ODRAS: Origin and One-Page White Paper

How ODRAS Was Conceived (Concise History)

While supporting Navy ADT and the Weapons Integration Risk Reduction (WIRR) effort as a systems and digital engineering partner, it became clear that meetings often lacked synthesis. Airframe vendors pitched their preferred solutions, and the Navy team spoke in general terms about desired outcomes. What was missing was a structured, defensible conceptual system derived directly from driving documents → the Capabilities Development Document (CDD), Initial Capabilities Document (ICD), and Analyses of Alternatives (AoAs) ← at a stage when requirements are still malleable and must be extracted rather than dictated.

With the broad availability of large language models and study of ontology-driven methods, a practical idea emerged: use a systems-engineering ontology to guide probabilistic requirement extraction and generate confidence-weighted conceptual system options, not just a single, brittle result. I (J.K. DeHart) championed the approach. An early prototype, reqFlow, demonstrated feasibility even without formal ontology tooling or a UI and received strong interest among peers, particularly for ontology-guided LLM interactions and confidence-weighted alternatives. These signals led to formalizing the concept as ODRAS, an Ontology-Driven Requirements Analysis System designed to bootstrap credible system estimates in minutes and give teams a structured starting point instead of starting from a blank page.

One-Page White Paper

Problem

Early program discussions and vendor engagements frequently “just talk” around needs. Current practice relies on manual recollection and modeling sessions (e.g., in Cameo) to capture requirements that actually live in lengthy source documents. This slows kickoff, obscures traceability, and risks biasing outcomes toward whoever is in the room rather than what the documents and mission analysis support.

Solution

ODRAS (Ontology-Driven Requirements Analysis System) reads authoritative inputs such as CDDs, ICDs, and AoAs, uses a systems-engineering ontology to orchestrate LLM-based extraction, and produces a traceable conceptual system estimate: components, functions, processes, and interfaces. Rather than a single output, ODRAS can yield multiple candidate architectures with confidence weighting, giving decision-makers immediate options grounded in source text. The result is a defensible starting point that accelerates analysis, improves synthesis, and keeps teams anchored to documented needs.

How It Works

ODRAS ingests program documents and performs requirement candidate extraction aligned to an ontology. It normalizes terms, links statements to ontology concepts, and constructs a concept graph. From this graph, it estimates system structure and behavior and exports to a graph database for visualization or into downstream modeling environments. The process runs in minutes on commodity hardware and can operate air-gapped to meet security constraints.

Why Now

LLMs are now reliable enough, when constrained by an explicit ontology, to extract and organize requirement candidates with useful precision and recall. Programs are under schedule pressure, yet source documents keep growing. ODRAS turns that document load into an advantage by rapidly bootstrapping a structured baseline for engineering discussion and trade studies.

Defense acquisition tempos are compressing, from traditional 20–25-year cycles toward 5-year cadences reminiscent of the 1950s–1960s Century Series, necessitating tools that convert documents into decision-quality models in days, not months. ODRAS is designed for this pace.

Early Results

In 10/2023, the reqFlow prototype extracted 435 requirement candidates from the FLRAA CDD and bootstrapped a conceptual architecture with components, processes, functions, and interfaces. The run completed in about five minutes on a laptop with a small GPU, fully air-gapped. Outputs were loaded into a graph database for visualization and enabled a more structured discussion about options and gaps.

Fit with Existing Processes

ODRAS complements, rather than replaces, tools like Cameo. It supplies a fast, traceable baseline that modelers can refine, reducing the time from kickoff to a first viable model and giving engineers a defensible starting point rooted in the CDD/ICD/AoA. It also improves objectivity by presenting confidence-weighted alternatives rather than a single favored design.

Risks and Mitigations

Adoption by non-technical engineers is a concern. ODRAS addresses this with a simple user experience, guided workflows, and transparent traceability back to source text. The “tool in search of a need” risk is mitigated by piloting with real programs and aligning outputs with WIRR and ADT artifacts, ensuring immediate utility in current review and decision forums.

Next Steps

Run a focused pilot with a Navy ADT team: ingest the program’s CDD/ICD, calibrate the ontology with SMEs, generate candidate system estimates, and compare time-to-first-model and traceability quality against current practice. Use pilot feedback to finalize the UI and ontology management features and establish an integration path into existing modeling toolchains.

Appendix: Overarching Decision Process

and ODRAS’ Role

This appendix outlines a disciplined chain from ontology to decision and feedback. It clarifies responsibilities and artifacts at each step and shows where ODRAS accelerates synthesis, preserves traceability, and enables rapid iteration.

ODRAS primarily covers steps 1–3; steps 4–7 run in the broader enterprise. In Data (step 2), its role is a limited demonstration given enterprise-scale scope. ODRAS also serves as a process test bed across these steps.

Step	Description	ODRAS’ role
Process & Governance (cross-cutting)	Defines workflow, roles, entry/exit criteria, change control, and iteration cadence spanning all steps.	Limited: program-owned; ODRAS provides traceability and audit artifacts.
1. Ontology	Develop/curate SE ontology encoding mission concepts, system elements, functions, interfaces, constraints, and MOEs/MOPs.	Primary: ontology-led extraction and mapping (piloted in ODRAS).
2. Data	Aggregate/govern sources (CDDs, ICDs, AoAs, CONOPS, TTPs, standards); normalize, version, and preserve provenance.	Primary: document ingestion and provenance linking (piloted).
3. Models	Translate ontology-anchored statements into conceptual models: components, functions, processes, interfaces, constraints; prepare exports to MBSE/graph.	Primary: bootstrap initial conceptual models and variants (piloted).
4. Simulation	Instantiate executable surrogates of conceptual models; exercise with mission threads and scenarios to explore	No direct role: external simulation tools; ODRAS

Step	Description	ODRAS’ role
	behaviors and trades.	exports candidates.
5. Analysis	Evaluate alternatives vs objectives, constraints, and MOEs/MOPs; quantify gaps/risks/sensitivities; assess requirement quality/conflicts.	Limited: provides traceable baselines; analysis runs in external toolchain.
6. Decision	Select architectures, increments, and mitigations with evidence traceable to documents and analyses.	No direct role: decision authority; ODRAS informs via artifacts.
7. Decision Field Feedback	Capture operational/test feedback; update data and ontology; re-run extraction and generation to keep the baseline current.	Limited: re-ingests new artifacts and regenerates; broader system manages.