ANNUAL
CONFERENCE & EXPO 2026

GenAI Workbench

AI-Assisted Analysis and Synthesis of Engineering Systems from Multimodal Engineering Data

Hasan Sinan Bank   Daniel R. Herber
Department of Systems Engineering, Colorado State University

IISE Annual Conference & Expo 2026

Section 1

Introduction

Motivation: The Fragmented Engineering Stack

- Modern engineering design platforms excel at **discipline-specific** (e.g., eCAD, mCAD) and **task-specific** (e.g., CAD, CAM, CAE) tools, but lack native systems engineering frameworks
- System-level **requirements** and **architectures** typically live in separate tools of their own — distinct from the detailed component design tools

- Consequences:
  - Information silos that *break the digital thread*
  - Verification & validation fragmented across the product lifecycle
  - Increased integration risk and late-stage rework

Motivation: this fragmentation motivates **GenAI Workbench** — an integrated, comprehensive, and intelligent paradigm for systems engineering.

Research question

How can we bridge the gap between unstructured requirements, physical geometry, and system-level architecture — all within the designer's primary workflow?

Publication-driven validation: the viability of the workbench is argued step by step through a sequence of papers, each covering one facet of the framework:

- **CatalogBank** [Bank2024CatalogBank] — structured catalog dataset & semi-automatic annotation tool (*DocumentLabeler*) for engineering documents (mCAD-leaning instantiation)
- **Retrieval-Bench** [Bank2026RetrievalBench] — extends the CatalogBank notion into **eCAD**: multimodal dataset & retrieval benchmark across *BOMs, schematics, and netlists* on OSHWA-certified hardware *(under double-blind review at ICLAD 2026)*
- **GenAI Workbench** [Bank2026GenAIWorkbench] — integrated conceptual framework *(this talk)*
- **Semantic–Physical–Relational linking** [Bank2026SemanticPhysical] — multimodal linking mechanisms and alignment metrics
- **Architecture Synthesis** [Bank2026ArchitectureSynthesis] — text → DSM generation from requirements
- **Design-OS** [Bank2026DesignOS] — specification-driven framework for engineering system design

Central hypothesis: embedding AI-driven systems engineering capabilities directly into the designer's workflow establishes a **unified digital thread** that enhances engineering effectiveness.

Proposed instrument: the **GenAI Workbench** — a methodological proof-of-concept.

Section 2

Background

- **AI-driven requirements engineering**
  - LLMs (GPT-4 [OPENAI2023GPT4], LLaMA-2 [TOUVRON2023LLAMA2]) enable zero/few-shot extraction, but accuracy is limited (∼45–52% for formal spec extraction [LI2025SPECSLLM])
  - Two-step "annotate-then-convert" pipelines [LI2025SPECSLLM] mitigate hallucinations by narrowing the task
  - Motivates a **multi-stage refinement** approach
- **Vision-language models on engineering documents**
  - Hybrid YOLO + transformer parsers (e.g., Donut) extract dimensions and tolerances from drawings (93.5% F1 [KHAN2025DRAWINGS])
  - Fine-tuned VLMs currently outperform general-purpose multimodal LLMs on technical content [GARCIA2023TECHMB]
- **Emerging MBSE Co-Pilot visions** [ZHANG2025MBSE]
  - GPT-4 integrated into SysML environments to auto-generate model elements [LONGSHORE2024LEVERAGING]
  - Feeding MBSE models into LLM assistants reduces hallucinations in requirement generation [PATEL2024EASING,GARCIA2024OPM]

Graph-Based Integration and the Multimodal Gap

Graph-Based Integration

- **Graph databases for MBSE** (e.g., Neo4j-backed SysML stores [SCHUMMER2022GRAPHMBSE]) enable traversal queries that traditional SysML tools struggle with
- **Intercax Syndeia** [SYNDEIA2021] and **OpenMBEE** [OPENMBEE2018] provide federated graph links across CAD, requirements, and simulation
- **Lifecycle-artifact graphs** [HEDBERG2020USING] unify requirements, CAD parts, and process plans as typed edges across the digital thread

