American Journal of Interdisciplinary Studies

Volume: 7; **Issue:** 2
Pages: 53-72
eISSN: 3067-5146

Generative PLM: Utilizing Large Language Models for Automated Engineering Change Management and Requirements Traceability: A Case Study in Smart Manufacturing

Md Syeedur Rahman[1]; Sharmin Rahman[2];

[1]. *Solution Developer, Plant Design I/O, Gonzales, Louisiana, USA; Email: syeed_rahman@live.com; https://orcid.org/0000-0001-8475-0615*

[2]. *Department of Electrical & Computer Engineering, Lamar University, Beaumont, Texas, USA; Email : sharmin.rahman89@outlook.com https://orcid.org/0009-0007-3444-9040*

Doi: 10.63125/xbj93w89

Received: *07 March 2026;* ***Revised:*** *14 April 2026;* ***Accepted:*** *11 May 2026;* ***Published:*** *02 June 2026;*

Abstract

Generative large language models are emerging as a promising enabler for digital transformation in product lifecycle management, particularly in engineering change management and requirements traceability. This paper presents a case study in smart manufacturing that explores how generative PLM can support the automation of change request interpretation, impact analysis, and bidirectional traceability between engineering requirements and design artifacts. The proposed approach integrates large language models with structured PLM workflows to assist engineers in extracting relevant entities, identifying affected components, and linking changes to upstream requirements and downstream implementation records. In the case study, the method is evaluated in a smart manufacturing environment where frequent design revisions and cross-disciplinary dependencies create substantial coordination overhead. Results indicate that LLM-assisted workflows can reduce manual effort, improve consistency in traceability maintenance, and accelerate response time for engineering changes, while also highlighting the need for human oversight in safety-critical and ambiguous cases. The findings suggest that generative PLM can complement existing engineering information systems by improving adaptability, knowledge reuse, and decision support across the product lifecycle. Overall, this study contributes a practical framework for applying large language models to engineering change management and requirements traceability in Industry 4.0 settings.

Keywords

Generative PLM; Large Language Models; Engineering Change Management; Requirements Traceability; Smart Manufacturing; Product Lifecycle Management; Industry 4.0; Case Study.

INTRODUCTION

BACKGROUND: LARGE LANGUAGE MODELS IN DESIGN AND MANUFACTURING

The rapid development of large language models (LLMs) has stimulated growing interest in their use across design and manufacturing workflows. Recent studies indicate that these models may support a range of activities, including concept generation, transformation of design intent into manufacturing instructions, evaluation of design alternatives, and performance-driven design space exploration (Makatura et al., 2024). Broader surveys of the field likewise suggest that generative AI has begun to reshape discussions of manufacturing productivity, quality control, and process support, while also raising questions about the limits and risks of relying on such systems in engineering contexts (Doanh et al., 2023; Li et al., 2024). Within this emerging literature, LLMs are increasingly framed not merely as text generators, but as tools with potential relevance to engineering knowledge work and decision support throughout product realization processes (Kampik et al., 2024; Makatura et al., 2024b).

At the same time, the literature also emphasizes that current capabilities remain uneven. Studies of LLMs in engineering design report both opportunities and limitations, and they identify tasks and workflow stages that are especially brittle and require additional investigation (Chiarello et al., 2024; Makatura et al., 2024b). In design and manufacturing, existing work has begun to examine how generative AI may assist with computer-aided design, iterative refinement of outputs, and broader workflow integration (Daareyni et al., 2025), while other contributions stress the need to adapt general-purpose models to domain-specific tasks rather than assuming that out-of-the-box systems will be sufficient (Jiang et al., 2025). These observations collectively suggest that the central challenge is no longer whether LLMs can be applied to engineering work in principle, but how they can be incorporated in ways that are reliable, task-appropriate, and operationally meaningful.

DYNAMIC REQUIREMENTS AND TRACEABILITY

A parallel stream of research has examined requirements engineering and traceability, where natural language requirements remain difficult to manage at scale. Prior work notes that engineering requirements and related upstream and downstream tasks grow in number and complexity, and that these requirements are typically expressed in natural language, requiring substantial human expertise to interpret and maintain (Norheim et al., 2024). Related studies show that language-model-based techniques can help identify incompleteness in requirements by predicting missing terminology, although such approaches require filtering and careful evaluation to reduce noise (Luitel et al., 2024). More recent work also suggests that LLMs can support tasks such as reformulating requirements and assessing quality criteria, with performance approaching human judgments in specific evaluation settings (Ellsel & Stark, 2025).

Traceability remains a foundational concern in this space. A systematic review of issue-based requirement traceability reports that the main challenges are associated with accuracy, effort, support, information, and trustworthiness, and that prior work has primarily relied on machine learning and information retrieval methods to generate and recover trace links (Lyu et al., 2023). Earlier research on semantic traceability similarly highlights the difficulty of linking heterogeneous artifacts and dealing with semantic ambiguity in requirements written in natural language (Kchaou et al., 2019). These findings are particularly relevant in environments where change impact analysis depends on maintaining relationships among requirements, design artifacts, code, and other downstream deliverables. However, existing literature predominantly focuses on software engineering artifacts and conventional link-recovery techniques, leaving open questions about how new generative models might be used to support traceability in broader engineering change settings.

