SYM-AM-26-109

Excerpt from the Proceedings

of the

Twenty-Third Annual Acquisition Research Symposium and Innovation Summit

Thursday, May 7, 2026 Sessions

Volume II

"Accelerating Warfighting Capabilities"

The Policy Test Lab: An Agentic AI-Based Simulation Tool

Published: April 30, 2026

The research presented in this report was supported by the Acquisition Research Program, Graduate School of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, please contact:

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
E: arp@nps.edu
www.acquisitionresearch.net

The Policy Test Lab: An Agentic AI-Based Simulation Tool

Douglas J. Buettner, PhD—is the AIRC Deputy Chief Scientist at the Stevens Institute of Technology and is an Adjunct Assistant Professor for the University of Utah's Systems Engineering Program. His doctoral research at the University of Southern California's Viterbi School of Engineering examined issues with DoD software-acquisition strategies, identifying acquisition optimizations. His research is found in Acquisition Games and Software Defects: Dynamics of the Software Developer's Dilemma. Dr. Buettner has BS and MS degrees in Physics, with a minor in Atmospheric Science from Oregon State University, and his PhD from USC is in Astronautical Engineering. [dbuettne@stevens.edu, 201-216-8300]

Raz Saremi, PhD—is an AIRC Research Scientist at the Stevens Institute of Technology. Dr. Saremi brings extensive industry experience in the machine learning, data science, and advanced analytics domain, with leadership roles at Archer Daniels Midland (ADM), Johnson & Johnson, Celgene (BMS), and Merck. At ADM, she led the company's generative AI strategy, designing large-scale market intelligence and forecasting systems using Large Language Models, network science, time series forecasting, and multi-objective optimization to improve strategic decision-making across commercial and supply chain operations. Her PhD is in Software and Systems Engineering from Stevens. [rsaremi@stevens.edu, 201-216-8300]

Ethan Silverstein—is a graduate student at the Stevens Institute of Technology. He is a Research Assistant in the Mobile AI Cybersecurity Computing laboratory, focusing on AI Operations for 5G communications systems, formal verification methods for distributed AI systems, and UAV security protocols with emphasis on scalable AI deployment and monitoring. He spent last summer as a Research Intern at the Air Force Research Laboratory designing multi-agent reinforcement learning policies for distributed command and control problems, outperforming traditional heuristic methods. Ethan's BS is in Computer Science from Stevens. [esilvers@stevens.edu, 201-216-8300]

Zoe Szajnfarber, PhD—is the SERC Chief Scientist at George Washington University and is a Professor of Engineering Management and Systems Engineering, International Affairs, and Computer Science, Faculty Director of the GW Trustworthy AI Initiative. Her research group focuses on the design and development of complex systems in their socio-technical context. Recent work has focused on strategies for leveraging nontraditional design inputs, from open innovation to AI-enabled workflows. Dr. Szajnfarber's MS and PhDs are from the Massachusetts Institute of Technology. [zszajnfa@email.gwu.edu, 202-994-6080]

Abstract

As an emerging technology, Agentic Artificial Intelligence or Large Language Models (LLM)–based/Generative AI agent systems, have been increasingly adopted to enable autonomous reasoning, tool-using, and decision-making systems that transcend the traditional boundaries and use cases of language models. Many of these systems rely on multiple interacting agents, each capable of perception, reasoning, and adaptive behavior. Such agents collaborate in multilayered ecosystems to achieve cognitive and operational objectives. The policy test system introduced in this paper extends the agentic paradigm into the policy simulation domain by designing an LLM-based orchestration agent that autonomously interprets academic documents and translates them into executable implementations of simulation and optimization models.

While this research is still in its infancy, the agentic tool already extracts an analogous model from peer reviewed literature, where the LLM serves as a cognitive controller, parsing unstructured knowledge into executable code. The resulting code provides simulation agents with the underlying dynamics of policies, resource flows, and behavioral adaptation. This work is a step towards a future where generative reasoning agents autonomously analyze, simulate, and optimize complex socio-technical systems in support of informed policy exploration.

Introduction

Policy test simulation has been an active area of research by the Stevens Institute of Technology's Systems Engineering Research Center (SERC) for many years (Rouse, 2014;

Rouse & Bodner, 2012, 2013a, 2013b; Rouse et al., 2019; Kannan et al., 2022). However, traditionally the approach requires upwards of 2 years to complete, starting with interviews of the affected stakeholders, developing a model of the policy, finding and using the data for calibration and testing the model, and then exploring the ramifications of those policies on the affected stakeholders. While the approach should still be considered as the “gold-standard,” the immediate need for policy change to affect our acquisition transformation strategy across our war fighting enterprise outweighs the time required to fully develop and test these models. This begs the question, *Is there another way?*

The Policy Test Lab (PTL) that is now under development by the Acquisition Innovation Research Center (AIRC) leverages Agentic Artificial Intelligence (AI) or Large Language Models (LLM)–based/Generative AI agents with autonomous reasoning and tool-using capabilities in support of exploring a decision-maker’s policy questions. Even though the PTL is in its infancy, it has the potential (within computational constraints of course), to simulate virtually any complex socioeconomic system. Initially, it will be used to simulate and test Department of War (DoW) acquisition transition strategies (DoW, 2025).

Figure 1 shows the Operational View (OV-1) of Rouse’s original Policy Flight Simulator (PFS).[1] Once implemented, the PTL is expected to operate in much the same manner but will use Agentic AI to implement many of Rouse’s steps. Furthermore, as models are built, explored, and completed for the policy-makers, they will be added to a central repository for future exploration and refinement, becoming building blocks for more complex models. Hence, an OV-1 of the PTL is quite similar, containing the same essential steps under the hood, but now they are automated using skilled AI agents executing this multistep process. A revised OV-1 with the “Multi-Level Computational Policy Simulator” or (MCPS) at its core is shown in Figure 2.

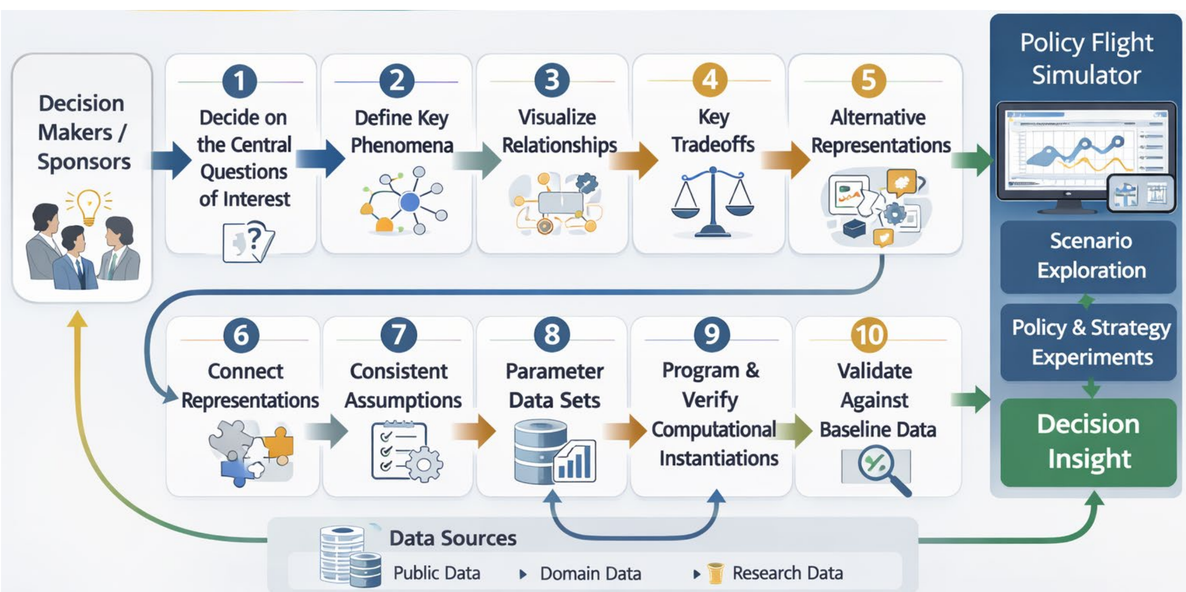

Figure 1. Operational View (OV-1): Rouse’s Policy Flight Simulator

[1] Appendix A directly quotes Rouse’s (2014) 10-Step Methodology and is provided for reference.

