SYM-AM-23-098

Excerpt from the Proceedings of the Twentieth Annual Acquisition Research Symposium

Acquisition Research: Creating Synergy for Informed Change

May 10–11, 2023

Published: April 30, 2023

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

Disclaimer: The views represented in this report are those of the author and do not reflect the official policy position of the Navy, the Department of Defense, or the federal government.

The research presented in this report was supported by the Acquisition Research Program at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact any of the staff listed on the Acquisition Research Program website (www.acquisitionresearch.net).

A Reference Architecture for a Policy Test Laboratory

Alejandro Salado—is an Associate Professor of systems engineering with the Department of Systems and Industrial Engineering at the University of Arizona, and consults in areas related to enterprise transformation, cultural change of technical teams, systems engineering, and engineering strategy. Alejandro conducts research in problem formulation, design of verification and validation strategies, model-based systems engineering, and engineering education. Before joining academia, he held positions as systems engineer, chief architect, and chief systems engineer in manned and unmanned space systems of up to $1 billion in development cost. He obtained his PhD in Systems Engineering from the Stevens Institute of Technology. [alejandrosalado@arizona.edu]

Hanumanthrao "Rao" Kannan—is an Assistant Professor in the ISEEM department at the University of Alabama in Huntsville. Prior to this, he was an Assistant Professor in the ISE department at Virginia Tech since 2018. Dr. Kannan received a BE in Aeronautical Engineering from Anna University, India, in 2010, an MS in Astronautical Engineering from the University of Southern California in 2011, and a PhD in Aerospace Engineering from Iowa State in 2015. He then worked as a Postdoctoral Research Associate at Iowa State and Virginia Tech. His research focuses on formalization of Systems Engineering by leveraging disciplines including Decision Analysis, formal philosophy, and engineering design. [hk0049@uah.edu]

Zoe Szajnfarber—is the Professor and Chair of the Engineering Management and Systems Engineering Department at the George Washington University. She studies the design and development of complex systems, primarily in the aerospace and defense sectors. Dr. Szajnfarber holds a PhD in Engineering Systems and dual SM degrees in Aeronautics & Astronautics and Technology Policy, all from Massachusetts Institute of Technology, and a BASc degree in Engineering Science from the University of Toronto. Outside of Academia, she has worked as a systems engineer at Dynacon and MDA Space Systems, and a technology and innovation policy advisor at European Space Agency and NASA. [zszajnfa@gwu.edu]

William B. Rouse—is Research Professor in the McCourt School of Public Policy at Georgetown University, and Professor Emeritus and former Chair of the School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research focuses on mathematical and computational modeling for policy design and analysis in complex public–private systems, with particular emphasis on health care, higher education, transportation, and national security. He is a member of the National Academy of Engineering and fellow of IEEE, INCOSE, INFORMS, and HFES. Rouse received his BS from the University of Rhode Island, and his SM and PhD from MIT. [wr268@georgetown.edu]

Young-Jun Son—is the Head and Professor of School of Industrial Engineering at Purdue University. Prior to this position, he was the Department Head and Professor of Systems and Industrial Engineering at University of Arizona. His research focuses on a data-driven, multi-scale, simulation and decision model for various applications, including manufacturing enterprise, homeland security, and social network. He has authored over 110 journal papers and 100 conference papers. He is a Fellow of IISE and has received the SME 2004 Outstanding Young ME Award, the IISE 2005 Outstanding Young IE Award, and the IISE Annual Meeting Best Paper Awards. [yjson@purdue.edu]

Nirav Merchant—received a BS in industrial engineering from the University of Pune and an MS in systems and industrial engineering from The University of Arizona. He is the Co-PI for NSF CyVerse and NSF Jetstream. Over the last two decades, his research has been directed toward developing scalable computational platforms for supporting open science and open innovation, with emphasis on improving research productivity for geographically distributed interdisciplinary teams. His research interests include data science literacy, large-scale data management platforms, data delivery technologies, managed sensor and mobile platforms for health interventions, workforce development, and project-based learning. [nirav@arizona.edu]

Abstract

The government has identified several obstacles to inform effective and efficient acquisition policies. Effective modeling, simulation, and analysis of acquisition policies require a multi-domain, multi-scale approach. However, existing research in acquisition policy analysis has primarily remained siloed. Policy researchers lack a platform that enables sharing, reusing, or integrating the methods, models, and data developed and/or generated by different research teams in different projects. Government envisions a Policy Test Laboratory (PTL) as a potential solution to this need. The PTL is conceived as a service where a domain model developed in a project can be used and/or integrated with another model of a different domain developed in a different project. This paper presents a reference architecture for the PTL, defined as a set of guidelines and constraints that will enable (1) the sharing and use across acquisition research projects of data, models, and tools, and (2) the construction and composition of multi-disciplinary models of government acquisition, that addresses both technical and governing aspects.

Introduction

The government has identified several obstacles to inform effective and efficient acquisition policies. The defense budget serves many purposes, with many stakeholders. This could lead to inherent conflicting objectives. For example, socioeconomic objectives, including free and fair competition for taxpayer money, can be at odds with the most expedient means to achieve military objectives. We suggest that this complex context results in NDAA, statutes, requirements, etc., that are driven by an overreliance on process metrics because of an inability to define outcome metrics.

