ACQUISITION INNOVATION RESEARCH CENTER

November 22, 2022

Acquisition Innovation Research Center: Innovation for Digital Transformation and Policy Analytics

Dr. Hanumanthrao Kannan, *The University of Alabama in Huntsville*
Dr. Nirav Merchant, *The University of Arizona*
Dr. William Rouse, *Georgetown University*
Dr. Alejandro Salado, *The University of Arizona*
Dr. Young-Jun Son, *Purdue University*
Dr. Zoe Szajnfarber, *The George Washington University*

Sponsor: Dr. Dennis K. McBride, AIRC Director, Office of the Under Secretary of Defense for Acquisition and Sustainment

Cleared for Public Release.

Final Technical Report AIRC-2022-TR-011

WRT-1049.12c2

Task Order No. 0309

The Acquisition Innovation Research Center is a multi-university partnership led by the Stevens Institute of Technology and sponsored by the U.S. Department of Defense (DoD) through the Systems Engineering Research Center (SERC)—a DoD University-Affiliated Research Center (UARC).

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 No. 0309.

Table of Contents

List of Figures

List of Tables

Research Team

Name	Organization	Labor Category
Arianna M. Alfaro	The University of Arizona	Undergraduate researcher
Hanumanthrao Kannan	The University of Alabama in Huntsville	Co-PI
Nirav Merchant	The University of Arizona	Co-PI
William Rouse	Georgetown University	Co-PI
Alejandro Salado	The University of Arizona	Co-PI
Young-Jun Son	Purdue University	Co-PI
Zoe Szajnfarber	The George Washington University	Co-PI

Executive Summary

This report presents the results of the Policy Test Laboratory (WRT-1049.12c2) task, under Task Order WRT 1049 Acquisition Innovation Research Center: Innovation for Digital Transformation and Policy Analytics. The work was performed by the research team over a period of 2 months.

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.

The need for multidisciplinary modeling and the integration of heterogeneous models has been identified in other research fields. Efforts with different scopes and goals are undergoing, with different approaches in both technical and organizational governance. Some efforts focus on developing an integrated model out of which research questions may be addressed. Some efforts focus on enabling integration of models to let integration efforts emerge to respond to a wider variety of research questions. These existing efforts were assessed based on their background, goal, maturity (or state of the development), types of research questions they can support and application domains they serve, kinds of disciplinary models, data, and tools they are intended to support, architecture, and technical and organizational governance. This assessment has been used to increase the understanding of what type of approach may better fit the needs of the DoD to support acquisition research.

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. The <u>Application layer</u> handles aspects related to how organizations and infrastructure engage (e.g., security aspects or UI/UX). The <u>Problem class/Research question layer</u> handles aspects related to assessing if a given task can be supported by the PTL (as an integrated assessment tool). The <u>Models, Data, Tools layer</u> handles the actual research artifacts indicated by their names. The <u>Infrastructure layer</u> handles all aspects related to hosting, storing, and exchanging the research artifacts with the PTL consumers. The specific design of each layer embeds elements that contribute to trust and to model composability. The use of standard metadata accompanying the models, data, and tools injected into the PTL has been found to be a key asset to support interoperability. While some standards exist, they seem to be tailored to specific application domains. A new standard is required for the reference architecture of the PTL.

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 AIRC researchers to specific models, tools, or modeling approaches. Instead, AIRC 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 an AIRC team, resulting in a PTL that will grow larger, more mature, and more capable with every new AIRC-sponsored project. As the PTL matures, AIRC could incorporate additional constraints to be met by AIRC 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.

As a start, it is recommended that every AIRC award should incorporate an appendix to their contracts that require every awardee to deliver the following artifacts at the completion of their project:

- An Interface Control Document (ICD) for every model, dataset, and tool that they generate and/or deliver in their project.
- An ontology that captures all elements (objects, definitions, relationships) included in the models, datasets, and tools generated and/or delivered in their project.
- A repository that hosts every model, dataset, and tool they generate and/or deliver in their project, and that makes them available for use by any other AIRC researcher.

For sustainment of the PTL, six major tasks have been identified: Ontology development, Problem class taxonomy development, Infrastructure & application layers development, Data & models review and curation, Study review and curation, and General sustainment. These will require the setup and funding of dedicated teams under AIRC sponsoring.

Introduction

This report presents the results of the Policy Test Laboratory (WRT-1049.12c2) task, under Task Order WRT 1049 Acquisition Innovation Research Center: Innovation for Digital Transformation and Policy Analytics. The work was performed by the research team over a period of 2 months.

The government has identified several obstacles to inform effective and efficient acquisition policies. The defense budget serves many purposes, with many stakeholders. This leads 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. 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.

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. 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. 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.

1.1 Project Goals & Scope

The goal of this project was to establish 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.

Due to the anticipated multidisciplinary nature of the PTL, technical and governing aspects of the PTL were addressed during the development of the reference architecture. Existing reference architectures, frameworks, and standards for federated modeling and simulation and muti-scale modeling and simulation were assessed as a starting point for this effort.

Furthermore, the reference architecture is envisioned as a living artifact that can and will evolve to accommodate the needs and results of acquisition research as it progresses and matures. Guidelines to support this evolutionary aspect were developed as part of this project.

The development of the PTL itself was outside of the scope of this project.

1.2 Major Assumptions and Limitations

A set of Use Cases for the PTL was not provided at the beginning of the project. Three main usages were assumed:

1) The Government has a policy question that could be answered using the PTL.
2) A (policy) researcher wants to leverage the PTL.
3) A researcher wants to integrate their work in the PTL.

The research team identified that the main intent of the PLT could be (a) to support a suite of activities aimed at answering a wide diversity of policy questions or (b) 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 research team assumed that the PLT can begin with the intent in (a) and slowly transition to intent in (b) as data, models, and results from (a) become available.

The research team also assumed that the reference architecture had to be deployed soon after completing this project. Having in mind the duration of the project (2 months), this implied delivering an initial draft reference architecture with a high risk of presenting significant limitations during its first uses to support acquisition research, and the consequent need of rework and update. In this sense, the research team had freedom to explore reusing existing reference architectures or frameworks, tailoring them, adapting them, or developing a new one with no ties to existing ones.