Figure 2. Operational View (OV-1): The Agentic AI-based Policy Test Lab

The remainder of this paper discusses the MCPS design and architecture, and its current implementation status.

The Multilevel Computational Policy Simulator (MCPS) Architecture

Internal to the PTL's MCPS, the task flow starts with a policy question and uses a chat agent to create a scenario that is used to help identify existing literature in the policy question's area, focusing on papers containing models frequently used in policy studies, namely system dynamics, discrete event, and agent-based or hybrid models. The scenario contains key words and synonyms for a detailed literature search, and identifies a list of key stakeholders and suggested names for stakeholder agents in the policy area, associated topics, and model types to look for with potential sources of data. The scenario is provided to the PTL's Orchestrator Agent, which then builds a list of queries for a literature search. The user also has the option of expanding from a generic OpenAI "websearch," to include in each query a targeted organization's name, such as RAND, MITRE, DTIC, SERC, or AIRC, among others.[2] The general concept is to first identify what has been already studied. We decided to start with this very generic approach to allow the tool to use models that have already been built, if they exist, and if nothing is found, then we have the basic information from which to start building our own models from scratch.

Figure 3 is the process view sequence diagram from our initial phase of development.

[2] It is important to note that when including organizations, the literature search can find potentially hundreds of articles, which can add up quickly even though we have defaulted to using cheaper models.

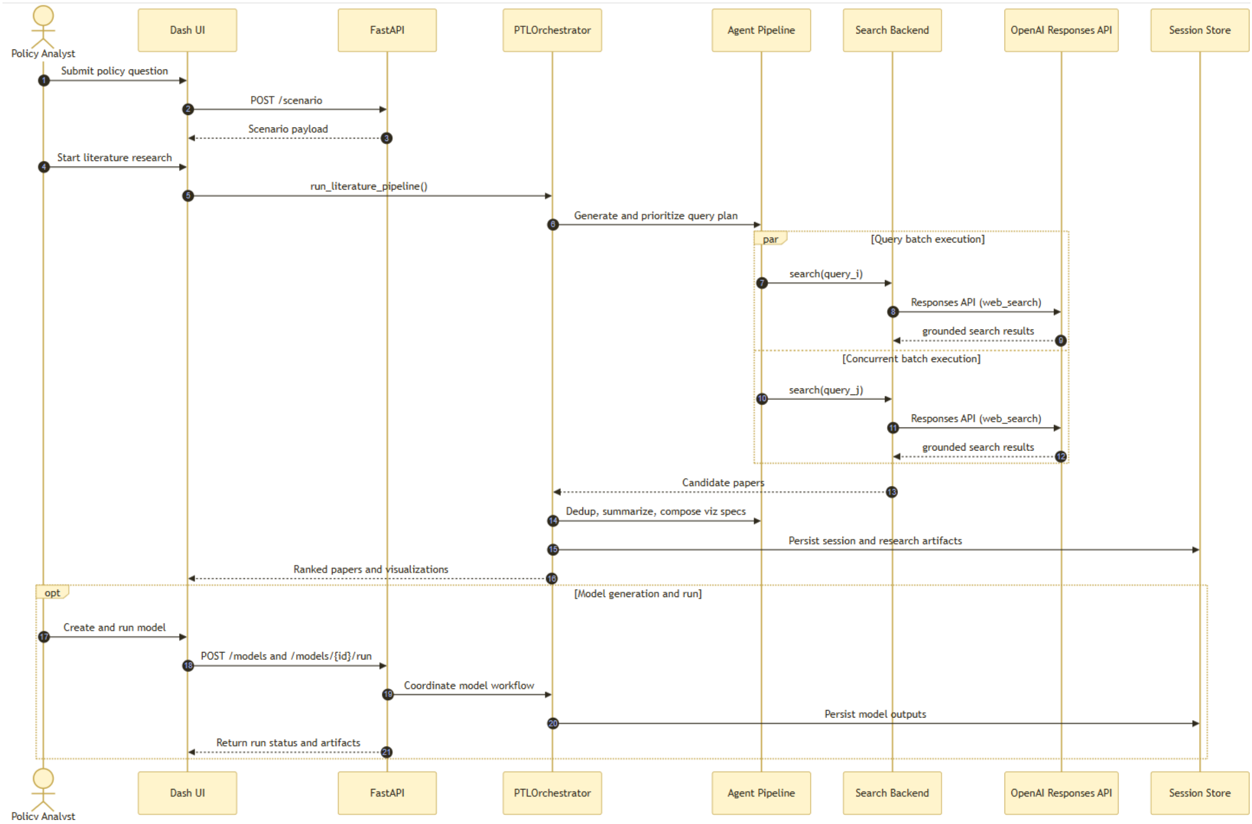

Figure 3. Process (View) Flow for this Version of the Policy Test Lab's MCPS

Select 4 + 1 architectural views are provided in Appendix B.

The PTL's Simulator Generator

If one or more promising papers are identified, the policy analyst can then send that paper to the Simulator Generator suite of agents to carefully analyze the paper's contents for existing models, which includes coding agents that generate Python code for the models found within. To contain costs while the PTL is under development and being debugged, we have restricted these agents from analyzing papers with more than 50 pages of text, and then we have focused on building very simplistic representations of potentially complex models.[3] The Simulator Generator's multi-document upload interface is provided in Figure 4, with Figures 6, 7, and 8 showing extracted System Dynamics Model (SDM) simulation, Agent-Based Model (ABM), and Discrete Event Simulation (DES) outputs respectively.

[3] We expect to include more agents in the future to accurately reproduce a paper's model.