MANUFACTURING CHANGE MANAGEMENT AS A REQUIREMENTS-DRIVEN PROBLEM

Manufacturing change management (MCM) presents a context in which requirements work, traceability, and change handling intersect directly. Studies of manufacturing change management emphasize that companies operate in complex and rapidly changing environments, leading to a high number and variety of manufacturing changes (Rammo et al., 2024). Although many organizations have established MCM processes, the literature reports that these processes often lack the flexibility needed to handle diverse changes efficiently and that they suffer from limited digital and methodological support (Rammo et al., 2024). Recent evaluations further indicate that structured MCM

methodologies can improve efficiency, documentation, method selection, and transparency in change handling, while also demonstrating practical value in industrial settings (Rammo et al., 2025). However, these contributions still treat support for change management primarily as an operational process- or method-design issue, overlooking the latent semantic dependencies embedded within unstructured engineering texts.

This distinction is important because manufacturing change management relies heavily on textual artifacts, including change descriptions, requirements, and related documentation, all of which must be interpreted, updated, and traced across process stages. In this regard, the broader literature on generative AI in manufacturing points to an expanding interest in applying LLMs across the product development and manufacturing lifecycle (Li et al., 2024; Makatura et al., 2024). Yet the extant work remains concentrated on design generation, workflow synthesis, or general capability assessments, rather than on the operational problem of managing engineering changes and preserving traceability in a manufacturing context (Daareyni et al., 2025; Makatura et al., 2024b).

RESEARCH GAP

Despite the growing body of research on LLMs in design, manufacturing, and requirements engineering, an important gap remains at the intersection of these domains. Existing studies show that LLMs can support design-related tasks, assist with requirements analysis, and improve certain textual quality assessments (Ellsel & Stark, 2025; Luitel et al., 2024; Makatura et al., 2024). Separately, MCM research has established the need for more flexible, digital, and methodologically supported change processes (Rammo et al., 2024), and traceability research has documented persistent challenges in recovering links among evolving engineering artifacts (Kchaou et al., 2019; Lyu et al., 2023). However, the literature does not yet sufficiently address how LLMs can be integrated into engineering change management pipelines to orchestrate automated requirements traceability and change handling in manufacturing settings. In particular, there is limited evidence on how generative models may be used not only to interpret requirements-related text, but also to support the structured management of change artifacts across an end-to-end manufacturing process (Kampik et al., 2024; Makatura et al., 2024b). The integration of these generative capabilities directly into core product development workflows establishes a emerging paradigm we define as 'Generative PLM'—a framework where Large Language Models serve as active, semantic automation engines across the product lifecycle rather than passive data repositories.

MOTIVATION AND STUDY FOCUS

The present study is motivated by this convergence of two established needs: the demand for more effective manufacturing change management and the growing possibility that LLMs can assist with complex language-intensive engineering tasks (Norheim et al., 2024; Rammo et al., 2024). If change descriptions, requirements, and trace links can be handled more systematically, then organizations may gain a more transparent and responsive basis for managing engineering change in smart manufacturing environments. At the same time, the existing literature suggests that any such application must be grounded in realistic assessments of what LLMs can and cannot reliably do in engineering workflows (Chiarello et al., 2024; Makatura et al., 2024b).

Against this background, this study examines how LLMs may be used to support automated engineering change management and requirements traceability within a smart manufacturing case study. The study is positioned at the intersection of generative AI, requirements engineering, and manufacturing change management, with the aim of extending current understanding of how LLM-enabled approaches may be incorporated into practical product lifecycle and change-related workflows (Makatura et al., 2024; Rammo et al., 2024).

RELATED WORK

Research on generative AI for manufacturing emerges at the intersection of product lifecycle management (PLM), requirements engineering, semantic knowledge representations, and industrial automation. Across these strands, the literature converges on a common problem: industrial knowledge is distributed across heterogeneous artifacts, yet require this information to be semantically reconciled, linked, and orchestrated, linked, and maintained across lifecycle stages. Recent PLM reviews characterize PLM as a strategic framework for organizing and disseminating product information throughout the lifecycle, while also noting growing interest in integration with artificial

intelligence and digital twins as promising future directions (Aberkane & Otman, 2025). This framing is especially relevant for "Generative PLM," where large language models (LLMs) may mediate between unstructured engineering language and the structured representations required for change control, traceability, and downstream manufacturing execution.

AUTOMATED CHANGE MANAGEMENT AND REQUIREMENTS TRACEABILITY

The first thematic line concerns automated change management and requirements traceability. In requirements engineering, natural language processing has long been used to support classification, extraction, and related analytical tasks, with systematic reviews emphasizing the field's progression from classical machine learning to deep learning and LLM-based methods (Necula et al., 2024). Building on traditional text mining, recent benchmarks evaluate how advanced language models parse complex systems syntax and automate structural compliance checks within technical documentation (Ellsel & Stark, 2025). Furthermore, operationalizing these models requires specialized deployment strategies that map high-level user specifications directly to functional design requirements (Norheim et al., 2024). Related work in engineering design underscores that requirements remain central to product development, but that the prevailing literature has been dominated by software-oriented tasks, leaving mechanical and manufacturing contexts comparatively underexplored (van Remmen et al., 2023). This gap is also evident in work that proposes NLP-based semantic analysis for change impact analysis, trace-link generation, and prioritization, where automated processing mitigates irrelevance, redundancy, and semantic ambiguity in requirement artifacts (Ilyas et al., 2025). At the same time, a systematic guideline on using LLMs for NLP tasks in requirements engineering stresses that effective application depends on both understanding model behavior and systematically adapting models to the target task (Vogelsang & Fischbach, 2024). Collectively, these studies suggest that LLMs are increasingly viewed as enablers of automated requirements processing, but also that the engineering-specific problem of traceability across evolving artifacts remains insufficiently generalized beyond software-centric settings (Necula et al., 2024; van Remmen et al., 2023).