In our experience, effective modeling, simulation, and analysis of acquisition policies require a multi-domain, multi-scale approach. Among others, informing a policy decision requires understanding not only financial implications, market reactions, supply chain availability, resulting technical capabilities, societal impacts, and effects on national security, which requires assessing how they relate to each other. However, existing research in acquisition policy analysis has primarily remained siloed to the best of our knowledge. Policy researchers lack a platform that enables sharing, reusing, or integrating the methods, models, and data developed and/or generated by different research teams in different projects.

The Acquisition Innovation Research Center (AIRC) has envisioned a Policy Test Laboratory (PTL) as a potential solution to this need. The PTL is conceived as a service where a domain model developed in a project can be used and/or integrated with another model of a different domain developed in a different project. In this sense, the PTL is not necessarily a unique simulator or aggregated model. While it could be implemented in such a way, non-monolithic implementations are also considered.

This paper presents an initial reference architecture to support the development of the PTL. The reference architecture defines a set of guidelines and constraints that enable (1) the sharing and use across acquisition research projects of data, models, and tools, and (2) the construction and composition of multi-disciplinary models relevant to government acquisition policy research questions. In essence, the PTL's reference architecture is intended to guide research teams in developing models, gathering data, and performing simulations in different domains so that they can be reused and integrated by others.

Background

This section provides the results of an initial assessment of the characteristics, scope, drivers, and main capabilities of existing efforts in other domains that have attempted or are attempting to integrate models and data across disciplinary boundaries. The effort allocated to identify and assess existing architectures and/or frameworks was timeboxed. This section is not aimed at being comprehensive but rather exploratory.

Identification was performed by aggregating frameworks and architectures already known to the researchers, as well as by a quick online search. Assessment was performed based on publicly available documentation and/or conversations with some of the people involved in the architecture or framework being assessed. In line with the exploratory spirit, the activity was not intended to necessarily achieve an accurate characterization of existing work. Therefore, it is recognized that there may be some inaccuracies in the information provided in this section. Nevertheless, the information is still considered relevant and useful for the purpose of informing the developing of the initial reference architecture for the PTL.

Nine frameworks were assessed: the MIT Joint Program on the Science and Policy of Global Change,[1] the IEEE Std 1516-2010 (IEEE, 2010), the Multi-level Modeling framework (Rouse, 2019, 2022), CyVerse,[2] the National Socio-Environmental Synthesis Center (SESYNC),[3] the Simulation Framework (CSF; (Haynes et al., 2003; Singh & Mathirajan, 2014), the One Semiautomated Force (OneSAF; Parsons et al., 2005), the Modeling Architecture for Technology, Research, and Experimentation (MATREX; Hurt et al., 2006), and the OpenGMS.[4]

Each framework was assessed in the following attributes, nothing that for some frameworks some of this information might have not been available or identified during the activity: (1) Background, goal, and maturity (or state of the development) of the framework/architecture; (2) Types of research questions it is intended to support, including application domains it serves; (3) Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…); (4)Architectural aspects, such as layers, components, integration, relationship between parts, services it provides, etc.; (5) Technical governance, including maintenance and, if possible, rough estimate of effort; and (6) Organizational governance, including maintenance and, if possible, rough estimate of effort.

Existing frameworks display a wide variety of approaches to establish frameworks that enable the integration of models across disciplinary boundaries. There seem to be three main trends in establishing these frameworks:

Structural frameworks: These frameworks provide structure and guidelines that enable reuse and integration of models but do not provide any integrated model. These are independent of research question. Details of how integration occurs are left open for the different modeling actors to define. These frameworks are generally established through working groups or standards bodies.

Top-down frameworks: These frameworks are constructed around a research question. An integrated overarching model is constructed, even if not at once. Using and contributing the model requires evaluation and approval of a governing body that oversees the growth of the model. Answering research questions requires interacting with all or part of the integrated model. As a result, deployment requires a substantive portion of the integrated model to be constructed before it can be used. This, together with the extensive oversight required to maintain the model, leads to high upfront and sustainment investments.

Bottom-up frameworks: Similar to structural frameworks in the sense that a structure to enable integration is provided, but additional guidance and infrastructure are provided to

[1] https://globalchange.mit.edu/
[2] https://cyverse.org/
[3] https://www.sesync.org/
[4] https://geomodeling.njnu.edu.cn/

integrate models around a class of research questions. These frameworks often rely on open source and open access artifacts, as well as a decentralized contribution from researchers, which reduces the investment needs to deploy and sustain the resulting models.

MIT Joint Program on Science and Policy of Global Change

The MIT Joint Program on the Science and Policy of Global Change has the mission of "advancing a sustainable, prosperous world through actionable, scientific analysis of the complex interactions among co-evolving, interconnected global systems." Founded in 1991, the "program" has pursued research that enables decision-makers to answer policy questions related to sustainability. Specifically: environmental protection, economic viability, and social equity. It has always been a collaboration mainly between earth scientists and economists and specializes in "integrated assessments." It is a research program in that it houses a mix of faculty, research scientists, and graduate assistants (at multiple levels, but weighted the technology and policy program masters students). It received anchor funding from the Department of Energy and also works with a consortium of sponsors. Over a history, it has done a mix of inquiry-driven development vs. infrastructure development; that balance has shifted over time.

The program was designed to provide relatively quick comprehensive analysis to support decision-making on global and climate relevant policy questions. Their work in seven focus areas. The most relevant to this project is the policy scenarios.