Multimodal Gap

- These tools link artifacts; they do *not* link **semantic content** to **physical features**
- Standards such as OSLC [OSLC2021CORE] define URI-based linking across tools, but leave **semantic interpretation** to the engineer
- **Our target:** link requirements directly to the elements of physical viewpoint (e.g., geometric features), not to CAD files as opaque artifacts

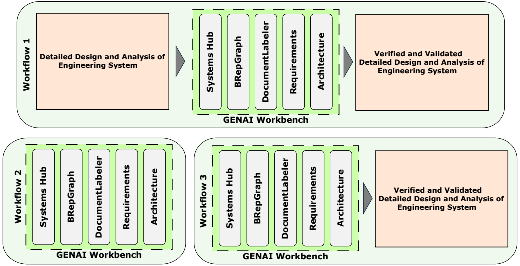
Important

Open gap: Bridging *semantic* (text) $\leftrightarrow$ *geometric* (B-rep) $\leftrightarrow$ *relational* (graph) data through a unified MBSE workflow.

Section 3

Methodology

GenAI Workbench — Proposed AI-Assisted Workflow



The figure shows **three complementary workflows**:

- **(1) CAD-to-CAD mediator** — Workbench acts as an *interface for CAD*: existing design in, processed (verified & validated) design out — a geometry-level interop bridge *with semantics*
- **(2) Standalone Workbench** — used on its own (no external CAD I/O); engineers work *inside* the Workbench via its components (SystemsHub, DocumentLabeler, BRepGraph, Requirements, Architecture)
- **(3) Workbench → CAD** — Workbench as the *source*: synthesizes from internal documents/requirements and outputs a verified & validated detailed design to CAD

Interoperability hinges on open formats (STEP/AP242, PDF, JSON) and the PLM/CAD backbone's programmable API.

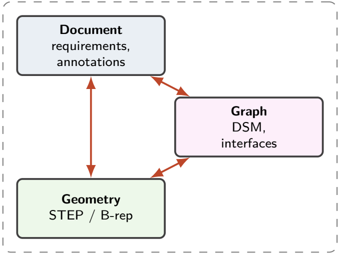
PLM/CAD Backbone

- Programmable API (Python) for geometry and automation
- STEP/AP242 with **B-rep** access via an open geometry kernel
- Extensible for custom workbenches

Multimodal Linking

- A **combination** of complementary mechanisms ties *Document*, *Geometry*, and *Graph* together — no single identifier carries the whole load

Systems Hub – multi-modal linking



Remark

Multimodal linking lets a requirement change ripple through geometry and architecture — details in [Bank2026SemanticPhysical].

Document Ingestion

- xLM (and OCR) extraction of raw text (and images) from source PDFs/specs
- Prompt-guided identification of requirement statements
- Each requirement stored as a first-class PLM object, trace-linked to its source text
- Automatic compilation of a glossary of key terms

Multimodal Linking

- Binds each component's *semantic*, *geometric*, and *relational* representation
- Combination of complementary mechanisms rather than a single identifier
- Current implementation and evaluation: [Bank2026SemanticPhysical] (in preparation)

Remark

Architecture synthesis (text → DSM) is under review at the *Wiley Journal of Systems Engineering* [Bank2026ArchitectureSynthesis], where we evaluated a range of proprietary and open-source LLMs across *hundreds of experiments* — not part of the current workflow slice.

Human-in-the-Loop Refinement

- Engineer reviews requirements: edits phrasing, corrects misinterpretations, adds missing items — PLM objects updated in place
- Reviews and edits the multimodal links (component ↔ geometry ↔ graph) through the UI [Bank2026SemanticPhysical]
- Behavior and script refinement follows the specification-driven loop of [Bank2026DesignOS]
- **AI jumpstarts the tedious work; the human validates domain knowledge**

Linking with CAD

- Each confirmed system component is tied to its CAD part/assembly
- One-to-one correspondence between architecture nodes and CAD B-rep entities
- Enables bidirectional traceability: *requirement* ↔ *component* ↔ *geometric feature*

Remark

The multimodal data model ties *Document*, *Geometry*, and *Graph* together through a combination of linking mechanisms — the technical backbone of the digital thread.

Core Components: Implementation Architecture

Project Database — SQLAlchemy persistence root, accessed through the hub.