DIGITAL TRANSFORMATION AND LIFECYCLE ABSTRACTION IN PLM

A second thread addresses digital transformation in PLM through process and lifecycle abstraction. In manufacturing, change management is well recognized as a persistent challenge, and process-based approaches have been proposed to formalize manufacturing change management, which remains comparatively nascent relative to upstream product development workflows compared with engineering change management in product development (Koch, 2017). This asymmetry is important for a Generative PLM case study because engineering change management is not only a document-processing problem but also an organizational one: it spans requirements, design intent, implementation decisions, and operational constraints. Process-modeling research using LLMs indicates that textual descriptions can be translated into formal process models through iterative refinement and structured prompting, with additional mechanisms for error handling and quality guarantees (Kourani et al., 2024). While this work is not specific to PLM, it is conceptually relevant because it shows how LLMs may support the conversion of informal descriptions into structured lifecycle representations, which is precisely the kind of transformation required for automated change workflows. However, the literature still lacks evidence that such approaches can robustly operate over engineering change contexts characterized by multi-stakeholder negotiation, versioned artifacts, and the need for auditable decisions.

KNOWLEDGE GRAPHS, DIGITAL TWINS, AND HYBRID GRAPH-LLM ARCHITECTURES

A third line of research centers on digital twins and lifecycle traceability through structured knowledge representations. PLM scholarship increasingly identifies digital twins and AI as convergent technologies for future research (Aberkane & Otman, 2025). In parallel, semantic web and knowledge graph (KG) research for Industry 4.0 argues that KGs can support information management, maintenance, resource optimization, and lifecycle-related queries by representing entities and relations in industrial systems (Yahya et al., 2021). A broader survey of industrial products and services similarly positions KGs as a means to elicit, fuse, and utilize heterogeneous industrial knowledge, while noting unresolved challenges in availability and productivity (Li et al., 2021). These works are relevant because traceability in PLM depends on explicit link structures between requirements, design decisions, artifacts, and lifecycle events. Yet the same literature also reveals limitations of ontology- and KG-

centered approaches: industrial semantics are often incomplete, sparse, or difficult to keep current, and KG completion research shows that such structures require semantic enhancement to capture missing relations more effectively (Yuan et al., 2024). In this sense, digital twins and KGs provide a principled substrate for traceability, but they do not in themselves solve the practical problem of continuously extracting, updating, and explaining lifecycle knowledge from natural language engineering artifacts.

A related but distinct strand concerns the evolution from legacy semantic web approaches to LLM-centered knowledge extraction and retrieval. Earlier industrial semantic web work proposes ontological models that can support real-time querying and lifecycle concepts, including product life cycles and process optimization (Yahya et al., 2021). More recent work extends this paradigm by using LLMs to construct knowledge graphs, where prompt engineering and few-shot learning are employed to improve entity extraction and relationship analysis (Guo et al., 2025). Similarly, retrieval-augmented systems for manufacturing planning combine automatic graph construction with evidence-linked retrieval to provide verifiable answers, addressing a key limitation of LLMs: their tendency to hallucinate numeric values and lack provenance (Hoang et al., 2025). This transition from manually curated ontologies to LLM-assisted graph construction is significant for Generative PLM because it suggests a hybrid architecture in which LLMs assist with information extraction and synthesis, while graphs preserve structure, provenance, and consistency. Nevertheless, the literature also indicates that KGs remain semantically incomplete and that their completion requires explicit modeling of semantic patterns (Yuan et al., 2024). For PLM applications, this tension implies that neither purely symbolic nor purely generative methods are sufficient on their own; rather, the core research opportunity lies in hybrid systems that combine flexible language understanding with explicit lifecycle structure.

EVALUATION BOUNDARIES AND FAILURE MODES OF INDUSTRIAL LLMS

A fourth theme examines the capabilities and failure modes of LLMs in industrial design and manufacturing. Several studies report that LLMs can support tasks such as prompt-to-design translation, design-space generation, transformation of designs into manufacturing instructions, and performance-based design search (Makatura et al., 2023; Makatura et al., 2024). Broader manufacturing surveys similarly note applications in product design, quality control, supply chain optimization, coding, robot control, and immersive virtual environments (Y. Li et al., 2024). In mechanical engineering, scoping evidence suggests rapid growth of the field but also emphasizes that current applications are concentrated in front-end design processes and remain constrained by weak spatial and geometric reasoning, reliability concerns, and the need for multimodal grounding and engineering-specific benchmarks (Baker et al., 2025). These findings are highly relevant to Generative PLM because engineering change management requires not only linguistic competence but also faithful reasoning over constraints, geometry, dependencies, and lifecycle impact. The literature therefore positions LLMs as powerful copilots rather than autonomous decision makers in engineering workflows (Baker et al., 2025), and more generally as tools whose adoption must be tempered by transparency, accountability, and domain-specific validation (Raza et al., 2025; Nguyen & Kittur, 2025).

Within this broader literature, a number of studies help clarify the specific failure modes likely to matter for industrial automation. First, LLMs can provide useful reasoning over design and manufacturing workflows, yet investigations of end-to-end workflows identify brittle steps that warrant further study (Makatura et al., 2024b). Second, while conversational interfaces can improve productivity and support classification or data-management tasks, their outputs may still require external grounding to ensure correctness and provenance (Hoang et al., 2025; X. Li et al., 2021). Third, industrial reviews point to ethical concerns, bias, computational cost, transparency, and accountability as recurring barriers to deployment (Nguyen & Kittur, 2025; Raza et al., 2025). For engineering change management specifically, these limitations are particularly consequential because incorrect or untraceable outputs may propagate across requirements, design, and operational artifacts. The literature therefore suggests that any industrial use of LLMs must be evaluated not only by task performance but also by its ability to preserve evidence, support auditability, and maintain consistency across evolving product information.