Most of the Joint Program's work leverages the Integrated Global System Modeling (IGSM). It has two interacting components: (1) The Economic Projection and Policy Analysis (EPPA) model (a computable general equilibrium model from economics) and (2) the MIT Earth System model (MESM; from atmospheric science). Both include discipline-specific models of the "physics" of the relevant system. EPPA draws on trade data that was curated over decades.

A version of each of these models existed at the time when the program was founded. Since then, most of the new work has focused on building additional resolution in segments of the economy of the earth system when they are needed to answer a specific policy question. For some specific purposes, new models are developed that use different data sets or aggregate sectors differently.

For the first 20 years of the program, technical development was led by one key research scientist. He worked with every student contributing a module and retained authority to include a new module into the live EPPA instance. Most new technical tasks focus on "building-out" a specific relevant module. Before it is integrated into the overall model, would take responsibility for V&V. As the program has grown, there are a few more technical leads, but the group is still small, and technical governance is centralized. Their approach has been quite centralized too, in that there has generally been a Director/PI for each of the economic and earth systems sides, with a few senior research staff and a lot of student research assistants. They collaborate through weekly lunch tag ups where the RAs got to watch the discussions of the senior folks about what work to prioritize. Even though the effort is highly problem-driven making external stakeholders were important, the team retains a strong emphasis on the overall goal of developing "this global modeling competency." This has led to a lot of co-creating of the intersection of model extensions to support groups of pressing questions.

IEEE Std 1516-2010

The IEEE Std 1516-2010 describes the framework and rules of the High Level Architecture (HLA), which is an integrated approach to provide a common architecture for federated simulations. HLA was initially developed under the leadership of the U.S. Department of Defense in the mid 1990s. In 1998, the Defense Modeling and Simulation Office (DMSO) released HLA 1.3, an official document of HLA. The second version (HLA-2000) and third version (HLA-2010) were then further refined and published by IEEE. The latest version (HLA 1516-20XX; HLA 4) is currently developed by Simulation Interoperability Standards Organization (SISO).

The goal of HLA was to assure interoperability and reusability of defense models and simulations (training, analysis, and control) (original goal), which was extended to broader applications (e.g., manufacturing/supply chain management, health care, infrastructure, and more). It enables us to connect simulations running on different computers, locally or widely distributed, independent of their operating system and implementation language, into one federation. Maturity: Run-time Infrastructure (RTI) is major component of HLA, and a software that provides a standardized set of services, as specified in the HLA interface specification. In the past two decades, multiple RTIs have been developed as an open source (e.g., http://porticoproject.org), by a commercial sector (e.g., MAK Technologies), or research team projects (web services).

HLA has been used to address interoperability and reusability of defense models (e.g. DoD projects), development of supply chain network simulation, integrating geographically dispersed member simulations (e.g., National Institute of Standards and Technology and Boeing), and a city-level traffic simulation (the Federal Highway Administration). In these projects, researchers and practitioners used HLA to integrate a mix of the following elements: (1) system dynamics (aggregate level), (2) discrete event (process flows), (3) agent based (decision making, communications), (4) dynamic systems or physics-based game engine, (5) hardware (e.g., robots, machines, drones; simulations running in real-time), and (6) human (simulations running in real-time).

Following a publish and subscribe architecture, HLA can be applicable to various types of operating systems, software, applications, and languages. For example, it allows integration of wide ranges of software: AnyLogic, Simio, Arena, ProModel, Repast, DynusT (traffic simulator), hardware (robots, machines, drones), Unity (game engine), and more.

To maintain or govern models, an open source Portico (http://porticoproject.org) or a commercial RTI (e.g., MAK Technologies) can be used, and efforts are needed to develop technical governance. In addition, a governance structure and agreement need to be established among sponsors and users.

Multi-Level Modeling

The Multi-Level Modeling approach to modeling represents an enterprise or an ecosystem at four levels of abstraction: people, processes, organizations, and society. The levels are typically represented by agent-based models (people) discrete event or network process flow models (processes), microeconomic models of decision making (organizations), and macroeconomic models of policies (society). This framework has been in use, and continually refined, for over 10 years, addressing research questions related to economics of scaling clinical trials to broader use (Emory, Indiana, Penn, Vanderbilt), likely impacts and efficacy of health policies (ACA, CMS), and impacts of incentives on consumer energy behaviors (Accenture, GM).

The engagement of sponsors and subject matter experts is central to this approach. Such dependency makes scheduling and conducting meetings a challenge. The approach is not very adaptable, and models are difficult to update once the sponsor’s questions have been answered. As such, the models are not necessarily maintained or governed. Instead, each new question demands the development of new models, which require an investment in order of $200,000–300,000 for familiar domains and $500,000–1,000,000 for new domains.

CyVerse

CyVerse provides scientists with powerful platform to handle huge datasets and complex analyses, thus enabling data-driven discovery. CyVerse offers extensible platforms that provide data storage, bioinformatics tools, data visualization, interactive analyses, cloud services, and APIs, among others, with the purpose of transforming science through data-driven discovery. It is conceived as a federated platform for enabling diverse teams to collaboratively develop and share solutions for data driven questions, and support analyses that need domain specific models and machine learning workloads. Current applications range from astronomy to Earth sciences to hydrology, traffic engineering, and life sciences.