Given the timeframe of this project and the lack of existing acquisition research projects where the reference architecture could be applied, the proposed reference architecture should not be considered validated.

Finally, several assumptions were made on the ability of the AIRC to enforce and/or incentivize the use of the reference architecture, as well as about the relationships between the AIRC, sponsors, and researchers. These are described later in the report, in the context of implementing and sustaining the reference architecture.

1.3 Report Organization

The rest of the report is structured in six sections, as follows:

- *Existing frameworks*: 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.
- *Reference Architecture*: This section describes and justifies the proposed initial reference architecture for the PTL.
- *Initial Testing*: Using a notional case, this section shows how the Reference Architecture may be used in practice.
- *Implementation Plan*: This section proposes a set of tasks and activities necessary to implement the Reference Architecture.
- *Sustainment Plan*: This section describes the set of tasks and activities necessary to sustain the Reference Architecture.
- *Conclusions and future steps*: This section summarizes the findings of this project.

Existing Frameworks

The idea of connecting models, tools, and data across domains to answer 'higher-level' questions is not novel. Several efforts have been undertaken to establish reference architectures or modeling frameworks to enable such integrations in different research areas. This section presents some of them, together with a summary of their main attributes.

2.1 Identification and Assessment Protocol

Due to time constraints, the effort allocated to identify and assess existing architectures and/or frameworks was timeboxed. The objective of this activity was not to be comprehensive but rather exploratory to get some idea of different ways in which a reference architecture for the PTL could be accomplished.

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.

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:

- Background, goal, and maturity (or state of the development) of the framework/architecture.
- Types of research questions it is intended to support, including application domains it serves.
- Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).
- Architecture: layers, components, integration, relationship between parts, services it provides, etc.
- Technical governance, including maintenance and, if possible, rough estimate of effort.
- Organizational governance, including maintenance and, if possible, rough estimate of effort.

2.2 Results

The assessment of each reference architecture or framework is presented below.

2.2.1 MIT Joint Program on the 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."[1] Its assessment is summarized in Table 1.

[1] https://globalchange.mit.edu/

Table 1. MIT Joint Program on Science and Policy of Global Change assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Founded in 1991 • Consortium model, with DOE as anchor tenant. High level policy impact and use. • Many publications.
Types of research questions it is intended to support, including application domains it serves.	• The program was designed to provide relatively quick comprehensive analysis ("integrated assessment") to support decision-making on global and climate relevant policy questions. They work in seven focus areas: Earth systems, Managed resources, Infrastructure & investment, Energy transition, Policy scenarios, Regional analysis, and Multi-sector dynamics. • E.g., *Our air pollution studies evaluate climate policy co-benefits for improved air quality, and environmental and health impacts of specific climate and/or air pollution policies. These include focused studies of the impact of policies on aerosols, PM2.5, ozone and mercury. Finally, we evaluate potential impacts of economic development policies on managed resources.*
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	Most of the Joint Program's work leverages the Integrated Global System Modeling (IGSM). It has two interacting components: • The Economic Projection and Policy Analysis (EPPA) model (a computable general equilibrium model from economics) and • 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. • For some specific purposes, new models are developed that use different data sets or aggregate sectors differently.
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	• Architecture of ecosystem: layers of "earth" and "economy" • Architecture of models: multiple levels of abstraction & aggregation suitable for disciplinary models • Architecture of software: EPPA uses GAMS which is standard for GGE models.
Technical governance, including maintenance and, if possible, rough estimate of effort.	• Program has had relatively steady funding on the order of $20M per year for more than a decade. • Each new study can be quite inexpensive: ~cost of an MS research assistant for period. (Many related studies at any one time). • Historically, they had ~2 full time staff scientists maintaining the core models – that has expanded some in the last several years as the overall program has expanded significantly.
Organizational governance, including maintenance and, if possible, rough estimate of effort.	• The above includes governance costs. • Interdisciplinary team of PIs (salary not included in above costs) do most of sponsor engagement, with some business manager support. • The staff research scientists play a critical role sustaining the shared modeling infrastructure and training new contributors.

2.2.2 *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.[2] Its assessment is summarized in Table 2.

Table 2. IEEE Std 1516-2010 assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Background: The High Level Architecture (HLA) was initially developed under the leadership of the U.S. Department of Defense DOD) 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 SISO. • Goal: 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, healthcare, 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 a 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).
Types of research questions it is intended to support, including application domains it serves.	• Interoperability and reusability of defense models (DMSO; DOD projects) • Development of supply chain network simulation, integrating geographically dispersed member simulations (NIST; Boeing) • A city-level traffic simulation, which calls driving behaviors (e.g., route diversion) using web services (DOT; FHWA)
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	Using HLA, researchers/practitioners have integrated a mix of the following elements: • System dynamics (aggregate level) • Discrete event (process flows) • Agent based (decision making, communications) • Dynamic systems or physics-based game engine • Hardware (e.g., robots, machines, drones; simulations running in real-time) • Human (simulations running in real-time), among others

[2] https://standards.ieee.org/ieee/1516/3744/

Attribute	Assessment
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	• HLA follows a publish and subscribe architecture • Operating system, software, applications, language agnostic • 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
Technical governance, including maintenance and, if possible, rough estimate of effort.	• An open source Portico (http://porticoproject.org/) or a commercial RTI (e.g., MAK Technologies) can be used, in which case efforts are needed to develop technical governance
Organizational governance, including maintenance and, if possible, rough estimate of effort.	• A governance structure/agreement needs to be established among sponsors and users.

2.2.3 *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 (Rouse 2019, 2022). 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). Its assessment is summarized in Table 3.