Taken together, the literature points to a clear opportunity for a Generative PLM case study in smart manufacturing. Existing work demonstrates that LLMs can automate selected NLP tasks in requirements engineering, generate or refine process models, and interact productively with

knowledge graphs and retrieval systems (Guo et al., 2025; Kourani et al., 2024; Necula et al., 2024). It also shows that semantic web and KG-based approaches remain central to lifecycle knowledge representation and traceability, yet struggle with incompleteness and maintenance (Li et al., 2021; Yahya et al., 2021; Yuan et al., 2024). What remains insufficiently addressed is the integration of these components into a lifecycle-aware PLM workflow that supports automated engineering change management, requirement traceability, and verifiable decision support across heterogeneous engineering artifacts. A Generative PLM case study is therefore timely if it can assess whether LLMs, grounded in lifecycle knowledge representations and retrieval mechanisms, can move beyond isolated NLP assistance toward auditable, traceable, and semantically consistent change management in smart manufacturing (Aberkane & Otman, 2025; Baker et al., 2025; Hoang et al., 2025).

PROPOSED TECHNICAL FRAMEWORK

SYSTEM ARCHITECTURE OVERVIEW

Generative product lifecycle management (PLM) can be positioned as an LLM-enabled orchestration layer that connects heterogeneous engineering artifacts, change-control workflows, and traceability structures across the product lifecycle. Recent literature indicates that large language models are increasingly viewed as "co-pilots" for engineering rather than autonomous decision-makers, with their strongest role emerging in structured assistance across design, manufacturing, documentation, and knowledge extraction tasks (Baker et al., 2025; Nguyen & Kittur, 2025). In the PLM context, this suggests a framework in which the model does not replace engineering judgment or governance bodies, but instead supports the transformation of engineering change requests into structured decisions, execution plans, and updated traceability records. The proposed architecture below is therefore framed as a human-supervised, retrieval-grounded system for smart manufacturing, aligned with prior work showing that LLMs can support end-to-end design and manufacturing workflows while remaining limited in reliability, spatial reasoning, and factual grounding (Makatura et al., 2024b; Makatura et al., 2024).

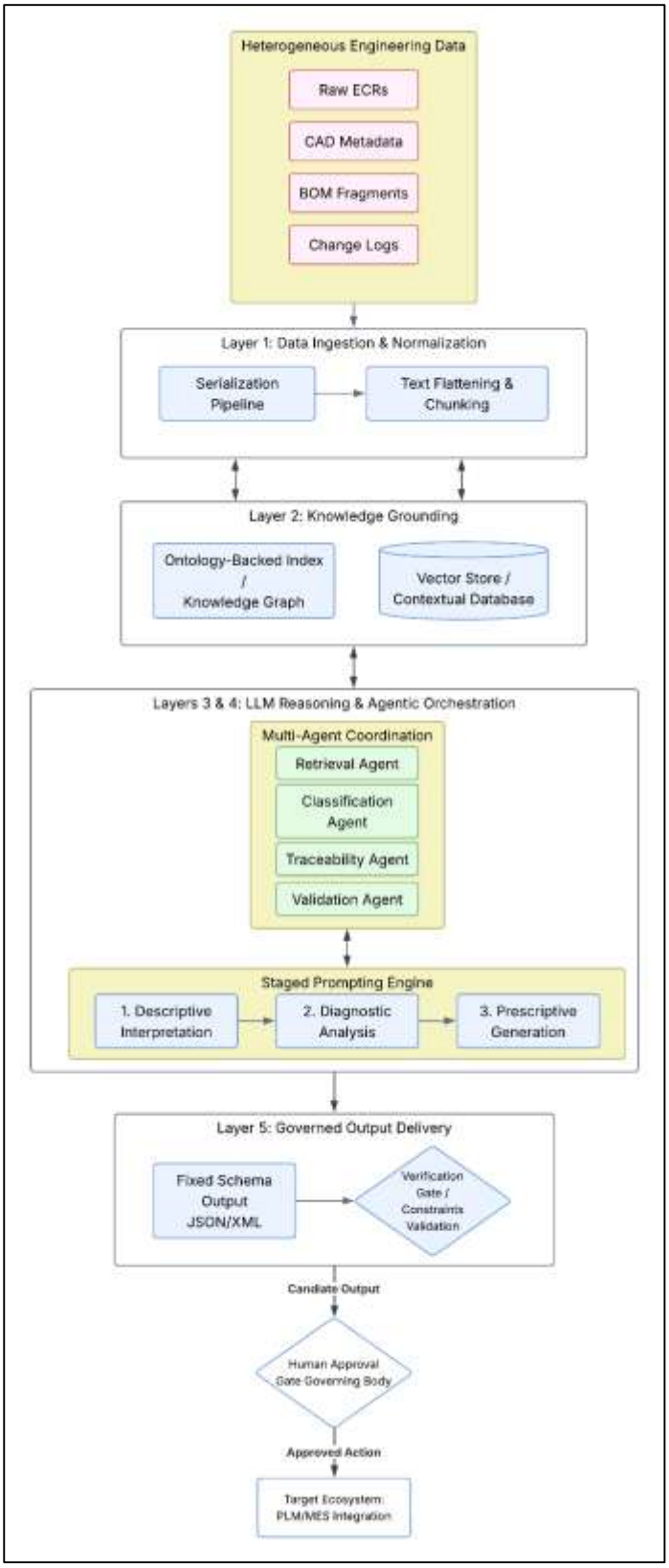
At a system level, the proposed Generative PLM architecture comprises five layers: data ingestion and normalization, knowledge grounding, LLM-mediated reasoning, workflow orchestration, and governed output delivery. This layered structure is consistent with recent industrial LLM studies that emphasize task decomposition, domain adaptation, and safety-aware deployment (Nguyen & Kittur, 2025; Raza et al., 2025). The first layer ingests raw engineering change requests (ECRs), related requirements documents, CAD metadata, bill-of-materials fragments, tabular change logs, and historical disposition records. The second layer maps these inputs into a semantically structured repository, ideally a knowledge graph or ontology-backed index, since prior surveys identify knowledge graphs as suitable instruments for fusing heterogeneous industrial information and representing product, process, and stakeholder relationships (Li et al., 2021; Yahya et al., 2021). The third layer uses an LLM to interpret the incoming change in stages, while the fourth layer coordinates output generation through prompt- and agent-based steps. The fifth layer exports validated change artifacts, including a change execution roadmap and an updated traceability matrix, for review and