CyVerse is built on a layered architecture that abstracts data storage and execution environments. Data management is driven by metadata, remaining agnostic of the physical storage provider (which can be on the cloud, private premises, etc.). Access to the execution environment is secure with federated sign on. Layers are connected through automation using APIs and the end user facing applications are customized for specific purposes through web interfaces. This allows for developing methods and securing sharing underlying tools/pipelines and data without needing software installation on client side.

Operationally, CyVerse has a public deployment and the capability to be deployed privately at different organizations. The public deployment is maintained by the University of Arizona, and it can be integrated with private infrastructure.

SESYNC

The SESYNC, established in 2011, brings together the science of the natural world with the science of social systems and decision making to solve problems at the human-environment interface. SESYNC has accelerated research and learning that seeks to understand the structure, functioning, and sustainability of coupled social and environmental systems. This is achieved by enhancing teams' and individual participants' capacities and skills to bridge varying epistemologies, methods, and approaches. SESYNC has supported over 340 projects, engaging over 4,700 researchers in over 70 countries. Its research output accounts for over 750 peer-reviewed publications.

SESYNC research relies on many different forms of information (data collected by quantifying an event or outcome, running a computer simulation, collecting photographs, transcribing interviews, or capturing social media activity), highly heterogenous data, and synthesis and analysis methods (systematic literature reviews, meta-analyses, expert elicitation, statistical and spatially explicit modeling, system dynamics, and agent-based modeling).

The SESYNC builds upon a decentralized infrastructure of several dedicated software and tools. In terms of organizational governance, all products developed under SESYNC-sponsored activities are made accessible with no restrictions for use and dissemination through FTP or code repository services.

CSF

The CSF was initiated by the U.S. Army Aviation and Missile Research, Development, and Engineering Center (AMRDEC) in 1999 as a standardized structure to support dynamic simulations. The original intent for the framework was to be domain agnostic, but it evolved as a specific toolkit to support modeling and simulation of tactical missile systems. It supports both discrete event simulation and differential equations. It supports simulation of missile deployment, 6 DOF Propulsion Aerodynamics Controls and Kinematics module, and hardware-in-the-loop testing, with both real-time and non-real-time capabilities. The framework is flexible and supports various models, data, and tools, with a common library approach and C++ implementation. It has a GUI for model composition and allows for plug-ins.

OneSAF

The OneSAF is intended to foster interoperability and reuse across modeling and simulation communities of the Army. The framework supports the development of advanced concepts for doctrine and tactics, training of unit commanders and staffs, development of new weapon systems, and production of data as input to other simulations. Its applications include testing algorithms for real C4I systems, modeling WWII tank combat, creating a "cyber range" for cyber warfare analysis and training, and virtual training on operating construction equipment. The framework supports physics-based models for platforms, soldiers, equipment, logistical supplies, communications systems and networks, emerging threats, and aviation assets. It has a layered approach with components linked into a common executable, and data exchange occurs through method calls. However, the framework has no inherent mechanism to enforce assumptions and dependencies of a component if used in a different context, and the validity of the composed system is up to the developer.

MATREX

The purpose of MATREX is to develop a composable Battle Command-centric modeling and simulation MS environment consisting of multi-fidelity models, simulations and tools that are integrated and mapped to a Future ForceBlended Force architecture for use across the acquisition spectrum, specifically integrating live, virtual, and constructive models at the entity or engineering level. MATREX is not limited to a specific application and can be extended, serving as a support system for various types of research questions and application domains, including modeling command and control, communications actions and effects, and network centric warfare systems. The framework supports the integration of different disciplinary models, data, and tools, including OneSAF, Aviation Mobility Server, Countermine Server, Missile Server, and others. The architecture of MATREX includes a layered approach with three layers - Federates, Middleware, and Distributed execution infrastructure. Technical governance and organizational governance have not been assessed.

OpenGMS

OpenGMS supports open web-distributed sharing of modeling and simulation resources for geographic applications by providing a virtual community for collaboration among researchers from various domains. The models are heterogeneous, both in terms of domain of application and scale.

OpenGMS uses a layered architecture with four layers: Model repository, Data repository, Models as a service, and Thematic center. The model repository collects model resources to build a dictionary where all models (also include related tools, algorithms, etc.) are organized in a formal way. Users can find a model with its detailed information, conceptual and logical descriptions, computable resources, developing history, and applications. This repository publishes model resources under the permission of the author. The data repository collects data resources to build a community where users can explore modeling-related data through a universal center. Users can share their data resources to the data repository. Various data sharing sites can be also linked to support users so that they do not visit individual sites. This data repository publishes data and their related information under the permission of the author. This platform provides model, data, and computing resources as corresponding services in an open web environment. Users can setup input data and run a model via a web client, and the related model will be executed in a remote computer node. Users can invoke a model service before boarding and obtain results when get off. A set of alternative solutions are available to convert original models as reusable services, to publish data files as reusable services and to share computers as available services. Several thematic centers are constructed to help researchers collect models, data and other related resources. Topic-related or problem-related resources could be easily discovered within a thematic center.