Table 3. Multi-Level Modeling assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Has been in use, and continually refined, for over ten years • Have addressed questions for over 10 different sponsors • Reported in numerous theses, journal articles and five books - 2 by Wiley; 3 by OUP
Types of research questions it is intended to support, including application domains it serves.	• Economics of scaling clinical trials to broader use (Emory, Indiana, Penn, Vanderbilt) • Likely impacts and efficacy of health policies (ACA, CMS) • Impacts of incentives on consumer energy behaviors (Accenture, GM)
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	Have employed a mix of • System dynamics (aggregate behaviors) • Discrete event (process flows, queues) • Agent based (decision making, communications) • Model composition aross levels can be a challenge

Attribute	Assessment
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	• Architecture of ecosystem: influence diagrams, systemigrams, casual loop models • Architecture of models: multiple levels of abstraction & aggregation • Architecture of software: AnyLogic, Repast, Simio, Vensim, R, Python
Technical governance, including maintenance and, if possible, rough estimate of effort.	• $200 - $300K for familiar domain • $500 - 1,000K for new domains • Difficult to update (model parameters) once sponsor's question is answered
Organizational governance, including maintenance and, if possible, rough estimate of effort.	• Engagement of sponsors and SMEs are key • Scheduling & conducting meetings can be a challenge • Driverless cars project involved over 70 meetings • Sponsors slow to adopt models for own use • Tend to provide numerous new questions for us to answer

2.2.4 CyVerse

CyVerse provides scientists with powerful platform to handle huge datasets and complex analyses, thus enabling data-driven discovery.[3] 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. Its assessment is summarized in Table 4.

Table 4. CyVerse assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Develop a federated platform for enabling diverse teams to collaboratively develop and share solutions for data driven questions • Continuously developed over 13 years and in active development. Adopting contemporary computational methodologies and architecture, while providing consistent and stable access to its underlying and evolving technology stack • Supports over 100,000+ users, 100's of grants and publications utilize the platform
Types of research questions it is intended to support, including application domains it serves.	• Methods development, securely sharing underlying tools/pipelines and data without needing any software installation on client side (web browser-based access) • Extensibility for developing customized applications using domain specific tooling and community adopted frameworks • Support for reproducibility of analysis with containerized (docker/singularity) workflows, scale from laptop to cloud/HPC • Application domains range from Astronomy to Earth Sciences, Hydrology, Traffic engineering and Life Science

3 https://cyverse.org/

Attribute	Assessment
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	• Support for analysis that need domain specific models (Gene prediction, image processing etc.) • Support for Machine Learning workloads (models generated by training data) and inference tools that are part of popular data science workbenches (Jupyter, RStudio, Tensorflow, PyTorch etc.) • Application lifecycle management: integration with code repositories, CI/CD pipelines, application versioning and linked reproducible execution environment
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	• CyVerse abstracts data storage, execution environments in the form of: • Data Grid: Metadata driven data management, agnostic of physical storage provider (cloud, on prem etc.) by providing unified view and consistent name space and access to data • Execution Environments: Federated single sign on (Keycloak), secure token-based access across platform, (JWT). Identity revenue management. Group management (Grouper) • Automation: Expressive REST API's (Science API) connecting all supported layer for rapid application development and customization • Web Interfaces: End user facing applications, customized for specific teams/analysis type, can be private or public • Learning: Training tailored to users, developers of varying level of computational competencies. Topics range from basic usage to scalability and custom development. Modes include online documentation, webinars, in person bootcamps or codefest, hackathons etc.
Technical governance, including maintenance and, if possible, rough estimate of effort.	• Multiple subscription levels/tiers that support projects and teams of varying size and computational needs (Basic, Regular, Pro, Commercial) on public CyVerse. \$0, \$200, \$340, \$2400 per year per person • Private CyVerse installations for secure enclaves (HIPAA, Defense), national consortiums (UK, Austria etc.), fortune 100 companies. Cost varies based on level of development and customization • Maintenance of Public CyVerse is managed by the UArizona CyVerse team and is included in the subscription cost. Maintenance and support of private applications developed by organizations and made available in public Cyvers is delegated to author/integrating lab member • Modes of user support include chat, email, zoom/screen share. • Integration of private infrastructure (compute, storage, identity etc.) falls under Professional services (see below)
Organizational governance, including maintenance and, if possible, rough estimate of effort.	• Powered by CyVerse program allows organizations to develop solutions that utilize specific component/layers from CyVerse and extend their own tools/platforms • CyVerse professional services provides deeper engagement with technical and scientific staff to architect novel solutions, customized training, maintenance of private installations etc. • Costs vary from \$120-\$300/hour

2.2.5 SESYNC

The National Socio-Environmental Synthesis Center (SESYNC) 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.[4] For 10 years, SESYNC has accelerated research and learning that seeks to understand the structure, functioning, and sustainability of coupled social and environmental systems. Its assessment is summarized in Table 5.

Table 5. SESYNC assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Brings together the science of the natural world with the science of human behavior and decision making to solve complex environmental problems - funding awarded 2011 • Goal: accelerating research and scientific discovery focused on solving problems at the interface of humans and the environment. Enhancing teams' and individual participants' capacities and skills to bridge varying epistemologies, methods, and approaches to advance socio-environmental synthesis and science • To create a center with highly flexible and adaptive organization design that could provide extensive support across all aspects of the community's socio-environmental (S-E) synthesis needs • 341 Supported Projects - including postdoctoral fellowships, workshops, team synthesis projects, short courses, & more; 4791 Researchers - from academia, government, non-government organizations, the private sector, & more; 77 Individual Countries -represented by our research community; 750+ Peer-Reviewed Publications - resulting from our supported research
Types of research questions it is intended to support, including application domains it serves.	• Quantitative Synthesis Methods: Literature Reviews (Systematic and Meta-Analyses), Expert Elicitation • Methods include statistical and spatially explicit modeling, dynamic systems modeling, and agent-based modeling. •S-E 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
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	• Dynamical systems characterized by feedback loops both within and between social and ecological subsystems • S-E systems are examples of complex adaptive systems (CAS) - positive feedback loops, negative feedback loops to build resilience •Frameworks may be in the form of network diagrams if the interest is in understanding relationships between individuals, groups, or institutions - collectively referred to as "agents" by S-E researchers • Flowchart-type frameworks are often used when there is interest in developing different scenarios to explore how futures may play out based on different policies or assumptions

[4] https://www.sesync.org/

Attribute	Assessment
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	• Cyberinfrastructure: Python, R, Shiny Server, SSH Gateway, Virtual Machines, GitLab Server, Databases, Software • CI support for: Processing/analyzing data on remote servers (e.g., "Slurm" computing cluster, Rstudio server, Jupyter server), Sharing code and results (e.g., git, GitHub/GitLab, GitHub gists, data visualization, interactive visualizations), Sharing and managing data (e.g. SESYNC storage, SQL databases)
Technical governance, including maintenance and, if possible, rough estimate of effort.	• Committed to public dissemination of data and to open-source software development and distribution • Data products resulting from SESYNC-sponsored activities are made accessible with no restrictions for use and dissemination through explicit use of a 'Creative Commons Zero waiver' or its equivalent and will be deposited in a public repository or established open database • All software source code developed with SESYNC funding or participation are made freely available via an FTP or code repository service (e.g., GitHub); will be licensed under an OSI - approved Open Source license (see opensource.org).
Organizational governance, including maintenance and, if possible, rough estimate of effort.	N/A

2.2.6 Simulation Framework (CSF)

The CSF was initiated by the US Army Aviation and Missile Research, Development, and Engineering Center (AMRDEC) in 1999 as a standardized structure to support dynamic simulations.[5] 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. Its assessment is summarized in Table 6.