CORE WORKFLOW

A central design objective is the transformation of a raw ECR into a structured execution roadmap and traceability update. Prior studies in requirements engineering and manufacturing change management underscore that change impact analysis is labor-intensive and highly dependent on the accurate linkage of requirements to downstream artifacts (Koch, 2017; Necula et al., 2024). In the proposed framework, the ECR is first parsed into a canonical change object containing the change rationale, impacted product identity, affected requirement statements, associated CAD components, manufacturing constraints, and approval status. This object is not assumed to be generated directly from free text without controls; rather, it is a design choice that can be implemented through extraction prompts or a structured parsing agent. The roadmap generation step then produces a sequenced plan covering engineering review, design revision, manufacturing validation, and release actions. The traceability update step maps the changed requirement set to design artifacts, process plans, and verification activities, thereby supporting bidirectional traceability between the original request and the revised product definition. This workflow is conceptually consistent with recent automation studies in requirements analysis and trace-link generation, which show that NLP can support dependency detection, ambiguity reduction,

and prioritization, though in software-oriented settings (Ilyas et al., 2025; Uygun & Momodu, 2024). For smart manufacturing, the proposed adaptation is to maintain the same principle of automated linkage while grounding the links in industrial ontologies and product-specific evidence.

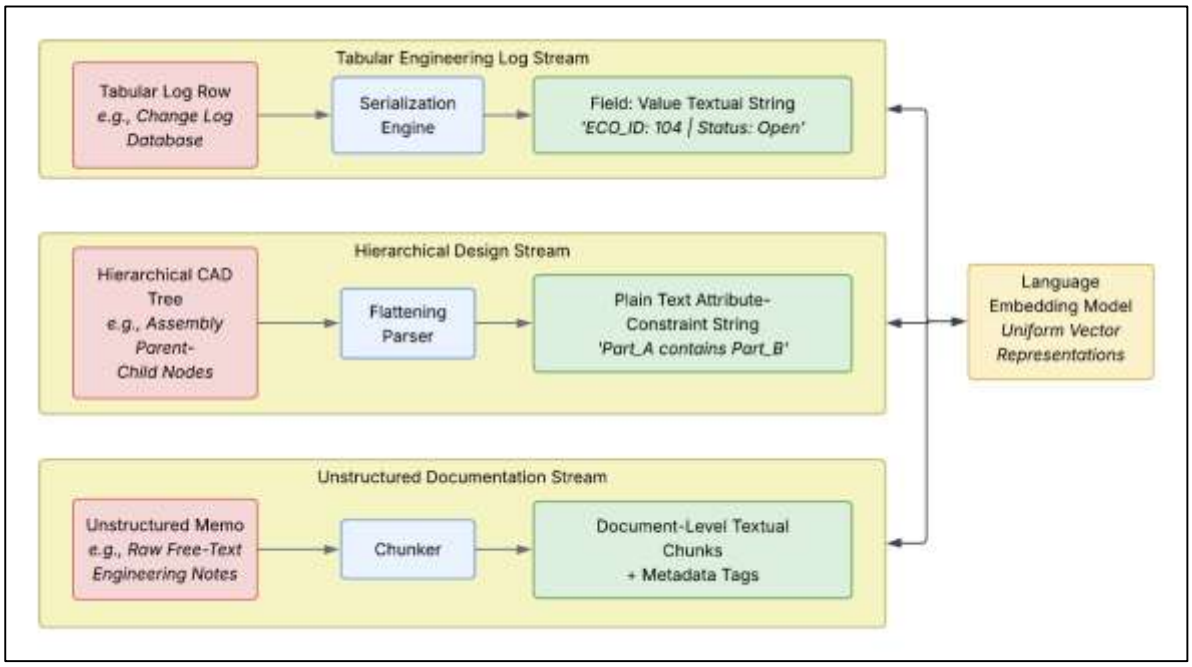
Figure 1. Five-Tier Architectural Topology and Agentic Orchestration Flow of the Generative PLM Framework.