Reference Architecture

Use Cases

Three main use cases were used to inform the development of the reference architecture:

1) The government has a policy question that could be answered using the PTL. Example: *How should investments in acquisition supply chains be managed across mission areas with highly uncertain demands?*
2) A (policy) researcher wants to leverage the PTL. Example: *How can a STEM policy found to be successful in a pilot study in one state, best be scaled to provide benefits to all states?*
3) A researcher wants to integrate their work in the PTL. Example: *How can a large data set on technology innovations in sensors and semiconductors be imported to the PTL for access and use by other researchers?*

Major Drivers

Success of the PTL requires two key contributions: researchers that use and contribute to the PTL, and a sponsor that trusts the results generated with the PTL. Having in mind that the needs of the sponsor will change over time, as well as the science, models, methods, and tools used by researchers, the ability to seamlessly evolve the PTL is likely to be instrumental for its own sustainment. Therefore, the major drivers that informed the development of the reference architecture were sponsor trust, researcher adoption, and evolutionary needs.

Researcher adoption is likely to be driven by two questions:

(1) As a researcher, why should I use the PTL?

(2) As a researcher, why should I make an extra effort to make my models, data, and methods reusable by other researchers and interoperable with other models, data, and methods that I do not plan to make use of?

Addressing these may require incorporating provisions for establishing incentives in the reference architecture.

Gaining the trust of the sponsors to use the results provided by the PTL will likely depend on several factors. There is abundant literature on this topic, but it was not possible to explore it in detail as part of the sponsored project due to time constraints. Instead, the team started off their own experience in working with sponsors in the context of modeling. A summary of factors leading to trusting different aspects of the modeling effort are summarized in *Table 1*. In addition, it is noted that trust between the modeler and the stakeholders takes time to build and the path to build such trust depends on the type of relationship between them.

Table 1. Informal model of sponsor trust

Who/What am I trusting?	**Modeler (track record of interacting with stakeholder or reputation)**	**Model (previously used/accepted or careful V&V in this context)**	**Inputs (provenance, e.g., censes, vs. careful look at representativeness for this application)**
Validity (solve my problem)	Gut of senior stakeholder	Classic model V*V Depends on generation (block 1 different than n)	Good data vs. right data for this application
Acceptability (in ways I prefer)	Comfort/confidence in understanding (and the way they talk to me)	Type of models used (understand representation, e.g., pde vs. econometrics) Explainability	Support credibility of the data (available in community and has been vetted by experts)

Viability (worth my time to learn how to do)	Cost-effort to work with expert vs. use their tool (either learning to work together or learning the tool)	Effort to develop my own comfort (learning curve)	Effort to compile/clean. Proprietary/classified/expensive?

Trust in the concept of the PTL adds an additional dimension; that of trusting the integration of models and data. Stakeholders do not only need to trust the individual components that form the integrated models, but also the integration process of the models and the resulting integrated model. Furthermore, whereas some models may have been considered valid, acceptable, and/or viable on their own by their dedicated stakeholders, these assessments may need to be revisited in the context of the integrated model and the new stakeholders. Assuming a bottom up PTL, as discussed earlier, stakeholders may not even have access to the modelers that modeled some of the components of the integrated model, which furthers hinders trust. Transparency and clarity on model usage may likely be a key aspect the reference architecture must facilitate.

Facilitating the evolution of the PTL with respect to research questions, scientific discoveries, modeling frameworks, novel methods, etc., results in some development challenges. While the reference architecture can provide flexibility, evolution cannot be unbounded. In fact, guidelines and bounded actions may be necessary to guarantee that existing models and data do not inadvertently become not usable due to the evolution. In other words, it is likely that the reference architecture does not simply facilitate evolution but that it guides it to maintain relevance, validity, and acceptability of the artifacts it possesses at the time of the evolution.

The ability to compose, in varying combinations, simulation components (e.g., models, applications, etc.) into simulation systems to satisfy specific user requirements. The defining characteristic of composability is that different simulation systems can be composed in a variety of ways, each suited to some distinct purpose, and the different possible compositions will be usefully valid. Composability is more than just the ability to assemble simulations from parts; it is the ability to combine and recombine, to configure and reconfigure, sets of components into different simulation systems to meet different needs (Petty & Weisel, 2019).

Furthermore, the different artifacts in the PTL must allow for model composability to enable integrating heterogeneous models. Model composability can have two interpretations (although both are needed for a valid composition): (1) Syntactic composability and (2) Semantic composability. Syntactic composability deals with the actual implementation aspects of model composition, where the focus is on the implementation details such as parameter passing mechanisms, external data accesses, and timing assumptions. This strives to ensure that the composed models are compatible for all of the different configurations that might be composed. In contrast, semantic composability is a question of whether the models that make up the composed simulation system can be meaningfully composed (i.e., if their combined computation is semantically valid). Even if the components can be composed syntactically, the models may or may not be composable semantically. Since one of the critical attributes of a simulation system is the degree of reorganizability, to answer a wide range of questions, semantic composability is a more appropriate notion. Note that syntactic composability is a necessary but insufficient condition for semantic composability.

Model composability requires metadata associated to each model and may be facilitated by certain tenets of the framework in which composability occurs. Desired model metadata that are required to enable composability include, among others (Petty & Weisel, 2019):

- **Nature:** assumptions, spatial and temporal resolution, boundary conditions, range of validity, inputs and outputs, details about model interpretations, etc.
- **Tools/Technology:** software, implementation language, operating system, compiler version, tools, etc.
- **Interfaces:** syntax, data definitions, standards
- **Applications:** run modes, performance, intended uses
- **Provenance:** developers, prior uses, validation history

Useful characteristics for a framework that can facilitate implementation of composability include, among others (Petty & Weisel, 2019):

- Dynamic model registration and discovery, supported by a directory (or directories) of registered models and repositories.
- Semantic query, search, and reasoning capabilities for model selection, supported by model specifications (i.e., metadata).
- Distributed processing across multiple platforms, systems, services, and domains.
- Support for intelligent and polymorphic proxies for models.
- Automated composition processes to combine models.
- Virtual repositories that include version control.
- Ability to save compositions and composition templates.
- Compliance with relevant standards.
- Software authentication and information exchange services.