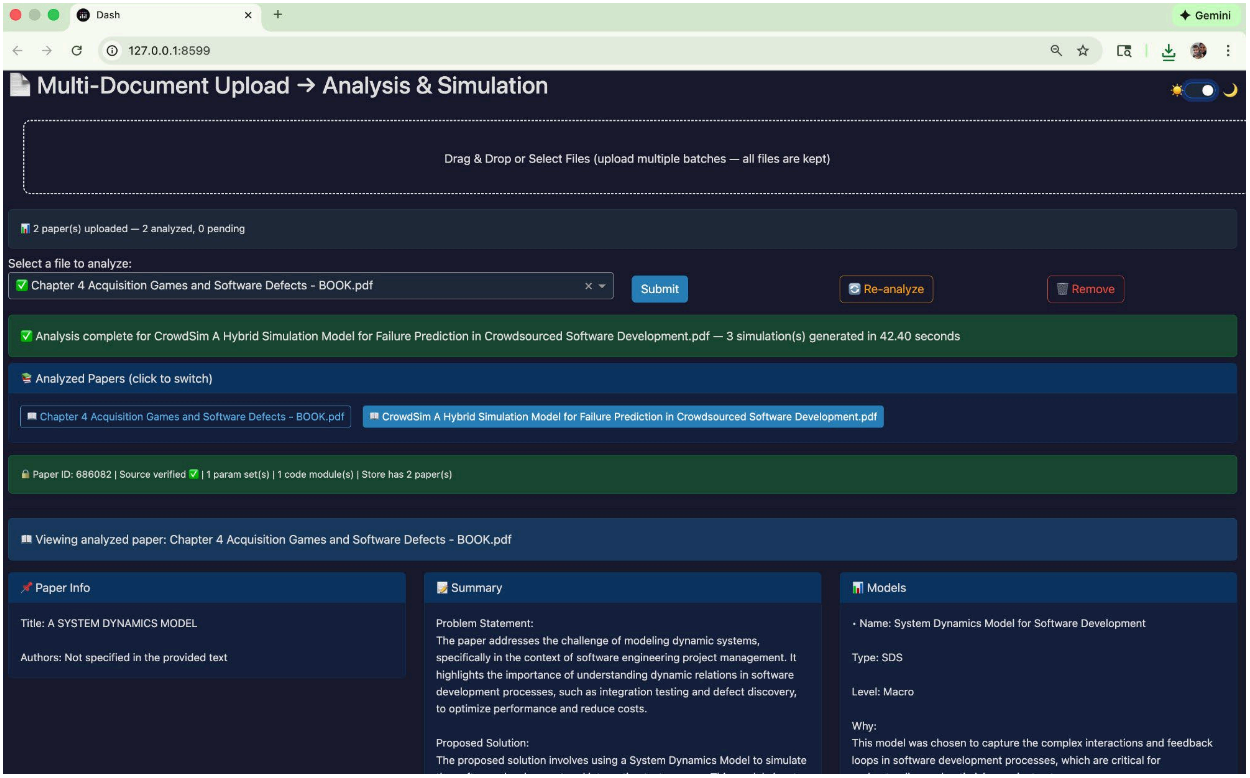

Figure 4. The Simulator Generator's Multi-Document Upload Interface

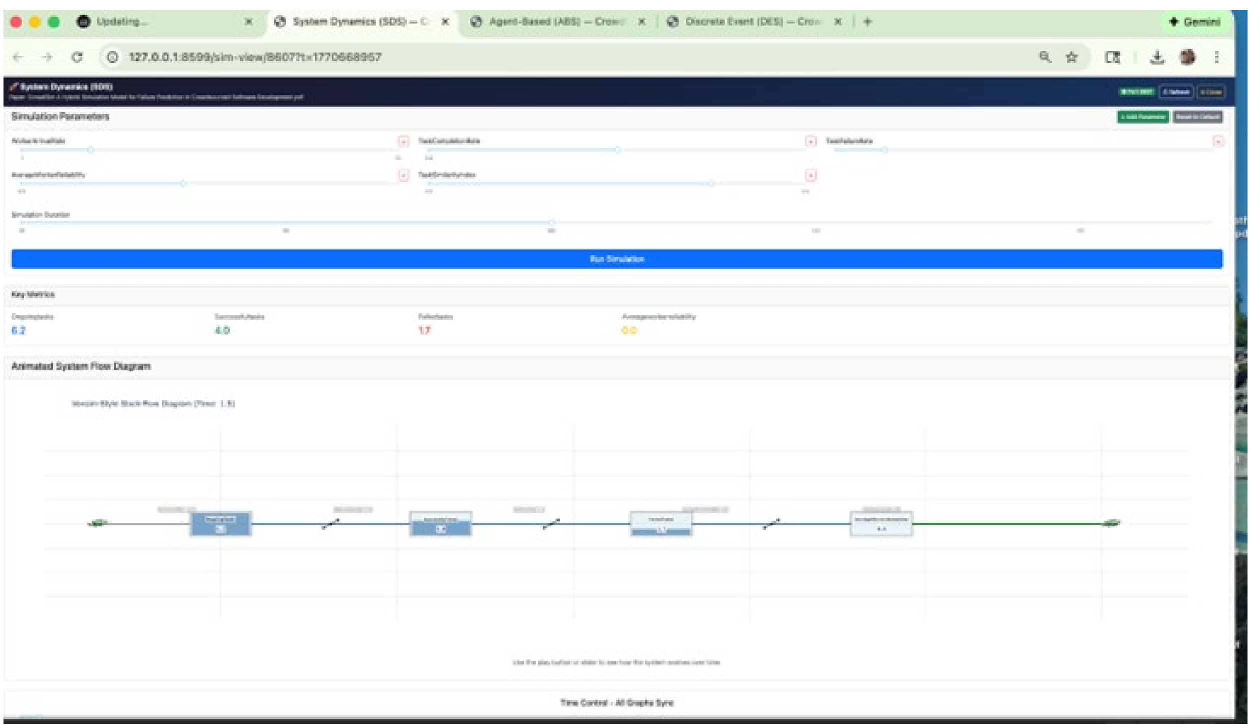

Figure 5. A Generated System Dynamics Model Simulation

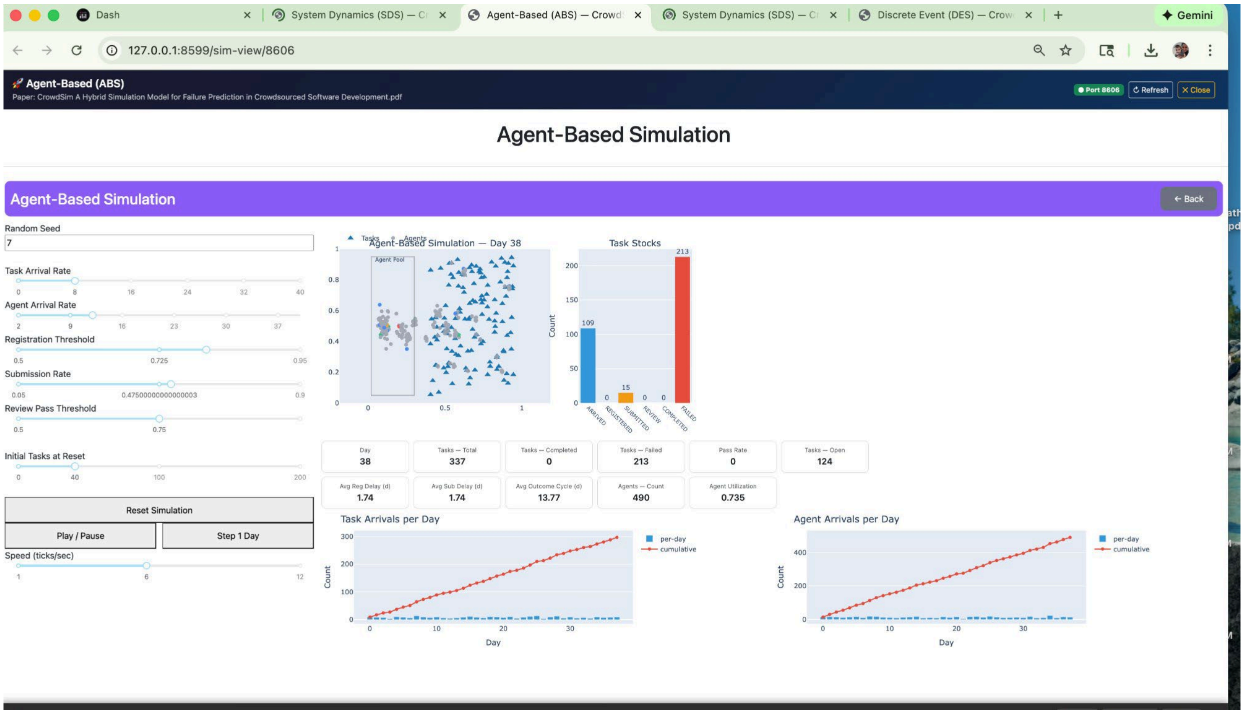

Figure 6. Agent-Based Model Simulation

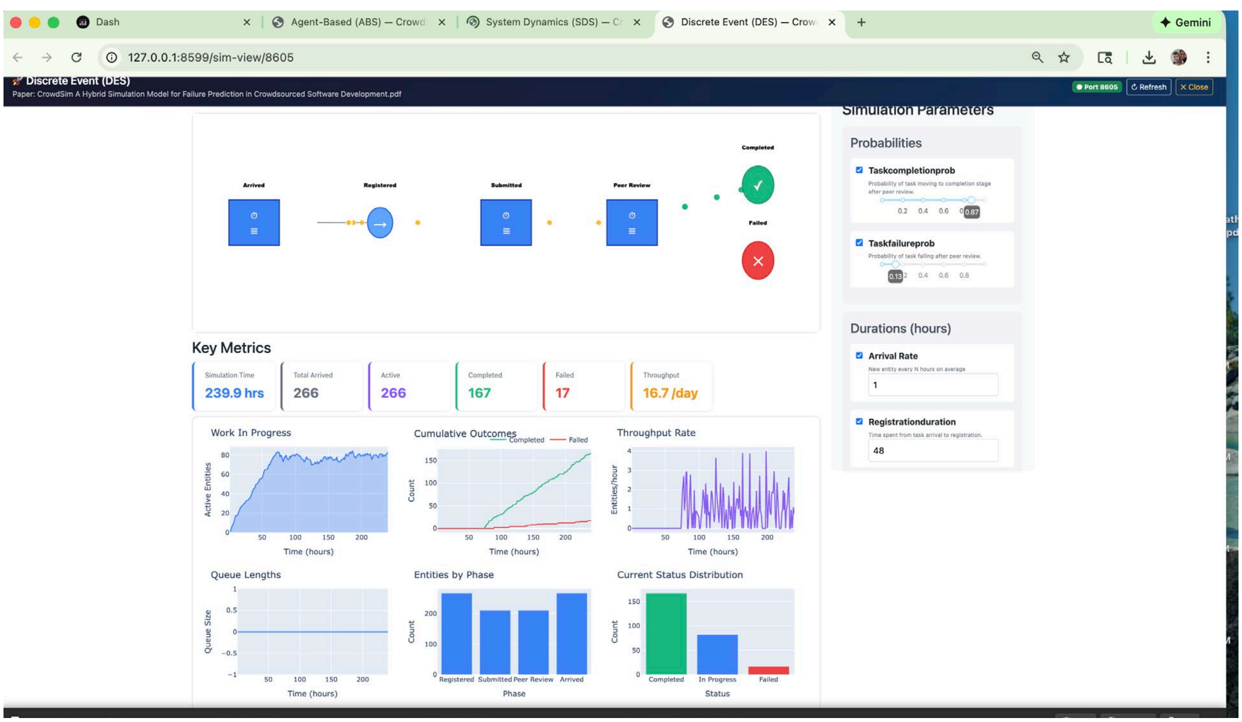

Figure 7. Discrete Event Simulation

The PTL's Database

The other key component of the PTL is the database. Table 1 provides the data tables and purpose by domain in our initial release.

Table 1. PTL Database Inventory by Domain

Domain	Table	Primary Purpose
Session Core	sessions	Master session metadata and UI flags
Session Core	Session_history	Ordered user/assistant history
Session Core	Session_params	Key-value session parameters
Session Core	Session_scenario	Scenario payload and research framing
Session Outputs	Session_research	Research paper rows and metadata
Session Outputs	session_viz	Visualiztaion specs and order
Session Outputs	Session_events	Append-only event timeline
Model Lifecycle	Generated_models	Generated model registry
Model Lifecycle	Session_models	Session-model link map
Model Lifecycle	Model_runs	Model execution runs and status
Model Lifecycle	Model_assets	Model-generated artifacts

The design strengths for the current schema include session-rooted coordination boundaries, append-only event logging via "session_events," flexible JSON-capable payload fields, and pluggable search backends throughout. The key rationale is controlled extensibility: new agent features can be introduced at the orchestration layer while the persistence contract remains stable enough for migration and audit, as shown in Figure 8.

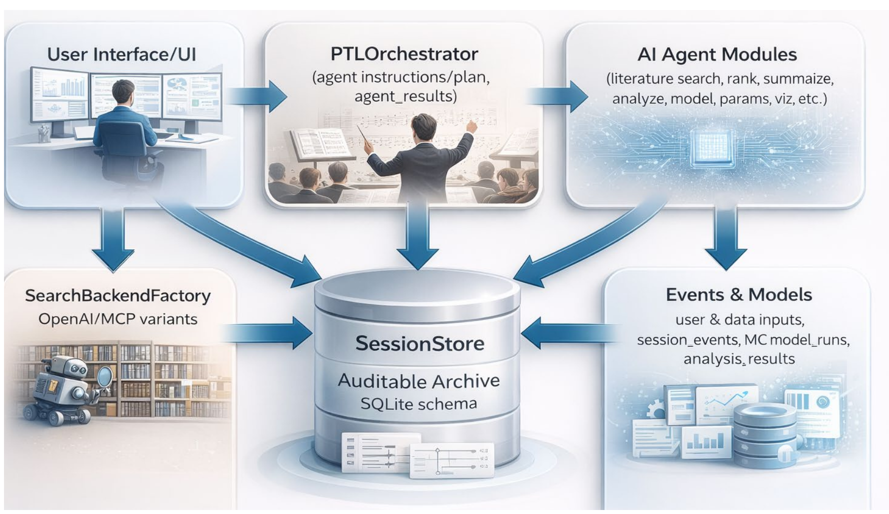

Figure 8. Multi-Agent Persistence and Auditable Coordinated Information Flow

The Enterprise Requirements & Acquisition Model