Table 6. CSF assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Initiated by US Army AMRDEC in 1999 • Framework to conduct dynamic simulations to support systems acquisition and testing • Serves as a toolkit to support M&S of tactical missile systems • Object-oriented simulation environment

[5] B. Haynes, T. Carroll, D. Tollison, W. Kendrick, and A. Salter, "Development of the Missile Component Simulation Library (MCLib) for tactical missile simulation," in *Proceedings of the Huntsville Simulation Conference*, Huntsville, Ala, USA, October 2003.

Attribute	Assessment
Types of research questions it is intended to support, including application domains it serves.	• Simulation of missile deployment • 6 DOF Propulsion Aerodynamics Controls and Kinematics module • Hardware-in-the-loop testing • Both real-time and non-real-time supported
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	• Flexible - component models are not constrained by standards • Examples: Discrete event simulation, Differential equations for continuous event simulation • Component interfaces need to be compliant with CSF's specifications
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	• Common library approach • Components are linked into a common executable • Data exchange: method calls • Implemented in C++ • GUI to facilitate composition of models • Plug-ins are possible
Technical governance, including maintenance and, if possible, rough estimate of effort.	Not assessed.
Organizational governance, including maintenance and, if possible, rough estimate of effort.	Not assessed.

2.2.7 One Semiautomated Force (OneSAF)

The OneSAF is intended to foster interoperability and reuse across modeling and simulation communities of the Army.[6] Its assessment is summarized in Table 7.

Table 7. OneSAF assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Initiated by US Army in 1996 • To simulate brigade and below combat and noncombat operations at the entity level

[6] https://www.peostri.army.mil/OneSAF

Attribute	Assessment
Types of research questions it is intended to support, including application domains it serves.	• Development of advanced concepts for doctrine and tactics • Training of unit commanders and staffs • Development of new weapon systems • Production of data as input to other simulations • Applications: testing algorithms for real C4I systems, modeling WWII tank combat, creation of a "cyber range" for cyber warfare analysis and training, and virtual training on operating construction equipment
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	Physics-based models associated with platforms, soldiers, equipment, logistical supplies, communications systems and networks, emerging threats, and aviation assets.
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	• Layered approach • Components are linked into a common executable • Data exchange: method calls • No inherent mechanism to enforce the assumptions and dependencies of a component, if used in a different context • Validity of the composed system is up to the developer
Technical governance, including maintenance and, if possible, rough estimate of effort.	Not assessed.
Organizational governance, including maintenance and, if possible, rough estimate of effort.	Not assessed.

2.2.8 Modeling Architecture for Technology, Research, and Experimentation (MATREX)

The purpose of MATREX Modeling Architecture for Technology, Research and Experimentation 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.[7] Its assessment is summarized in Table 8.

[7] https://apps.dtic.mil/sti/citations/ADA500771

Table 8. MATREX assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Initiated by US Army in 2006 • To support integration of live, virtual, and constructive models operating at either the entity or engineering level • Not specific to an application; can be extended • C3 Grid
Types of research questions it is intended to support, including application domains it serves.	• Provide services related to modeling command, control, communications actions and effects • Supports testing, debugging, executions control, results logging and analysis • Support of distributed test events for network centric warfare systems • Networked effects of command and control
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	OneSAF, Aviation Mobility Server, Countermine Server, Missile Server, Weather Server, Vehicle Dynamics Mobility Server, Chemical Biological Simulation Suite, Comprehensive Munitions and Sensor Simulation, Logistics Server, Vehicle Level Human Performance Model
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	• HLA • Layered approach – Three layers • First layer: Federates – physics-based models • Second layer: Defines the link between the federates – Middleware that provides the interface between the federates and the interoperability protocol • Third layer: Distributed execution infrastructure
Technical governance, including maintenance and, if possible, rough estimate of effort.	Not assessed.
Organizational governance, including maintenance and, if possible, rough estimate of effort.	Not assessed.

2.2.9 OpenGMS

OpenGMS supports sharing of your modeling and simulation resources for geographic applications by providing a virtual community for collaboration among researchers from various domains.[8] Its assessment is summarized in Table 9.

Table 9. OpenGMS assessment

Attribute	Assessment
Background, goal, and maturity (or state of development) of the framework/architecture.	• Provide a comprehensive geographical problem solving platform that can be used transparently, efficiently and collaboratively • Develop open web-distributed geographic modeling and simulation theories and technologies • Highly difficult task and still has longs ways to go, limitations include: improving the standards, enhance the practicability and automation of tools, validation & calibration, uncertainty & scale problems
Types of research questions it is intended to support, including application domains it serves.	Questions related to hydrology, forest growth, traffic noise, geodetector, B-SHADE-Sampling, B-SHADE-Estimation
Kinds of disciplinary models, data, and tools it is intended to support, including integration capabilities (i.e., connecting across models, data, tools…).	There are too many to summarize; over 2,600 models associated with natural regions and human activities (e.g., development, social, economic) at different scales (e.g., regional, global) that include geoinformation, remote sensing, geostatistical, intelligent computation, physical processes, chemical processes, biological processes, and human activity processes.