Architecture

The reference architecture for the PTL is defined as a set of guidelines and constraints that will enable (1) the sharing and use across acquisition research projects of data, models, and tools, and (2) the construction and composition of multi-disciplinary models of government acquisition, that addresses both technical and governing aspects.

A layered reference architecture is proposed (Figure 1). The Application layer handles aspects related to how organizations and infrastructure engage (e.g., security aspects or UI/UX). The Problem class/Research question layer handles aspects related to assessing if a given task can be supported by the PTL (as an integrated assessment tool). The Models, Data, Tools layer handles the actual research artifacts indicated by their names. The Infrastructure layer handles all aspects related to hosting, storing, and exchanging the research artifacts with the PTL consumers.

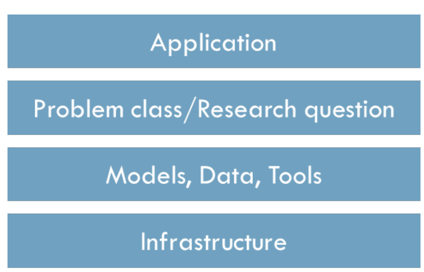

Figure 1. Reference architecture

This layered architecture allows for PTL designs that can embed the useful characteristics to facilitate model composability listed earlier. For example:

- Dynamic model registration and discovery, supported by a directory (or directories) of registered models and repositories. → Through the *Application Layer*, a user can query the *Problem Class/Research Question Layer*, which accesses the directory of models in the *Models, Data, and Tools Layer*.

- Semantic query, search, and reasoning capabilities for model selection, supported by model specifications (i.e., metadata). → Through the *Application Layer*, a user can semantically query the *Problem Class/Research Question Layer*, which accesses the directory of models and their metadata in the *Models, Data, and Tools Layer*.
- Distributed processing across multiple platforms, systems, services, and domains. → The *Infrastructure Layer* can be implemented as a distributed platform.

Specific choices of what characteristics to implement are left to specific PTL designs.

The next subsections provide further details and discussions for each layer in the reference architecture. Each layer is elaborated with a different depth, based on prioritizations made by the research team in the scope of the sponsored project leading to this paper.

Application Layer

The Application layers provide a framework for the engagement of the different actors and the PTL. Three actors have been identified: researcher, sponsor, and the AIRC. Anticipated engagements are depicted in Figure 3. Note that, while the application layer is defined in the context of the tasks performed by the different actors, the application layer does not include the tasks but provides the means to the different actors to interact with the PTL to execute those tasks.

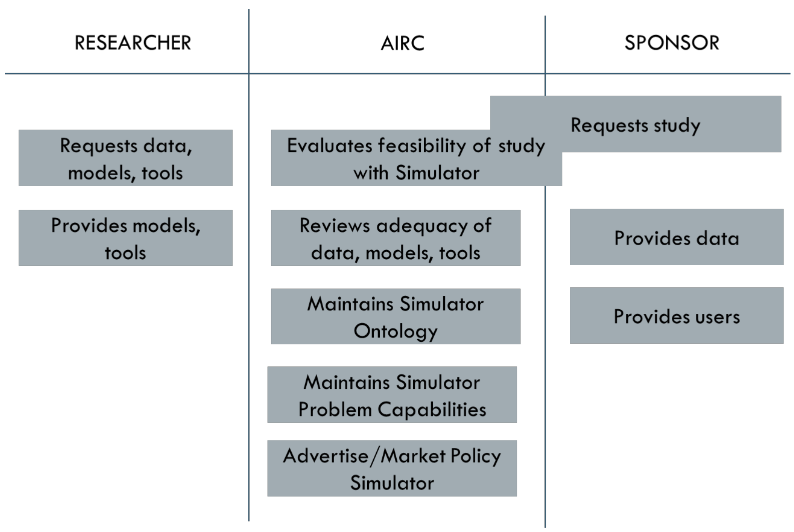

Figure 2. Engagements Between PTL actors

Researchers are anticipated to contribute with their models, data, and tools to build the PTL. Essentially, models, data, and tools resulting from their research projects will be fed into the PTL. At the same time, researchers are anticipated to be consumers or users of the data, models, and tools already available in the PTL. This is, in fact, the purpose of the PTL: a researcher can make use of models, data, and tools already in the PTL to conduct cross-disciplinary research. The application layer handles the exchange of requests and data, models, and tool exchanges between the researcher and the PTL, including aspects related to UX/UI, security, and access restrictions, among others. The extent to which the reference architecture should constraint these aspects needs to be addressed in future work.

Sponsors are anticipated to feed the PTL with data to support research and be the consumers and users of the results generated by the PTL. Furthermore, sponsors are also expected to initiate most of the research supported by the PTL. This is expected to be done in tandem with the AIRC team, which will possess the detailed knowledge of what research questions could the PTL support. As for researchers, the application layer handles these exchanges between the sponsors and the PTL, including aspects related to UX/UI security, and access restrictions, among others. The extent to which the reference architecture should constraint these aspects needs to be addressed in future work.