AIRC obtained from The Aerospace Corporation the latest version of the Enterprise Requirements & Acquisition Model (ERAM) based on Lt Col (Ret.) Robb Wirthlin's MIT PhD dissertation (Wirthlin et al., 2011). We plan to use ERAM to help jump start our database of models contained within the PTL to build new models that can be compared against. Further, we have already used it to benchmark a prototype of our agent-based approach.

Figure 9 shows a section out of the ERAM 2.3 model, which used the ExtendSIM tool. ERAM contains events for the entire Planning, Programming, Budgeting, and Execution (PPBE) process with constraints and representations of the progression from Pre-A phase through Phase C with funding- and governance-driven delays. Programs entering Pre-A are “gated” by funding availability and Acquisition Category (ACAT) classification, which then determines the magnitude of administrative and review delays. Additional PPBE effects are captured through preparation activities (e.g., acquisition panels, Request for Proposal coordination, etc.) and the possibility of rework loops when decisions or funding alignment fail.

We performed a preliminary comparison of the model’s ExtendSIM outputs to outputs from our AI agents decomposing ERAM 2.3 into a statistically equivalent python model was compared using a Monte Carlo run on a distribution of 500 programs across ACAT I, II, and III program types with success rate as the core validation metric. The approach used LLMs to “mimic” a distribution inside the “black box activity” in an attempt to match the output of the model. This comparison yielded an overall accuracy of 96.6% across all key metrics and demonstrates that our approach is sound, but a lot more work is needed to generalize the ERAM 2.3 model for our full needs.

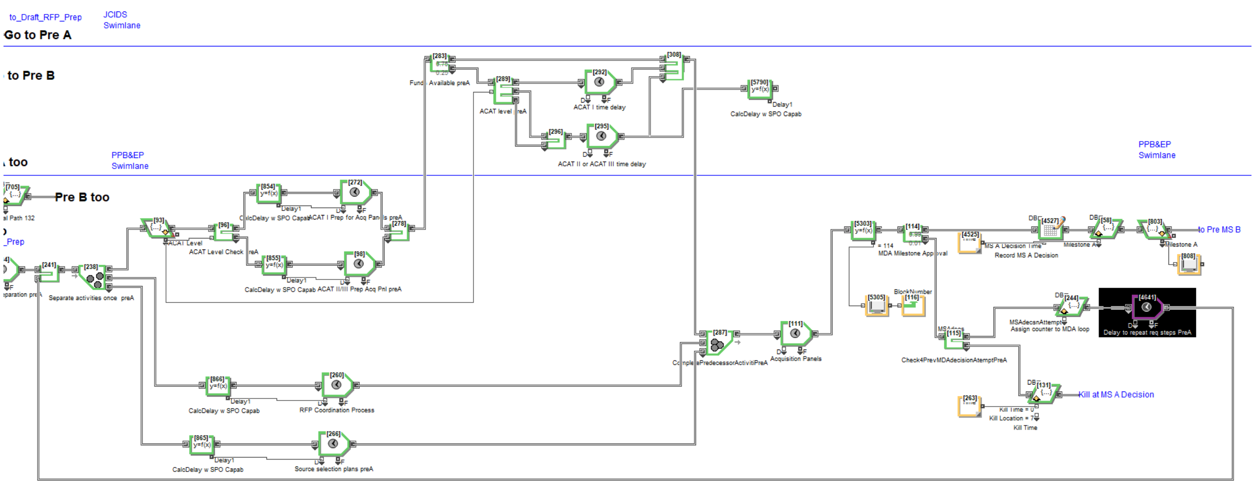

Figure 9. Aerospace’s Lt Col (Ret.) Rob Wirthlin’s MIT PhD Model, ERAM

Table 2 provides the results of the Original ERAM 2.3 model and the LLM agent generated SimPy version.