[8] http://geomodeling.njnu.edu.cn/

Attribute	Assessment
Architecture: layers, components, integration, relationship between parts, services it provides, etc.	Layered • Model repository: 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. • Data repository: 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. • Models as a service: 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. • Thematic center: Regarding a research topic or a specific problem, multi-disciplinary knowledge could be involved. 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. And various web applications would be offered in certain centers.
Technical governance, including maintenance and, if possible, rough estimate of effort.	• Service-oriented tools for transparent accessible model service, for configurable data service, for easy participatory computing service resource supermarket-oriented portal • Model resource: Model wiki, Classification information, Metadata information, Conceptual description, Diagram expression, Computable service • Portal-OpenGMS- Model Res: over 4000 model items and 3000+ invokable services - with CSDMS • Portal-OpenGMS-Data Res: over 2000 data item
Organizational governance, including maintenance and, if possible, rough estimate of effort.	• Data sharing to simulation resource sharing • Local use to remote service • Cloud computing to edge computing • Meeting-based discussion to network-based collaboration

2.3 Findings

Existing frameworks display a wide variety of approaches to establish frameworks that enable the integration of models across disciplinary boundaries. There seems 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 is left open for the different modeling actors to define. These frameworks are generally established through working groups or standards bodies.

- *Top-down*: 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 integrated overarching 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*: Similar to structural frameworks in the sense that a structure to enable integration is provided, but additional guidance and infrastructure are provided to 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.

Out of the existing frameworks presented in Section 2, OpenGMS seems to be the closest to how the PTL may eventually look like (as understood by the research team). A snapshot of its homepage is shown in Figure 1. OpenGMS provides three main capabilities: sharing of models and data, reuse of models and data, and integration capabilities to integrate different models and data.

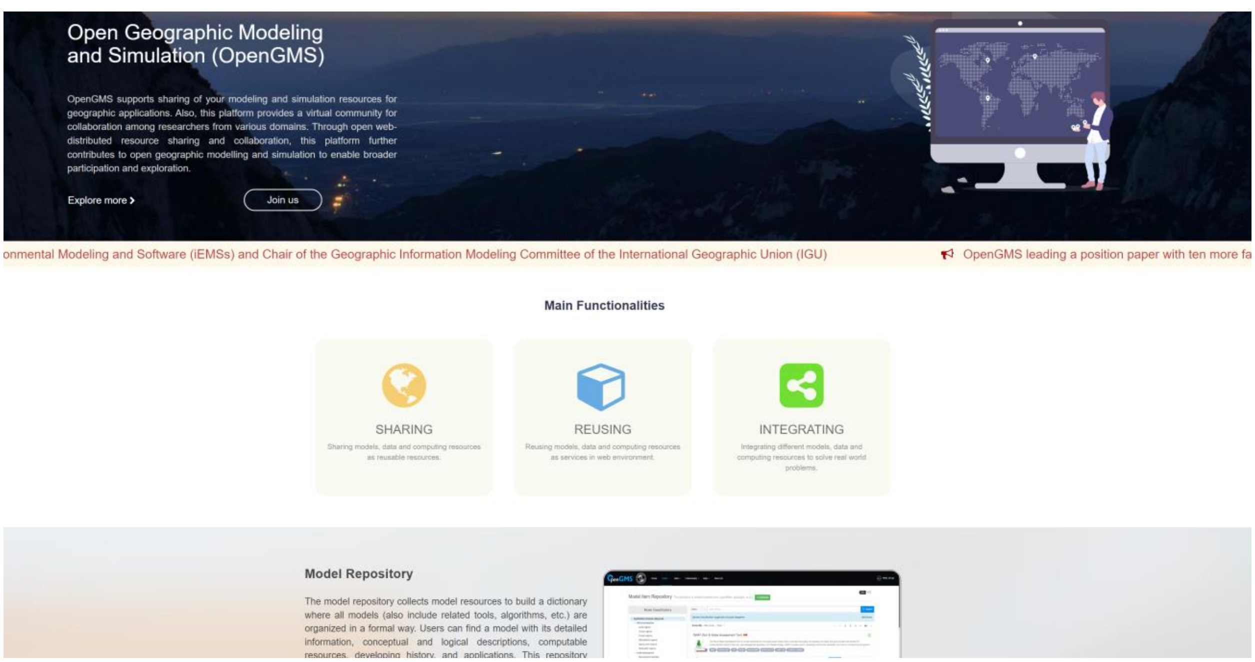

Figure 1. OpenGMS home webpage

An interesting aspect of their solution is that it is built on a service-oriented architecture. This means that a researcher can, instead of downloading or copying a model, simply “call” a model to run their data. This is handled directly in the OpenGMS repository, so the researcher only needs to provide their data and wait for the results to come back.

Reference Architecture

This section presents the initial reference architecture developed in this project. As a clarification, it should be noted that this is not the architecture of the PTL, but a reference architecture from which a PTL can be architected and developed.

The architecture is represented here without following any specific formalism due to the time limitation of the activity.

3.1 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?*

3.2 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) Why should I use the PTL?

(2) 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 in this 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 10. 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 10. 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 further 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 and 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. It should be noted that syntactic composability is a necessary condition for semantic composability, but not sufficient.

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 and 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 and 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.

3.3 Reference 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 (ref. Figure 2). 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.

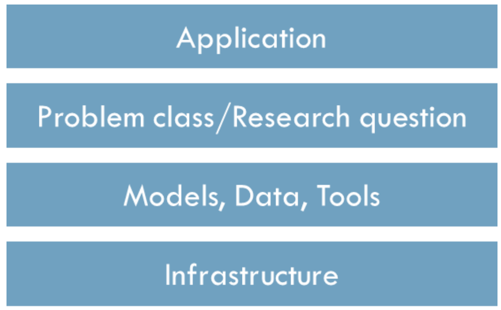

Figure 2. Reference architecture

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 the specific design of the PTL.

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 within the time constraint of the project.

3.3.1 Application Layer

The Application layer provides 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.