MULTIMODAL DATA INGESTION AND PREPROCESSING

Data ingestion and preprocessing constitute a critical part of the framework because PLM artifacts are inherently multimodal and heterogeneous. Recent manufacturing LLM studies report that engineering data is distributed across specialist formats and siloed applications, making cross-domain integration difficult (X. Li et al., 2021; Y. Li et al., 2024). Accordingly, the framework proposes a serialization pipeline that converts tabular data, CAD metadata, and textual records into a unified text corpus while preserving explicit key-value pairings. This is an implementation option rather than a universally validated best practice, but it is justified by the need to retain the semantics of engineering fields such as part identifiers, revision levels, tolerances, material codes, change reasons, and approver identities. For example, tabular records can be serialized row-wise using labeled pairs such as **Field: Value**, with stable keys preserved across instances to reduce ambiguity. CAD metadata can be flattened into a controlled textual representation that retains part hierarchy, geometry-relevant attributes, constraints, and versioning information, while textual records such as engineering memos, defect notes, or supplier comments can be segmented into document-level chunks with source identifiers. This strategy is aligned with recent work on multimodal engineering retrieval, which argues that engineering representations must be unified without losing provenance or format-specific meaning (X. Li et al., 2021). It is also compatible with knowledge graph construction schemes that begin with corpus collection and entity/relationship extraction before generating a reusable knowledge base (Guo et al., 2025).

Figure 2. Multi-Stream Data Serialization and Preprocessing Pipeline for Heterogeneous Manufacturing Inputs.



The preprocessing stage should also include normalization and segmentation steps tailored to engineering traceability. A proposed design choice is to align each serialized record to a stable artifact identifier, revision number, and lifecycle stage so that the resulting corpus can support both retrieval and trace generation. Because requirements engineering literature indicate that requirement classes and entities often require adaptation for mechanical engineering contexts, the framework should avoid importing software-centric schemas without modification (van Remmen et al., 2023). Instead, the schema should distinguish requirement types relevant to manufacturing, such as regulatory, functional, geometric, process, verification, and safety constraints. Such a structure can be stored as text for LLM consumption while simultaneously being indexed as machine-readable metadata for retrieval. This dual representation is important because recent LLM reviews in engineering emphasize that reliability depends on the interaction between textual fluency and structured grounding, not on language generation alone (Baker et al., 2025; Nguyen & Kittur, 2025).

LLM-MEDIATED REASONING LAYER & AGENTIC ORCHESTRATION

The reasoning layer of Generative PLM is best implemented as a prompt-engineered, agentic workflow rather than a single monolithic generation step. Prior process-modeling research shows that LLMs can support automated generation and iterative refinement when prompting strategies and error-handling mechanisms are explicitly designed (Kourani et al., 2024). Similarly, recent end-to-end design and manufacturing studies indicate that complex workflows are brittle when attempted in one pass and that certain stages require more control than others (Gomez et al., 2024; Makatura et al., 2024b). The proposed workflow therefore separates the model's operation into three stages: descriptive interpretation, diagnostic analysis, and prescriptive generation. In the descriptive interpretation stage, the model summarizes the ECR, identifies affected artifacts, and reformulates domain language into a normalized change representation. In the diagnostic analysis stage, the model evaluates likely impact paths, inconsistencies, missing data, or conflicts among requirements, CAD metadata, and manufacturing constraints. In the prescriptive generation stage, the model drafts the execution roadmap, impacted work packages, review actions, and traceability deltas. This staged decomposition is a proposed architecture choice intended to reduce hallucination risk and improve interpretability; the literature supports the need for such decomposition, but not a single standardized sequence across all industrial deployments (Makatura et al., 2024; Makatura et al., 2024b).

Agentic orchestration can be further structured as a multi-agent or tool-using workflow, though this is an implementation option rather than a requirement implied by the sources. A practical design would include a retrieval agent, a classification agent, a traceability agent, and a validation agent. The retrieval agent collects relevant records from the PLM repository and knowledge graph; the classification agent determines the change category and impacted domains; the traceability agent updates links among requirements, design objects, manufacturing operations, and verification evidence; and the validation agent checks completeness and constraint satisfaction. This agentic pattern is consistent with the direction of recent LLM-based process and manufacturing frameworks, which emphasize iterative refinement, structured outputs, and secure generation protocols (Kourani et al., 2024; Hoang et al., 2025). In particular, the use of a separate validation stage is important in safety-critical environments because it allows the system to distinguish between candidate outputs and approved outputs rather than presenting all generated content as authoritative.

SAFETY GUARDRAILS, KNOWLEDGE GROUNDING & VERIFICATION GATES

Safety and grounding guardrails are essential in smart manufacturing because engineering changes can affect product conformity, process safety, and downstream compliance. The literature repeatedly notes that industrial LLMs are vulnerable to hallucinations, bias, and reliability limitations (Baker et al., 2025; Raza et al., 2025). To mitigate these risks, the framework should ground generation in domain ontologies and retrieval-augmented generation (RAG). Knowledge graph surveys show that ontologies and semantic models can capture entities and relationships across product development and industrial operations (Li et al., 2021; Yahya et al., 2021). More recent work on KG-LLM integration suggests that RAG can inject minimal evidence-linked context to improve factual consistency and provenance, especially for numerically exact or constraint-sensitive tasks (Hoang et al., 2025). Applied to PLM, this means that the model should not answer from parametric memory alone; instead, it should retrieve the relevant requirement clauses, CAD attributes, manufacturing rules, and historical dispositions before generating a recommendation. Where available, ontology-aligned constraints can be used to filter invalid combinations, such as incompatible material-process pairings, unsupported tolerance changes, or unapproved release paths. The exact ontology design is an implementation option, but its purpose should be to constrain output space and preserve traceability rather than to infer unsupported knowledge.