Table 2. Test Set Performance (Validation)

Metric	Original	SimPy	Delta (%)
Success Rate	53.0%	48.9%	4.1%
ICD (Average Time)	843 days	859 days	2.0%
Cost Growth	1.45x	1.39x	4.0%

Future Plans and Discussion

Decision-makers will soon have a tool that is currently under development, which promises to revolutionize the field of policy analysis using agentic AI. AIRC is well on its way to building a tool for the DoW and its stakeholders to use to try and answer perplexing policy questions. The tool is, however, built in a manner that allows it to adapt to any policy area.

Even though the PTL is still in it is infancy, the ability to research academic literature and create the standard models used by policy researchers is a first step towards a Multi-Level

Computational Policy Simulation that grows over time to provide decision-makers with a capability for doing "*What if?"* policy impact studies looking for unintended consequences.

AIRC was able to identify an Agentic AI startup company (Anadyr Horizon Inc.) with an existing modeling capability that could adapt their analytical predictive intelligence tools, and recently, our OSW leadership facilitated a meeting with CORAS.ai, which has a complimentary effort underway to build a policy document analysis tool for the Navy (see Appendix C).[4]

We anticipate that our efforts will be coordinated in the near future.

Appendix A. Rouse's Ten-Step Methodology

We provide as is, a direct quote of Rouse's ten-step methodology (Rouse 2014, pp. 36-38) for easy reference.

Step 1: Decide on the Central Questions of Interest

The history of modeling and simulation is littered with failures of attempts to develop models without clear intentions in mind. Models provide means to answer questions. Efforts to model socio-technical systems are often motivated by decision makers' questions about the feasibility and efficacy of decisions on policy, strategy, operations, etc. The first step is to discuss the questions of interest with the decision maker(s), define what they need to know to feel that the questions are answered, and agree on key variables of interest.

Step 2: Define Key Phenomena Underlying These Questions

The next step involves defining the key phenomena that underlie the variables associated with the questions of interest. Phenomena can range from physical, behavioral, or organizational, to economic, social or political. Broad classes of phenomena across these domains include continuous and discrete flows, manual and automatic control, resource allocation, and individual and collective choice. Mature domains often have developed standard descriptions of relevant phenomena.

Step 3: Develop One or More Visualizations of Relationships Among Phenomena

Phenomena can often be described in terms of inputs, processes, and outputs. Often the inputs of one phenomenon are the outputs of other phenomena. Common variables among phenomena provide a basis for visualization of the set of key phenomena. Common visualizations methods include block diagrams, IDEF, influence diagrams, and systemigrams.

Step 4: Determine Key Tradeoffs That Appear to Warrant Deeper Exploration

The visualizations resulting from Step 3 often provide the basis for in-depth discussions and debates among members of the modeling team as well as the sponsors of the effort, which hopefully includes the decision makers who intend to use the results of the modeling effort to inform their decisions. Lines of reasoning, perhaps only qualitative, are often verbalized that provides the means for immediate resolution of some issues, as well as dismissal of some issues that no longer seem to matter. New issues may, of course, also arise.

Step 5: Identify Alternative Representations of These Phenomena

Computational representations are needed for those phenomena that will be explored in more depth. These representations include equations, curves, surfaces, process models, agent models, etc. – in general, instantiations of standard representations. Boundary conditions can affect choices of representations. This requires deciding on fixed and variable boundary conditions such as GDP growth, inflation, carbon emissions, etc. Fixed conditions can be embedded in representations while variable conditions require controls such as slider bars to accommodate variations – see Step 9.

---

[4] For more information about Anadyr Horizon, we refer you to CORAS.ai's website.

Step 6: Assess the Ability to Connect Alternative Representations

Representations of phenomena associated with tradeoffs to be addressed in more depth usually require inputs from other representations and produce outputs required by other representations. Representations may differ in terms of dichotomies such as linear vs. nonlinear, static vs. dynamic, deterministic vs. stochastic, continuous vs. discrete, and so on. They may also differ in terms of basic assumptions, e.g., Markov vs. Non-Markovian processes. This step involves determining what can be meaningfully connected together.

Step 7: Determine a Consistent Set of Assumptions

The set of assumptions associated with the representations that are to be computationally connected need to be consistent for the results of these computations to be meaningful. At the very least, this involves synchronizing time across representations, standardizing variable definitions and units of measures, and agreeing on a common coordinate system or appropriate transformations among differing coordinate systems. It also involves dealing consistently with continuity, conservation, and independence assumptions.

Step 8: Identify Data Sets to Support Parameterization

The set of representations chosen and refined in Steps 5-7 will have parameters such as transition probabilities, time constants, and decay rates that have to be estimated using data from the domain(s) in which the questions of interest are to be addressed. Data sources need to be identified and conditions under which these data were collected determined. Estimation methods need to be chosen, and in some cases developed, to provide unbiased estimates of model parameters.

Step 9: Program and Verify Computational Instantiations

To the extent possible, this step is best accomplished with commercially available software tools. The prototyping and debugging capabilities of such tools are often well worth the price. A variant of this proposal is to use commercial tools to prototype and refine the overall model. Once the design of the model is fixed, one can then develop custom software for production runs.

The versions in the commercial tools can then be used to verify the custom code. This step also involves instantiating interactive visualizations with graphs, charts, sliders, radio buttons, etc.

Step 10: Validate Model Predictions, at Least Against Baseline Data

The last step involves validating the resulting model. This can be difficult when the model has been designed to explore policies, strategies, etc. for which there inherently is no empirical data. A weak form of validation is possible by using the model to predict current performance with the "as is" policies, strategies, etc. In general, models used to explore "what if" possibilities are best employed to gain insights that can be used to frame propositions for subsequent empirical study.

Summary

The logic of the ten-step methodology can be summarized as follows, with emphasis on Steps 1-7:

- Define the question(s) of interest
- Identify relevant phenomena
- Visually compose phenomena
- Identify useful representations
- Computationally compose representations

Note that this logic places great emphasis on problem framing and formulation. Deep computation is preserved for visually identified critical tradeoffs rather than the whole problem formulation. Steps 8-10 of the methodology are common to many methodologies.

Not all problems require full use of this ten-step methodology. Often visual portrayals of phenomena and relationships are sufficient to provide the insights of interest. As just noted, such views are also valuable for determining which aspects of the problem should be explored more deeply.

Appendix B. Additional selected 4+1 Architectural views for the MCPS

In this appendix we provide the remainder of the primary architectural views of the Multi-Level Computational Model.