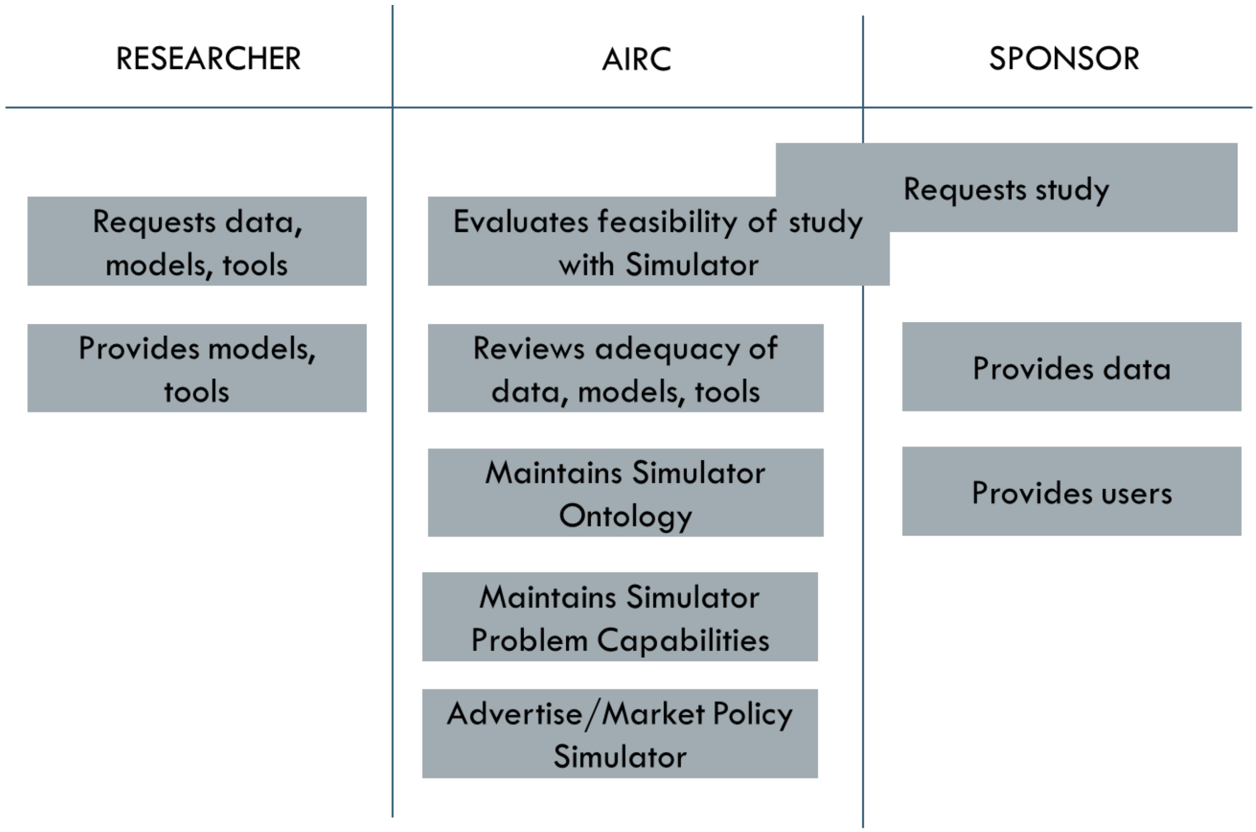

Figure 3. Engagements between PTL actors

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 has not been assessed in this project.

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 has not been assessed in this project.

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 has not been assessed in this project.

3.3.2 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 project.

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.

3.3.3 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) and the ODD (Overview, Design concepts, Details) Protocol.

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."[9] 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.[10] 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.

[9] https://www.ogc.org/standards/openmi

[10] https://www.go-fair.org/fair-principles/

        - 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, 2018). 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.

An initial set of metadata requirements for AIRC projects is provided in Appendix B.

3.3.4 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.

Initial Testing

This section presents some initial testing of the proposed reference architecture. Note that the reference architecture only provides a structure on which a PTL may be designed and developed. Testing a reference architecture would require showing that effective PTLs can be designed following the reference architecture. This is outside of the scope of this project. Instead, the testing is limited to show how a notional PTL would "activate" the different elements of the reference architecture when responding to a test case.

The test case represents a notional need of the DoD to inform decisions regarding acquisition supply chains.

4.1 Test Case

Beyond the normal demands on military supply chains, the Department of Defense (DoD) now is supplying Ukraine with weapons and munitions, and must prepare for the possible defense of Taiwan. The duration and nature of Ukraine's needs are uncertain, and the extent to which needs to defend Taiwan will emerge is highly uncertain. Given these three scenarios – two real and one emergent – how should the DoD manage its acquisition supply chains?

Answering this question involves addressing several issues. An immediate question concerns inventories of weapons and consumables needed to supply military operations by the US and its allies. Existing mission plans of the Department and our allies likely provide guidance on inventory consumption and replenishment needs. Of course, uncertain demands and consumption must be traded off against budgets available.

Further upstream are plans for acquiring new capabilities, both on the drawing board and under contract. Are these the right investments given the three scenarios? Should some things be accelerated and others slowed? Are there capabilities that should be expedited from the military research labs and defense companies? What are the risks associated with answering all these questions.

Can the likely demands of these three scenarios be influenced by political rather than military means? Can economic policies and incentives be used to temper the extent and timing of potential conflicts? What uncertainties and risks are associated with this approach? Will the Department's acquisition supply chains have to invest in hedges against failure of political means?

4.2 Level of Analysis

At the proximate level, the question is what monies are needed when to support the three demands. The technology supply chain is the most distal level. The spectrum of levels include:

- Money: How should <u>monies over time</u> be allocated among each of the supply chain challenges?
- Production: How should <u>production capacities</u> be allocated among each of the supply chain challenges?
- Capabilities: How should <u>weapons capabilities</u> be allocated among each of the supply chain challenges?
- Acquisition: What <u>new weapons capabilities</u> should be acquired for each of the supply chain challenges?
- RDT&E: What <u>new technologies</u> are needed to enable new capabilities for each of the supply chain challenges?

The choice of the most appropriate level of analysis depends on the sponsors' preferences. Answering these questions requires exploring different phenomena of interest simultaneously, since the answer to each question depends on every other answer. For example:

- Financial resources are limited. Therefore, the budget available to develop and deploy new capabilities is influenced by how much budget is allocated to produce and deploy existing capabilities.
- The need for new capabilities depends on the effectiveness and availability of existing capabilities.
- The deployment of existing capabilities depends on the availability and status of existing production capabilities.
- …

At the same time, each individual analysis requires different modeling paradigms and datasets, as described in the next section.