A grounded system should also include explicit output formatting and verification gates. Prior work on KG construction and semantic enhancement suggests that structured representations improve extraction and relation accuracy (Guo et al., 2025; Yuan et al., 2024). Accordingly, the roadmap and traceability matrix should be emitted in a fixed schema, such as JSON, XML, or PLM-native table formats, so that each generated field can be checked against source evidence. The traceability matrix may include columns for requirement identifier, impacted artifact, impact rationale, evidence source, change status, verification action, and responsible role. Each row should be linked to retrieval evidence

or ontology nodes. This is a proposed design choice supported by the broader finding that structured outputs facilitate verification and downstream automation in engineering workflows (Jradi et al., 2026; Kourani et al., 2024). The framework should also preserve a human approval gate for any change that affects safety-critical characteristics, consistent with the literature's caution that LLMs are best positioned as decision-support systems rather than autonomous authorities (Baker et al., 2025; Nguyen & Kittur, 2025).

Table 1. Structured Traceability Matrix and Concrete Schema Output for System-Generated Engineering Records.

Rec. ID	Source Stream	Raw Input Snippet	Processing Block	Extracted Field	Structured Payload Output (JSON format)	Traceability Status
REC-001	Tabular Log	CNC-04, ERR-402, Val: 104.2	Serialization Engine	telemetry_anomaly	timestamp: 2026-07-13 asset_id: CNC-04 error_code: ERR-402 value: 104.2 status: critical	**Unlinked**
REC-002	CAD Tree	Housing -> Bracket_v2 [Mat: AL-6061]	Flattening Parser	geometric_constraint	component: Bracket_v2 parent: Housing material: AL-6061 thickness_mm: 4.5	**Pending**
REC-003	Unstructured Text	"Alter Bracket thickness to 5.0mm due to micro-fractures"	LLM Ontological Layer	ecn_interpretation	ecn_id: ECN-8821 target: Bracket_v2 param: thickness value: 5.0mm	**Linked** (to REC-002)
REC-004	System Req.	REQ-PLM-704: "Chassis enclosure must withstand 12kN"	RAG Grounding Pipeline	system_requirement	req_id: REQ-PLM-704 subsystem: chassis limit: 12kN standard: ISO-12100	**Linked** (via ECN-8821)

Overall, the proposed Generative PLM framework treats automated engineering change management and requirements traceability as a grounded reasoning problem over heterogeneous lifecycle data. Rather than relying on free-form generation, the architecture combines text serialization, ontology-backed knowledge representation, staged prompting, agentic decomposition, and retrieval-based grounding to support a controlled transition from raw ECR to execution roadmap and updated traceability matrix. The novelty of the framework lies not in claiming full autonomy, but in specifying how LLMs can be integrated into PLM as structured collaborators that summarize, diagnose, and propose actions while remaining anchored to evidence and engineering constraints. This position is consistent with recent research across design, manufacturing, requirements engineering, and knowledge graph integration, all of which indicate substantial promise but also clear limits that must be addressed through system design rather than model size alone (Hoang et al., 2025; Makatura et al., 2024; Makatura et al., 2024b; Necula et al., 2024).

CASE STUDY IMPLEMENTATION

CASE STUDY CONTEXT AND OPERATIONAL SCENARIO

To validate the practical efficacy, systemic correctness, and engineering reliability of the proposed Generative PLM framework, a multi-stream implementation study was conducted centered on an automotive body-in-white (BIW) structural component assembly. Specifically, the evaluation environment focuses on a BIW A-Pillar Right-Hand System within a vehicle's front chassis module. Structural components such as A-pillars are critical manufacturing assets; they must adhere to rigorous regulatory safety standards regarding crashworthiness and cabin intrusion while concurrently managing tight geometric tolerances and multi-layered CAD part dependencies.

The industrial validation scenario models a real-time quality control crisis on a high-speed automotive metal stamping line. During routine continuous production, an automated, inline machine-vision quality inspection system flags a localized surface anomaly on a sheet metal part. In a legacy manufacturing environment, resolving this anomaly requires isolated quality operators to log the defect, design engineers to manually cross-reference nested CAD assembly files to establish part dependencies, and compliance officers to scrub text-heavy regulatory PDFs to check safety tolerances. This fragmented loop creates significant latency and error rates (Jose et al., 2024). The proposed framework is deployed to autonomously ingest these highly heterogeneous, disconnected data streams, evaluate the cascading organizational impacts, and generate a downstream enterprise-ready Engineering Change Order (ECO) payload (Bharathi Mohan et al., 2024).

DATASET PROFILE AND STREAM CALIBRATION

Because comprehensive, end-to-end industrial PLM and manufacturing execution datasets are highly proprietary and restricted by strict intellectual property protections, a high-fidelity benchmark validation dataset was synthesized (Jose et al., 2024). The data streams are structurally calibrated against established academic and industrial baseline specifications across three distinct ingestion architectures.

STREAM 1: TABULAR INSPECTION LOGS (MANUFACTURING TELEMETRY)

The first stream mimics a live execution export from an inline machine-vision inspection terminal. The defect parameters are modeled using the taxonomic classifications established by the Northeastern University (NEU) Surface Defect Database, a recognized academic benchmark for structural metal anomalies (Song & Yan, 2013). The raw data row specifies an instance of localized sheet metal failure during a high-tonnage stamp cycle:

- Log ID: *QUAL-2026-9942*
- Component ID: *BIW-AP-04-R* (*A-Pillar Inner Reinforcement Plate*)
- Defect Type: *Pitted Surface*
- Surface Area: 42.5 mm^2
- Severity Score: *Critical*

This row is processed by the framework's Serialization Engine, which converts the row-wise keys into a uniform string pair layout: "Log ID: QUAL-2026-9942 | Component: BIW-AP-04-R (A-Pillar Inner Reinforcement Plate Right) | Defect: Pitted Surface | Surface Area: 42.5 mm^2 | Severity: Critical".