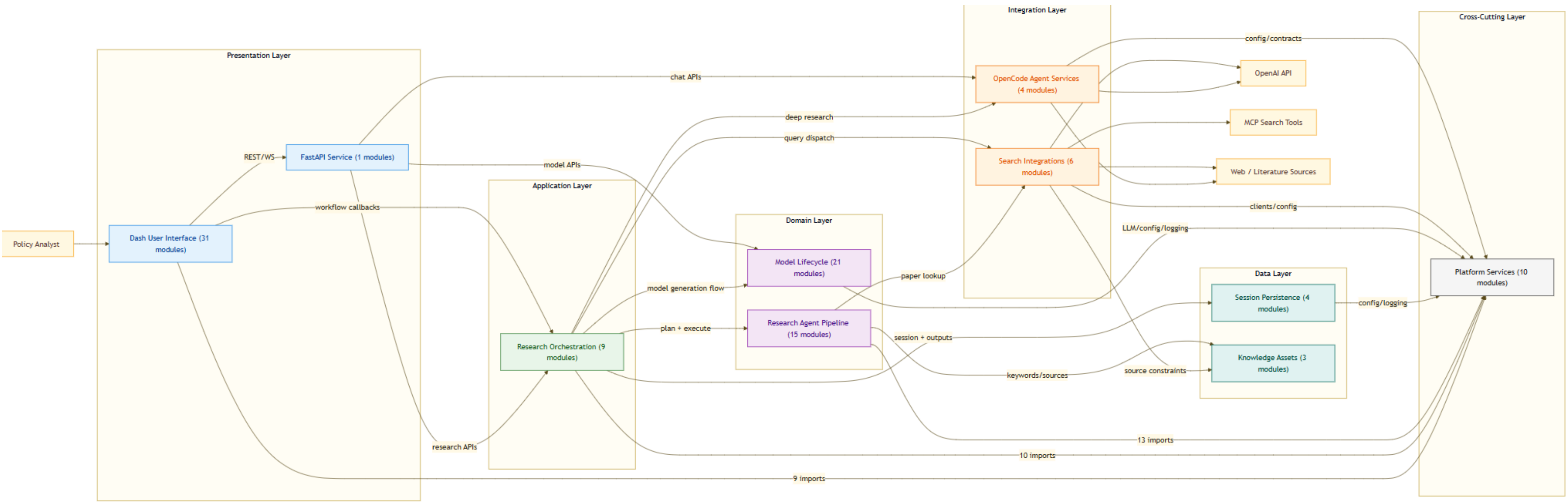

Figure 10. Logical View: Component Diagram

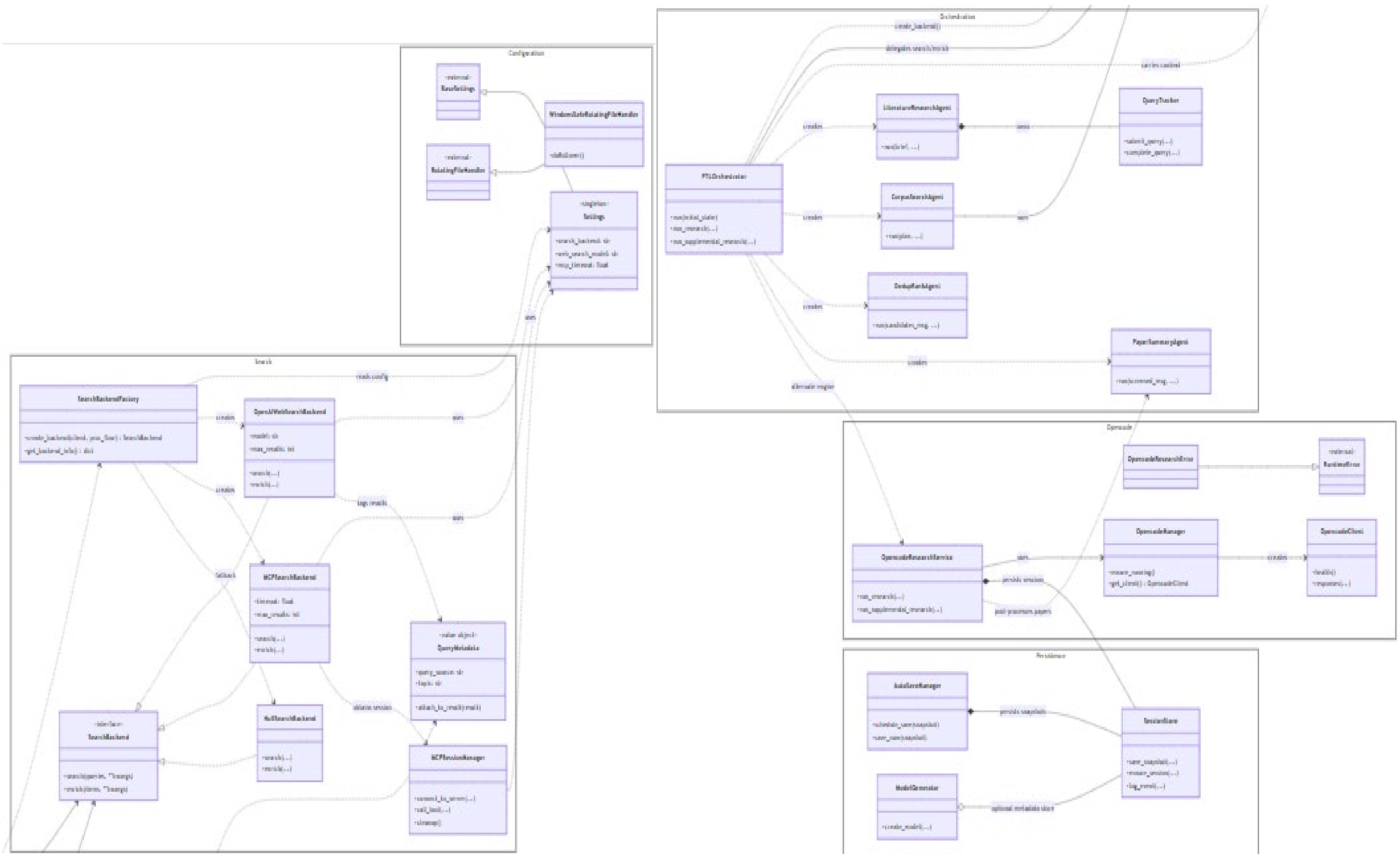

Figure 11. Logical View: Class Diagram – High Level