4.3 Phenomena of Interest and Alternative Representations

The following phenomena will need to be represented and computationally predicted as a functions of alternative scenarios and interventions:

- Money over time
- Demands on capacities per capability over time
- Delivery of capabilities over time
- Acquisition of new capabilities over time
- Maturity of new technologies over time
- Uncertainties (e.g., probability distributions of costs, capacities needed, capabilities needed and delivery times, acquisition times, technology performance)

The above phenomena can be represented in the following ways:

- Economic models of cash flows
- Probabilistic models of uncertainties
- Discrete event models of production
- Discrete event models of delivery
- Discrete event models of acquisition
- Sigmoidal models of technology maturity

4.4 Reference Architecture in Action

Table 11 shows the different actions that occur for responding to the test case, together with a description of what reference architecture aspect is activated in the response. Given that the objective of the test is limited to the reference architecture, no assumption is made about the actual design of the PTL. When possible, the table also indicates different ways in which the specific PTL could respond, showing that they are all conforming with the structure set forth by the reference architecture. Note that the actions are not necessarily sequential but are showing sequentially for convenience.

Table 11. Reference architecture in action (simplified)

ID	Action	Reference architecture response
1	The sponsor requests the study	Application layer handles the request.
2	AIRC evaluates feasibility of the request	AIRC interacts with the PTL through the application layer. The application layer sends the request to the Problem Class/Research Question Layer, which surveys existing artifacts in the Models, Data, Tools Layer. Through the Application Layer, the Problem Class/Research Question Layer informs the AIRC team of the feasibility of the request (with certain level of confidence) assuming that the Sigmoidal models of technology maturity and the Discrete event models of production, which are not yet in the PTL, are incorporated in the PTL.
3	The results of feasibility and need for additional models/data is provided to the sponsor	Application layer handles the interaction.
4	A contract is issued to a research team to develop the necessary models and, together with the PTL, respond to the sponsor's request.	n/a
5	The research team accesses existing PTL artifacts.	Through the Application Layer, the research team can find, select, and access necessary artifacts in the Models, Data, Tools Layer.
6	The research team develops the new models and data and injects them into the PTL.	Through the Application Layer, the research team submits new artifacts in the Models, Data, Tools Layer.
7	AIRC reviews and curates existing models	Through the Application Layer, AIRC is informed of new content in the PTL and asked to review and curate the information. Artifacts in the Models, Data, Tools Layer.
8	AIRC reviews PTL capabilities due to new artifacts	Through the Application Layer, AIRC reviews and updates the Problem Class/Research Question Layer using the curated artifacts in the Models,

		Data, Tools Layer, and commands the PTL to release results to the sponsor. This includes establishing confidence level of the integrated models, data, or tools.
9	The sponsor receives the results of the study.	The Application Layer handles the interaction, providing visibility on results (and confidence levels) to the requested research question.

All of PTL responses make use of the Infrastructure Layer, since it hosts the implementation of the PTL and all of its artifacts.

Note the ability of the reference architecture to handle the test case without specific design decisions of the PTL.

Implementation Plan

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 AIRC researchers to specific models, tools, or modeling approaches. Instead, AIRC 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 an AIRC team, resulting in a PTL that will grow larger, more mature, and more capable with every new AIRC-sponsored project. As the PTL matures, AIRC could incorporate additional constraints to be met by AIRC 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 (notionally sketched in Figure 4).

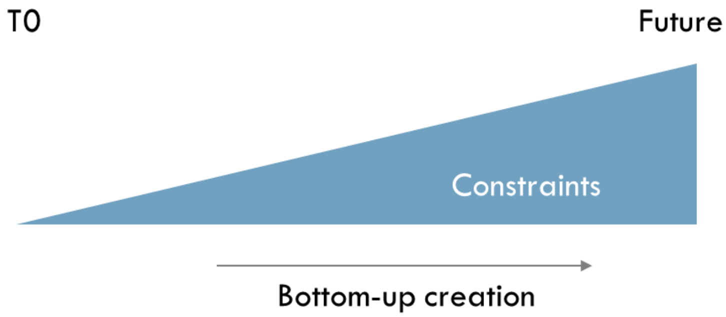

Figure 4. Notional sketch of anticipated effort to implement the PTL

As a start, it is recommended that every AIRC award should incorporate an appendix to their contracts that require every awardee to deliver the following artifacts at the completion of their project:

- An Interface Control Document (ICD) for every model, dataset, and tool that they generate and/or deliver in their project.
- An ontology that captures all elements (objects, definitions, relationships) included in the models, datasets, and tools generated and/or delivered in their project.
- A repository that hosts every model, dataset, and tool they generate and/or deliver in their project, and that makes them available for use by any other AIRC researcher.

A sample of such requirements is provided in Appendix B.

Sustainment Plan

Six tasks have been identified to support the sustainment of the PTL. These will require the setup and funding of dedicated teams under AIRC sponsoring. An initial, notional assessment of relative anticipated effort for each task is sketched in Figure 5.