The AIRC is anticipated to act as the governing body of the PTL, undertaking the activities associated with sustaining the PTL and supporting its operations. On the technical front, because, as described earlier, the PTL is expected to be developed bottom-up, the adequacy of the models, data, and tools injected by researchers and/or suppliers into the PTL should be assessed for conformance to guarantee future integration efforts. Furthermore, since some of these models, data, and tools may incorporate new aspects not previously addressed by the PTL, its ontology and evolving capabilities will also need to be maintained. On the programmatic front, the AIRC is anticipated to jointly work and support its sponsors in assessing the feasibility and adequacy of the PTL to support desired research questions, as well as to advertise the PTL's capabilities to reach to a wide variety of researchers and sponsors that could benefit from them. As for researchers and sponsors, the application layer handles these exchanges between the AIRC and the PTL, including aspects related to UX/UI. The extent to which the reference architecture should constraint these aspects needs to be addressed in future work.

Problem Class/Research Question Layer

The Problem Class/Research Question layer provides the necessary services to characterize the artifacts in the PTL in the context of trust. Particularly, these services evaluate the information contained in the different PTL's artifacts to determine the questions or problems that the PTL can support and the level of confidence to be expected in such support. This can be thought of as the identification of capabilities enabled by the models, data, and tools in the PTL; this includes those already existing and those that may be created during the research.

While a more in-depth assessment is necessary, this layer handles taxonomical aspects important to trust such as:

- *Scale*: it indicates the context in which the model, data, and/or tools have been used (e.g., from a successful prototype to a large-scale application).
- *Projection of tipping points*: it indicates the likelihood of achieving change (e.g., confidence on organizational or social change)
- *Risk assessment*: it indicates risks associated with using the different artifacts in the PTL for a particular problem class or research question.
- *Control mechanisms*: it indicates the interoperability of models, data, and/or tools with respect to a specific problem class or research question.

While traditional concepts, methods, and tools for model verification and validation are likely to be adopted in this layer, novel methods to forecast and execute verification and validation of integrated (heterogeneous) models may need to be developed. These will refine the constraints imposed in the metadata to be provided with every model, dataset, and method that is fed to and/or used together with the PTL.

Models, Data, and Tools Layer

This layer represents the models, data, and tools in the PTL. It encompasses the artifacts and their associated metadata or ancillary information, which include the information necessary to (1) integrate each model, data, and/or tool with other models, data, and/or tools, and (2) assess the confidence level or trust in the artifacts. The layer implements control mechanisms to guarantee that every artifact in the PTL conforms with pre-defined requirements for such metadata, ancillary information, and confidence characterization.

Several model metadata standards and/or protocols to enable model integrability are in use in other fields. Two examples are presented below, the Open Modelling Interface (OpenMI) (Moore & Tindall, 2005) and the ODD (Overview, Design concepts, Details) Protocol (Grimm et al., 2010).

In the field of geospatial information modeling, the OpenMI was defined with the goal to "bring about interoperability between independently developed modelling components, where those components may originate from any discipline or supplier" (Moore & Tindall, 2005). The standard is over 100 pages long and formally defined through schemas in UML. Its coverage is comprehensive, including requirements on model elements, interfaces, values, and linking capabilities and/or protocols. UX/UI are also covered with templates to document the metamodel and its conformance to the standard. The standard is very specific to geospatial modeling, so it cannot be directly reused for the PTL. However, it provides a good indication of the kind of effort that goes into defining a modeling interface for acquisition research.

The ODD Protocol has a narrower scope, focusing on fully defining agent-based models (Grimm et al., 2010). It requires every model to incorporate details of general nature (purpose, entities, state variables, scales, and process overview and scheduling), design concepts (basic principles, emergence, adaptation, objectives, learning, prediction, sensing, interaction, stochasticity, collectives, observation), and details of the model (initialization, input data, and sub-models).

The same applies to metadata standards. An example is the FAIR Data Standard (Wilkinson et al., 2016). FAIR provides a set of principles that have the goal to improve the Findability, Accessibility, Interoperability, and Reuse of digital assets, with an emphasis on machine-actionability:

- Findable:
    - F1. (Meta)data are assigned a globally unique and persistent identifier.
    - F2. Data are described with rich metadata (ref. to R1 below).
    - F3. Metadata clearly and explicitly include the identifier of the data they describe.
    - F4. (Meta)data are registered or indexed in a searchable resource.
- Accessible
    - A1. (Meta)data are retrievable by their identifier using a standardized communications protocol.
        - A1.1 The protocol is open, free, and universally implementable.
        - A1.2 The protocol allows for an authentication and authorization procedure where necessary.
    - A2. Metadata are accessible, even when the data are no longer available.
- Interoperable
    - I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
    - I2. (Meta)data use vocabularies that follow FAIR principles.
    - I3. (Meta)data include qualified references to other (meta)data.
- Reusable
    - R1. (Meta)data are richly described with a plurality of accurate and relevant attributes.
        - R1.1 (Meta)data are released with a clear and accessible data usage license.

- R1.2 (Meta)data are associated with detailed provenance.
- R1.3 (Meta)data meet domain-relevant community standards.

Detailed descriptions of the principles and several implementation examples are publicly available. There are also tools available to support the generation of metadata that guarantee abiding to some of the FAIR principles.