Systems Hub — UID-linked registry and cross-modal index.

DocumentLabeler & **BRepGraph** — parallel extractors: semantic content from PDFs/images [Bank2024CatalogBank] and face-adjacency graphs from STEP B-Rep geometry; both feed the hub.

Requirements — first-class verification objects linked to architecture elements.

Architecture — hierarchical decomposition supporting states and analysis scripts.

Canvas — shared node-graph editing interface between Requirements and Architecture.

Cross-Modal Compatibility

- **Geometric** compatibility: spatial alignment and dimensional consistency between connected parts
- **Functional/Behavioral** compatibility: parameters, capacities, operating ranges (from document annotations)
- **Relational** compatibility: graph-level consistency of connection types

Formal Verification

- Natural-language engineering constraints translated into **formal specifications** — e.g., grammars, temporal logics, symbolic AI techniques
- System models evaluated against derived specifications for **behavioral** and **temporal** properties
- Extends verification *beyond* pairwise checks

Important

Continuous V&V is a product of the integrated data model, not an afterthought.

Section 4

Applications

- UID-based integration makes end-to-end traceability **practical** rather than theoretical:
  - Click a CAD feature → see the requirements it satisfies
  - Query "*show all components affected by this requirement change*" and receive an accurate, automatically-maintained answer
- **Accelerating early-stage architecture exploration**
  - Weeks of manual decomposition collapse into hours of AI-assisted iteration
  - Architecture generation becomes a **parameterizable, repeatable, comparable** computational task
  - Early-phase decisions have the highest leverage — a broader design space earlier exposes superior architectures that would otherwise be missed

Walk-through: Roller Runout Inspection

- **One assembly, five panels** — the same Misumi catalog entry flows through the whole workbench.
- **1. DocumentLabeler** — vendor PDFs ingested and tagged with the target schema (e.g., FBS, SysML, or custom).
- **2. Systems Hub** — documents, images, geometries, and other assets centralized for easy access.
- **3. BRepGraph** — parts as nodes; edges coloured by *intersecting* / *touching* / *adjacent* / *contains*.
- **4. Architecture** — whole/part block diagrams, graph, DSM-like, and list representations.
- **5. Requirements** — 18 shall-statements (L1–L3, Functional / Performance / Interface) seeded from the design.
- Same pipeline can be extended to different domains, identical flow.

Section 5

Conclusion

Contributions

- A **conceptual framework** for integrated, multimodal systems engineering
- A **unified multimodal data model** based on UIDs that links semantic, geometric, and relational representations of the same system
- Three proposed **AI-assisted workflows** for analysis, synthesis, and generation of system models
- A **hierarchical V&V framework** combining cross-modal rule checks with formal verification
- An **implementation strategy** using an open-source PLM/CAD stack and an accessible geometry kernel

Remark

The GenAI Workbench is a *scientific instrument*: its purpose is to demonstrate that the proposed integration is computationally feasible and methodologically sound — *for now* not to replace the commercial packages.

Current Limitations

- **Metrics & experimentation:** the proposed workflows still need broader empirical evaluation
- **Human studies:** formal user studies with engineering designers have not yet been conducted
- **Scope of validation:** so far focused on catalog-style documents; industrial requirement documents remain to be exercised

Future Work

1. **Implementation and validation:** complete the proof-of-concept and run user/case studies
2. **Models:** adopt **spatially graph-aware** approaches (e.g., graph-aware transformers) on the geometry side of the pipeline
3. **Agentic synthesis and verification:** autonomous Python analysis scripts inside the workbench, moving beyond human-in-the-loop toward more autonomous verification

Thank you!

Questions & discussion welcome

GenAI Workbench

AI-Assisted Analysis and Synthesis of Engineering Systems from Multimodal Engineering Data

H. Sinan Bank & Daniel R. Herber

Department of Systems Engineering, Colorado State University

sinan.bank@colostate.edu

- arxiv.org/abs/2603.00251 *(GenAI Workbench preprint)*
- arxiv.org/abs/2602.16715 *(Architecture Synthesis preprint)*
- arxiv.org/abs/2603.20151 *(Design-OS preprint)*
- doi.org/10.1145/3685650.3685665 *(CatalogBank, DocEng '24)*