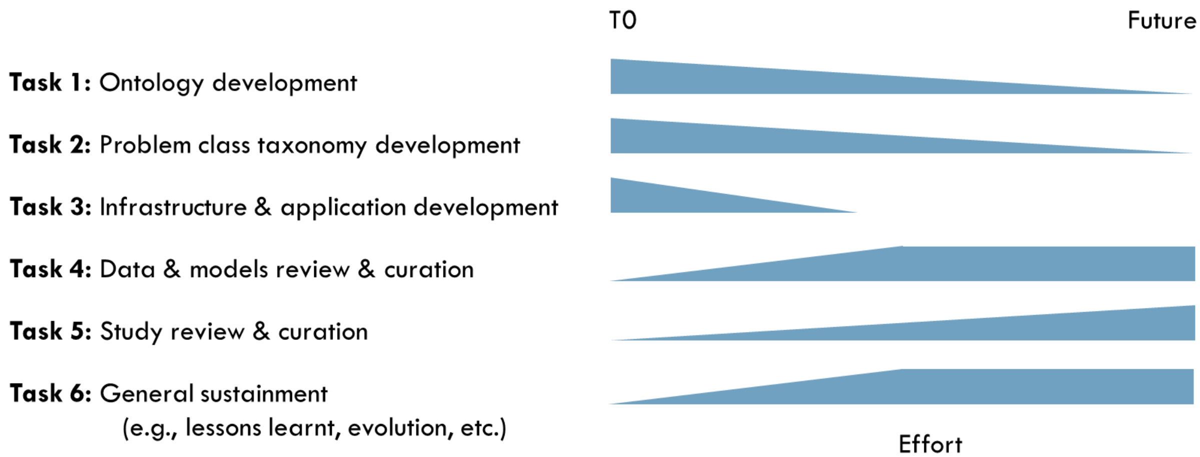

Figure 5. Notional PTL sustainment effort

Task 1. *Ontology development*. The goal of this task is to establish an ontology of acquisition. The acquisition ontology defines the overarching framework for integrating datasets and models in a meaningful manner. The construction of the ontology is expected to evolve as needed by the research questions sponsors are interested in and the models and datasets generated by researchers. In this way, only those elements that need to be integrated are represented in the ontology. The sustainment effort for the ontology is anticipated to decrease with time, as the ontology matures. It is recommended to support a dedicated research team for this task.

Task 2. *Problem class taxonomy development*. The goal of this task is to develop a taxonomy of the different classes of problems or kinds of research questions the PRL can answer. This includes developing verification and validation (V&V) methods to characterize the confidence yielded by models that result from integrating heterogeneous models. The sustainment effort is anticipated to decrease with time, as the taxonomy and V&V methods mature. It is recommended to support a dedicated research team for this task.

Task 3. *Infrastructure & application layers development*. The goal of this task is to design and implement the backend and the frontend of the PTL, that is, the Infrastructure and the Application Layers. This task is not anticipated to extend in time, but rather be a timeboxed activity. Note that this task is considered part of the sustainment plan and not of the implementation plan. This is because it is proposed to start the PTL with a bottom-up approach, giving freedom of implementation to researchers early on. In parallel, as more knowledge is available about the needs and constraints of the PTL, the infrastructure and application layers may be developed. It is recommended to support two dedicated development teams (no research needed) for this task.

Task 4. *Data & models review and curation*. The goal of this task is to make sure that every artifact that is fed into the PTL meets all necessary criteria to enable interoperability with other artifacts in the PTL. The task consists of (1) reviewing the models, data, and tools (with associated metadata) delivered by researchers and sponsors to assess conformance with project requirements, and (2) curate conforming models, data, and tools as necessary. Depending on the cadence with which models, data, and tools are fed into the PTL, this may be an ongoing effort or an on-demand assignment proportional to the use of and contribution to the PTL. Dedicated AIRC personnel must be allocated to support this task.

Task 5. *Study review and curation*. The goal of this task is to make sure that every study request that aims at depending on the PTL is feasible. The task consists of reviewing study requests, assessing conformance and/or feasibility with the existing capabilities of the PTL, and/or assessing the need of additional PTL capabilities and feasibility of developing them according to the programmatic needs of the sponsor. Dedicated AIRC personnel must be allocated to support this task. Depending on the cadence with which studies are requested, this may be an ongoing effort or an on-demand assignment.

Task 6. *General sustainment*. To guarantee the relevance and adequate functioning of the PTL, it is anticipated that some general sustainment tasks of all infrastructure, programs, operational processes, etc. is required. Effort and organization of this task should be defined as the PTL implementation progresses.

Conclusions and Next Steps

The need for multidisciplinary modeling and the integration of heterogeneous models has been identified in other research fields. Efforts with different scopes and goals are undergoing, with different approaches in both technical and organizational governance. Some efforts focus on developing an integrated model out of which research questions may be addressed. Some efforts focus on enabling integration of models to let integration efforts emerge to respond to a wider variety of research questions. These existing efforts have been used to increase the understanding of what type of approach may better fit the needs of the DoD to support acquisition research.

An initial reference architecture to support the development of the PTL has been presented. 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.

The reference architecture consists of four layers. 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.

As indicated earlier, the reference architecture has been conceived as a living artifact. Its evolution is expected, particularly as knowledge about the needs, constraints, and good practices of acquisition research increases. The proposed next steps in this direction have been laid down in the Implementation and Sustainment Plans. Particularly, this work suggests the following two main actions:

1) Add an Appendix requesting ancillary information for every model, data, and tools generated as part of an AIRC-sponsored project.

2) Establish dedicated research and development teams to start the development and sustainment of the PTL according to the guidelines provided by the proposed reference architecture.

Appendix A. Cited and Related References

Volker Grimm, Uta Berger, Donald L. DeAngelis, J. Gary Polhill, Jarl Giske, Steven F. Railsback, The ODD protocol: A review and first update, *Ecological Modelling*, Volume 221, Issue 23, 2010, Pages 2760-2768.

Petty, M. D., and E. W. Weisel. (2019). Model Composition and Reuse. *Model Engineering for Simulation*. Academic Press, 2019. 57-85.

Rouse, W.B. (2019). Computing Possible Futures: Model-Based Explorations of “What If?” Oxford, UK: Oxford University Press.

Rouse, W.B. (2022). Transforming Public-Private Ecosystems: Understanding and Enabling Innovation in Complex Systems. Oxford, UK: Oxford University Press.

W.C.H., Schoonenberg, I.S. Khayal, A.M. Farid. (2018). A Hetero-functional Graph Theory for Modeling Interdependent Smart City Infrastructure. Springer.

Appendix B: Sample Interface Control Document (ICD) Requirements For Acquisiton Research Models and Data

Pending refinement of the reference architecture and design of the PTL, the following metadata should be requested for every model and dataset generated as part of AIRC-sponsored projects.

For models:

- Conformance to ODD Protocol. Since some elements may need to be adjusted for the specific models developed in the project, the ODD Protocol is to be tailored in agreement with AIRC.
- Description and characterization of software, implementation language, operating system, compiler version, tools, etc. employed in the creation and use of the model.
- Interface description, including syntax, data definitions, and applicable standards.
- Characterization of performance, intended use, and uses for which the model is not intended.
- Identification of developers, prior uses, and validation history.

For datasets:

- Conformance to FAIR principles.

For all:

- Underlying ontology, which can be used to satisfy some of the requirements and/or principles in ODD and FAIR.
- Models and datasets shall be stored for at least 5 years and made available to AIRC through secure access.