Furthermore, this layer also incorporates two mechanisms that provide an underlying structure to foster internal consistency between the artifacts in the PTL:

(1) An acquisition ontology. In line with some of the requirements and principles identified earlier, an ontology will establish common understanding and interpretation of acquisition concepts, avoiding terminological and conceptual conflicts between models, as well as redundancies.

(2) (Potentially) A hetero-functional graph (Schoonenberg et al., 2019). Building upon the ontology, a hetero-functional graph provides a mathematical structure to integrate heterogeneous models. While this still needs some investigation, hetero-functional graphs may provide valuable capabilities to assess confidence and trust resulting from such integrations.

Infrastructure Layer

The Infrastructure Layer hosts all the artifacts of the PTL. It can be thought of as a repository containing models, datasets, and tools, as well as the tools that implement the different layers of the PTL.

Three major alternatives have been identified:

- Use an *existing infrastructure*, such as CyVerse. This alternative usually requires the minimum upfront development effort but might provide insufficient security protection for certain datasets.
- Use a *custom, centralized infrastructure*. In this case, AIRC would develop and maintain the repository. This option offers the maximum flexibility to satisfy sponsors hosting needs but likely requires a significant upfront development effort.
- Use a *decentralized approach*, where each researcher must host the artifacts that they develop and provide PTL users with access to them, both within requirements set forth by the AIRC.

The reference architecture does not need to constraint the implementation of the PTL to any particular alternative. The selection may be done in the context of the PTL design.

Conclusions

An initial reference architecture to support the development of a PTL has been presented. The reference architecture consists of four layers that are aimed at enabling the sharing and use across acquisition research projects of data, models, and tools, and the construction and composition of multidisciplinary models of government acquisition, that addresses both technical and governing aspects.

A PTL could be purposed to support a suite of activities aimed at answering a wide diversity of policy questions or to center on a type of policy problem and focus on building test range infrastructure over time. In the first option, the extent of reuse is mostly data and generic modeling strategies, standards, and best practices. In the second option, reuse goes beyond data and standards; it requires a core set of models that can be quickly customized for specific questions. The proposed reference architecture is intended to support both kinds of developments, particularly promoting the organic growth of the PTL as data, models, and results different projects become available and injected into the PTL.

In fact, given a lack of clarity on the existence of commonality or agreements related to modeling in acquisition-related research, a bottom-up implementation approach is suggested initially. The basic idea consists of, first, not constraining the work of acquisition policy researchers to specific models, tools, or modeling approaches. Instead, researchers are requested to deliver a set of artifacts (and metadata) associated with the models, datasets, and tools they generate during their project. These are then consolidated and aggregated by a dedicated team, resulting in a PTL that will grow larger, more mature, and more capable with every new acquisition research project. As the PTL matures, sponsors could incorporate additional constraints to be met by researchers to facilitate integrability with the PTL. It is anticipated that this implementation plan requires minimal upfront effort, which will organically increase as the maturity and capabilities of the PTL increase.

Validation of the proposed reference architecture is left for future work.

Acknowledgment

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Under Secretary of Defense for Acquisition and Sustainment (OUSD(A&S)) and the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) under Contract HQ0034-19-D-0003; TO 0309.

References

Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., & Railsback, S. F. (2010). The ODD protocol: A review and first update. *Ecological Modelling*, *221*(23), 2760–2768. https://doi.org/10.1016/j.ecolmodel.2010.08.019

Haynes, B., Carroll, T., Tollison, D., Kendrick, W., & Salter, A. (2003). *Development of the Missile Component Simulation Library (MCLib) for tactical missile simulation.* Paper presented at the Huntsville Simulation Conference, Huntsville, AL.

Hurt, T., McDonnell, J., & McKelvy, T. (2006, December 3–6). *The modeling architecture for technology, research, and experimentation.* Paper presented at the Proceedings of the 2006 Winter Simulation Conference.

IEEE. (2010). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules. *IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000)*, 1-38. https://doi.org/10.1109/IEEESTD.2010.5553440

Moore, R. V., & Tindall, C. I. (2005). An overview of the open modelling interface and environment (the OpenMI). *Environmental Science & Policy*, *8*(3), 279–286. https://doi.org/10.1016/j.envsci.2005.03.009

Parsons, D., Surdu, J., & Jordan, B. (2005). *OneSAF: A next generation simulation modeling the contemporary operating environment.* Paper presented at the European Simulation Interoperability Workshop, Toulouse, France.

Petty, M. D., & Weisel, E. W. (2019). Chapter 4 - Model composition and reuse. In L. Zhang, B. P. Zeigler, & Y. Laili (Eds.), *Model Engineering for Simulation* (pp. 57–85).

Rouse, W. B. (2019). *Computing possible futures: Model based explorations of "What if?"* Oxford University Press.

Rouse, W. B. (2022). *Transforming public-private ecosystems: Understanding and enabling innovation in complex systems*. Oxford University Press.

Schoonenberg, W. C., Khayal, I. S., & Farid, A. M. (2019). *A hetero-functional graph theory for modeling interdependent smart city infrastructure.* Springer.

Singh, R., & Mathirajan, M. (2014, December 9–12). *A conceptual simulation framework for the performance assessment of lot release policies.* Paper presented at the 2014 IEEE International Conference on Industrial Engineering and Engineering Management.

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 160018. https://doi.org/10.1038/sdata.2016.18

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net