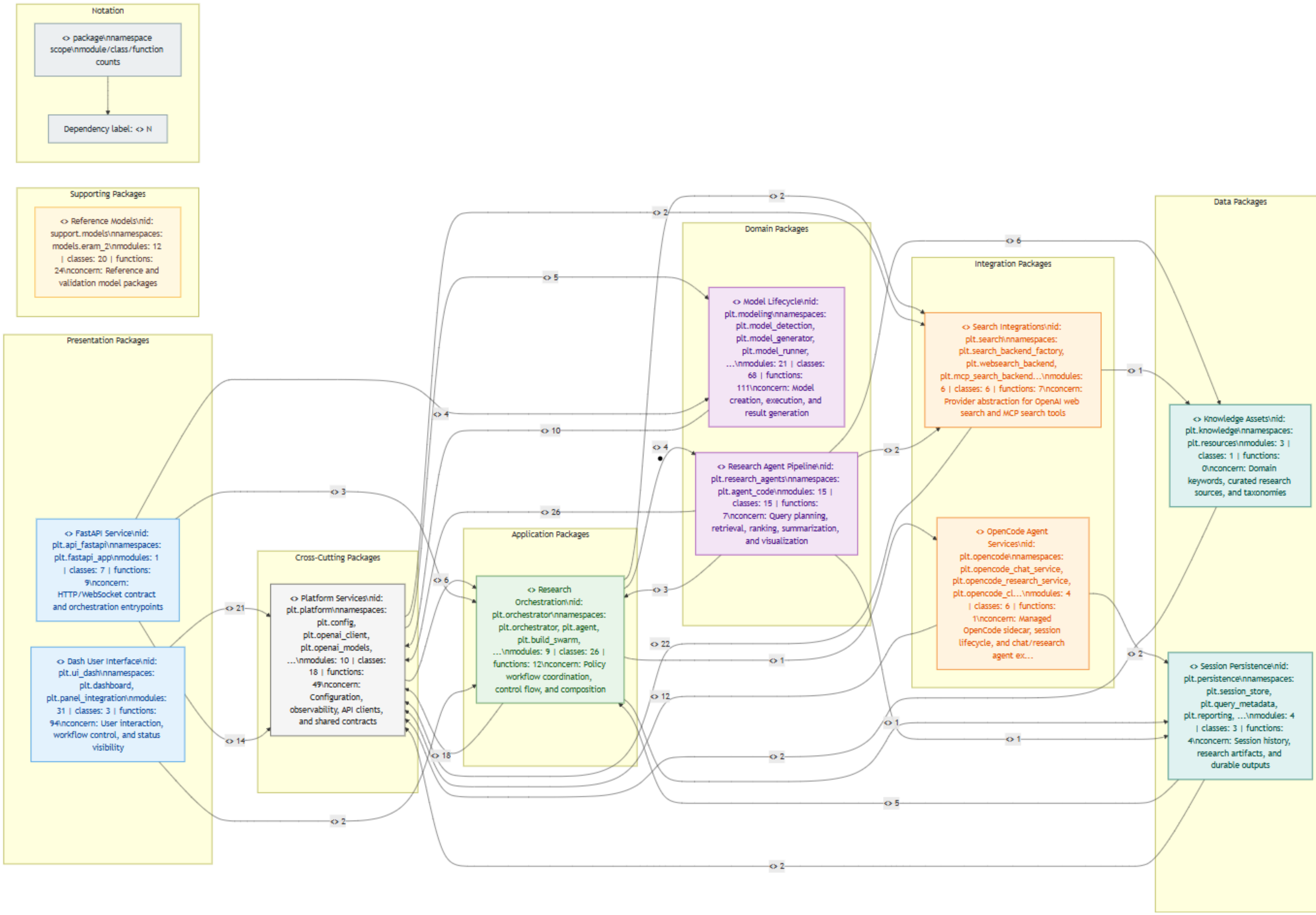

Figure 12. Development View: PTL Package Diagram

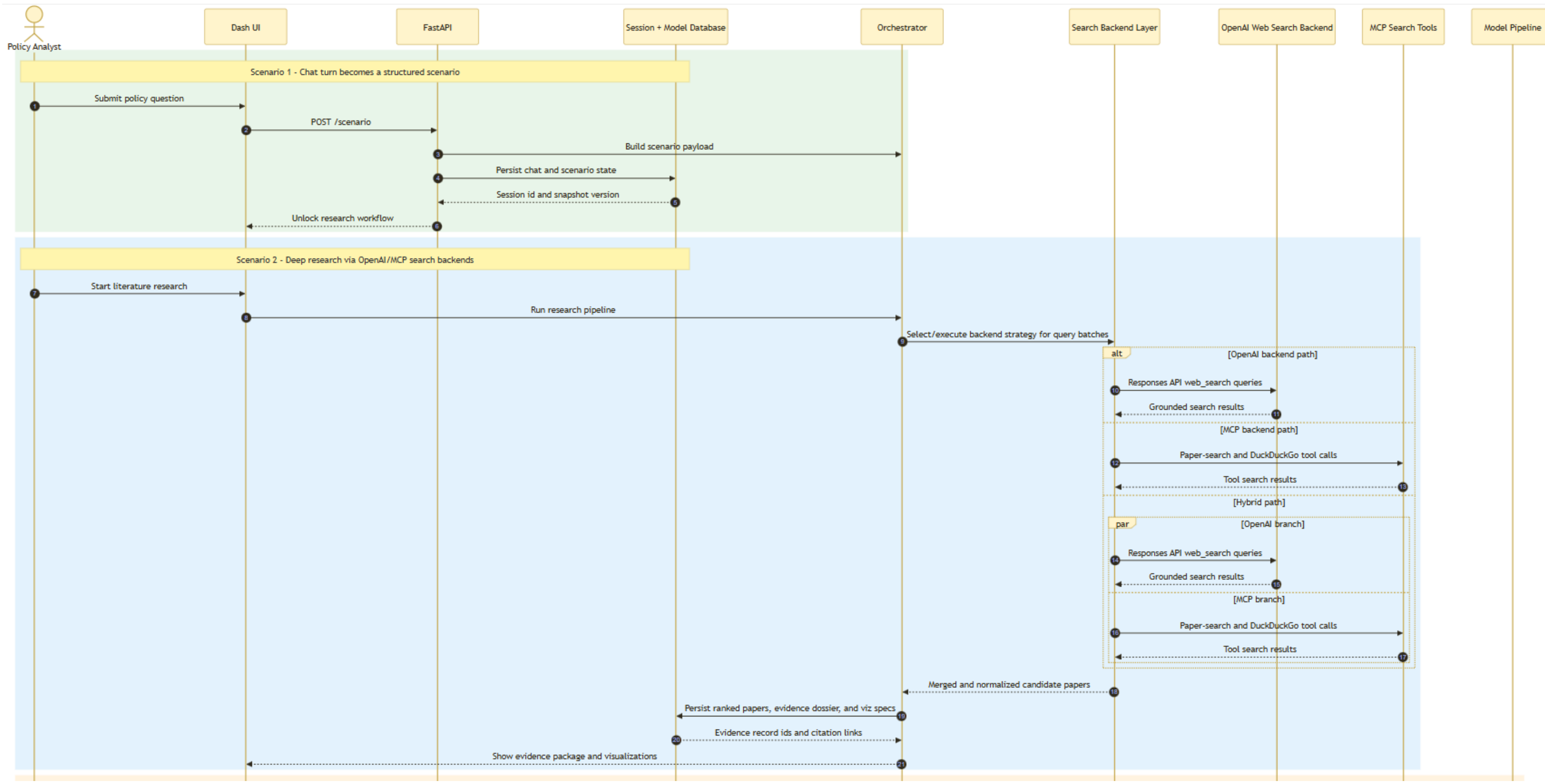

Figure 13. Scenarios: Standard Agent Pipeline Version (Top Half)

