps Battle Simulation ALSP integration at the Jet Propulsion Laboratory, and has been an Operations Research Analyst at the Institute for Defense Analyses. Ms. Zabek has a BS and an ME in Systems Engineering from the University of Virginia.