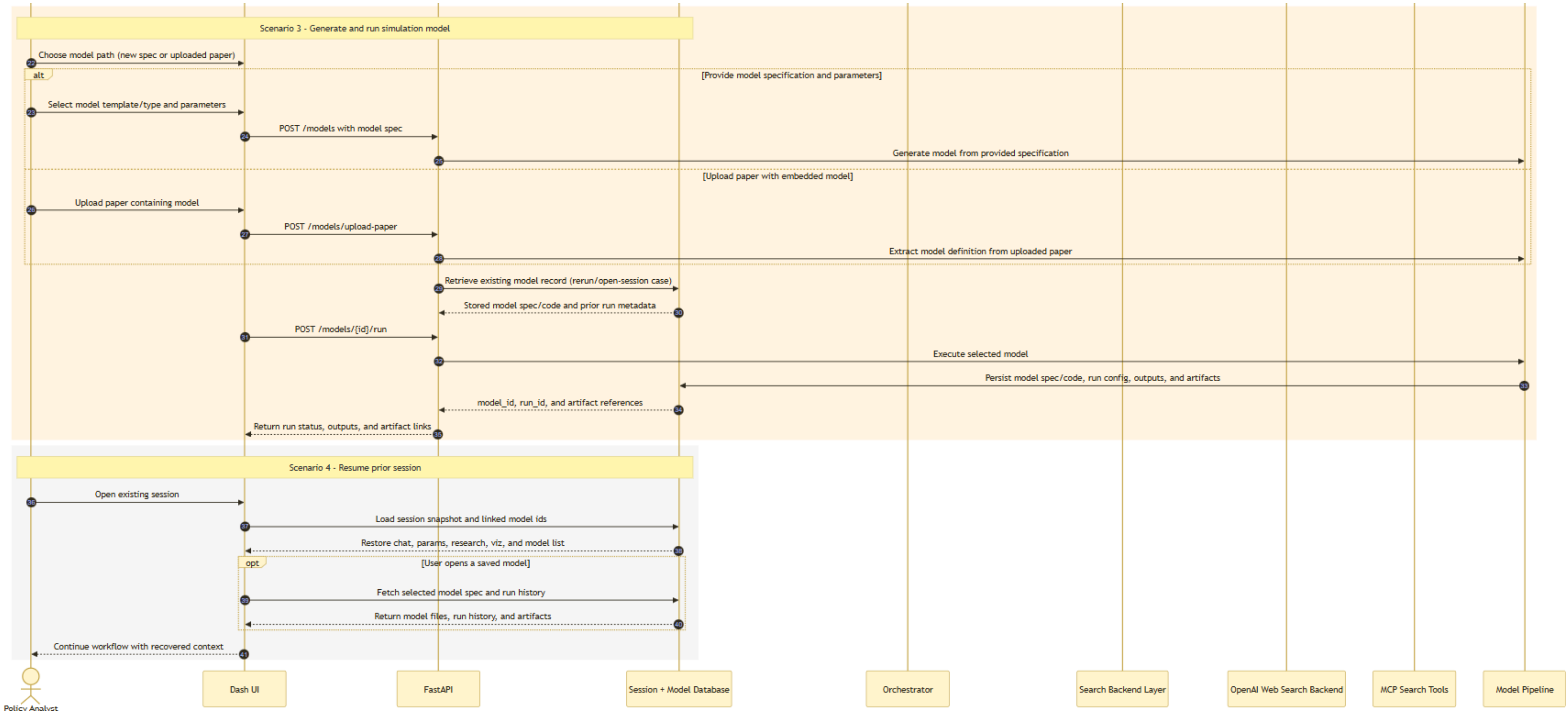

Figure 14. Scenarios: Standard Agent Pipeline Version (Bottom Half)

Appendix C. Agentic Systems Intelligence (ASI)

Simulation Infrastructure for Modeling Decision Dynamics in Geopolitical Crises

Provided courtesy of Anadyr Horizon Inc.

Problem Context

Strategic crises increasingly unfold across interconnected political, military, economic, and cyber systems. Decision-makers must navigate these environments under time pressure, incomplete information, and conflicting institutional incentives. Traditional analytic approaches—including expert judgment, static forecasting models, and episodic tabletop exercises—often struggle to capture how decisions propagate across complex systems and generate cascading effects.

Recent research in crisis simulation and negotiation systems highlights the importance of modeling **dynamic, multi-actor decision environments** rather than isolated events.

Technical Approach

Anadyr Horizon's proprietary technology, **Agentic Systems Intelligence (ASI)**, models geopolitical crises as networks of interacting decision-making agents embedded within strategic systems.

Each agent represents a real-world actor whose behavior is parameterized using historical data, behavioral research, and multi-level constraints. The system integrates agent-based modeling, probabilistic simulation, and scenario stress testing to generate large numbers of potential decision trajectories under defined crisis conditions.

Outputs and Applications

ASI enables analysts to explore how political signaling, military actions, sanctions, or misperceptions may propagate through a system of actors. **North Star**, Anadyr's ASI-powered software platform, generates probability-weighted escalation pathways, identifies potential tipping points and intervention windows, and produces scenario trees illustrating alternative crisis trajectories.

Applications include crisis planning, escalation analysis, red-team exercises, and geopolitical risk modeling. ASI does not attempt deterministic prediction. Instead, it produces probabilistic decision trajectories designed to inform expert judgment, strategic planning, and policy analysis.

Acronyms and Abbreviations

ABM	Agent-Based Model
AFRL	Air Force Research Laboratory
AI	Artificial Intelligence
AIRC	Acquisition Innovation Research Center
ASI	Agentic Systems Intelligence
C2	Command and Control
DES	Discrete Event Simulation
DLA	Defense Logistics Agency
DoD	Department of Defense
DoW	Department of War

DPCAP	Defense Pricing, Contracting, and Acquisition Policy
DPI	Dots Per Inch
DTIC	Defense Technical Information Center
ERAM	Enterprise Requirements & Acquisition Model
JSON	JavaScript Object Notation
LLM	Large Language Model
MCPS	Multi-Level Computational Policy Simulator
MIT	Massachusetts Institute of Technology
MITRE	MITRE Corporation
OV-1	Operational View
PTL	Policy Test Lab
RAND	RAND Corporation
R&D	Research and Development
RDT&E	Research, Development, Test, and Evaluation
SERC	Systems Engineering Research Center
SDM	System Dynamics Model
UARC	University-Affiliated Research Center
UI	User Interface
USD	Under Secretary of Defense
USD(A&S)	Under Secretary of Defense for Acquisition and Sustainment

Acknowledgements

We would like to thank SERC's Mr. Tom McDermott and Ms. Valerie Sitterle for guidance, as well as Ms. Stephanie Halcrow, Dr. Arun Seraphin, and Mr. Moshe Schwartz for the numerous conversations that led the team to investigate the use of the rapidly evolving Agentic AI (Large Language Model–based agents) to solve this challenging problem. We also gratefully acknowledge Dr. Mo Mansouri and Rai Singh Bindra for obtaining results from ERAM, The Aerospace Corporation's Andy Schickling for his support in obtaining ERAM, and Dr. Dan Houston for his help with running it.

Importantly, we want to thank our OSW(A&S)/API sponsors Mr. David Cadman, Ms. Anne Twist, and Ms. Nickee Abbott for their support.

The views, findings, and conclusions in this document are solely those of the authors and do not necessarily reflect the views or positions of the U.S. government (including the DoW or any other government personnel) or the Stevens Institute of Technology.

References

DoW. (2025). *Acquisition transformation strategy: Rebuilding the arsenal of freedom*. https://media.defense.gov/2025/Nov/10/2003819441/-1/-1/1/ACQUISITION-TRANSFORMATION-STRATEGY.PDF

Kannan, H., Rouse, W., Merchant, N., Salado, A., Son, Y., & Szajnfarber, Z. (2022). *The Acquisition Innovation Research Center: Innovation for digital transformation and policy analytics*. Acquisition Innovation Research Center. https://acqirc.org/publications/research/the-acquisition-innovation-research-center-innovation-for-digital-transformation-and-policy-analytics/

Rouse, W. B. (2014). *Visualization of phenomena: Policy flight simulators, modeling and visualization of complex systems and enterprises* (1st ed.). Wiley.

Rouse, W. B., & Bodner, D. A. (2012). *Multi-level modeling of socio-technical systems* (Technical Report No. SERC-2012-TR-020-1). Systems Engineering Research Center. https://sercproddata.s3.us-east-2.amazonaws.com/technical_reports/reports/1524587680.pdf

Rouse, W. B., & Bodner, D. A. (2013a). *Multi-level modeling of socio-technical systems: Phase 1* (Technical Report No. SERC-2013-TR-020-2). Systems Engineering Research Center. https://sercproddata.s3.us-east-2.amazonaws.com/technical_reports/reports/1524587866.pdf

Rouse, W. B., & Bodner, D. A. (2013b). *Multi-level modeling of socio-technical systems: Volume 2* (Technical Report No. SERC-2013-TR-020-3). Systems Engineering Research Center. https://sercproddata.s3.us-east-2.amazonaws.com/technical_reports/reports/1530802703-SERC-2013-TR%20020-3%20Multi%20Level%20Modeling%20Socio%20Technical%20SystemsVolume%202_RT%20044.pdf

Rouse, W. B. Naylor, M. D. PhD, Yu, Z., Pennock, M. J., Hirschman, K. B., Pauly, M.V., & Pepe, K. M. (2019, July–August). Policy flight simulators: Accelerating decisions to adopt evidence-based health interventions. *Journal of Healthcare Management*, *64*(4): 231–241. https://www.doi.org/1097/JHM-D-18-00114

Wirthlin, Joseph R., Houston, Daniel X., & Madachy, Raymond J. (2011). Defense acquisition system simulation studies. *Proceedings of the 2011 International Conference on Software and Systems Process*. https://hdl.handle.net/10945/60714

Acquisition Research Program
Department of Acquisition, Finance, and Manpower
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net\