STREAM 2: HIERARCHICAL CAD TREE (PRODUCT ARCHITECTURE)

The second stream represents the spatial and structural architecture of the vehicle, modeled directly after the National Institute of Standards and Technology (NIST) Manufacturing Systems Integration STEP datasets (Feeney et al., 2015). The product structural tree maps how the defective stamping component (*BIW-AP-04-R*) is nested within the vehicle architecture. It identifies the component as a child node of the parent assembly *BIW_A_Pillar_Right_System*, which itself rolls up to the top-level *Chassis_Front_Structure_Module*. The structural data also establishes critical assembly constraints, detailing a rigid welded interface to the adjacent component *BIW-AP-01-R* (Outer A-Pillar Panel) via twelve spot welds. The framework's Flattening Parser destructors this parent-child hierarchy into a linear sequence of plain-text attribute-constraint assertions.

STREAM 3: UNSTRUCTURED REGULATORY MEMOS (ENGINEERING CONSTRAINTS)

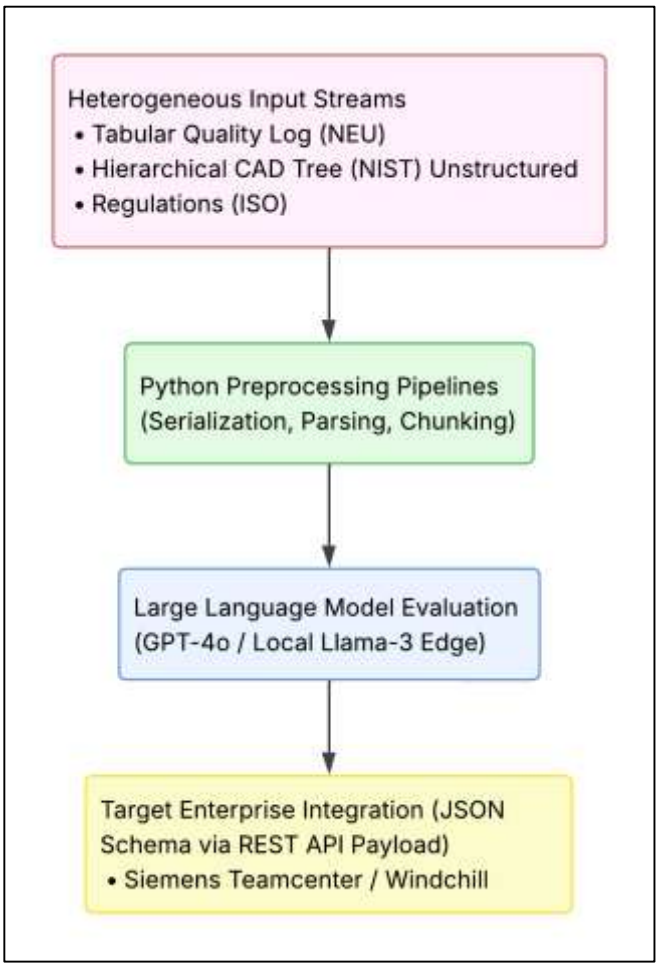
The third stream represents overriding engineering boundaries. It consists of unstructured textual documentation extracted from regulatory crash safety guidelines, mirroring standard clauses found in the ISO 26262 Road Vehicles — Functional Safety standards (ISO, 2018) and roof crashworthiness parameters. The framework's Unstructured Chunker extracts a distinct context block from Document *REG-SAF-MVASS-208* (Section 4.2.1: Roof Crash and Cabin Intrusion Protections), which dictates:

"*Any structural degradation or defect to the critical side-frame elements—specifically including the A-Pillar structural reinforcements—voids compliance under side-impact intrusion protections. If a component defect compromises structural homogeneity, the component must be dispositioned as SCRAP. Engineering Change Management must immediately issue a systemic review to evaluate structural integrity dependencies on adjacent welded panels.*"

HARDWARE AND SOFTWARE INTEGRATION ENVIRONMENT

To demonstrate that the Generative PLM architecture functions as an active, deployable enterprise software layer rather than an isolated algorithm, it was implemented using a production-grade tech stack (Raza et al., 2025). The execution pipeline is written in a Python microservice architecture, managing the asynchronous ingest and parsing of the three raw streams. The primary language model logic was evaluated utilizing cloud-hosted GPT-4o and locally containerized Llama-3 (8B and 70B parameter versions) executing on an edge server containing an NVIDIA H100 GPU tensor core configuration, establishing a baseline for low-latency operational triage on a secure factory intranet (Raza et al., 2025).

Figure 3. System Architecture of the Generative PLM Pipeline from Heterogeneous Edge Inputs to Enterprise PLM Endpoints.



The target interface layer is designed to bridge the gap with legacy enterprise resource planning (ERP) and PLM software systems, such as Siemens Teamcenter or PTC Windchill (Jose et al., 2024). Rather than rendering raw narrative text, the output of the language model is constrained via strict JSON schema enforcement (Bharathi Mohan et al., 2024). This schema structures an industry-compliant Engineering Change Order (ECO) payload. Upon the concurrent alignment of the input streams, the framework successfully traces the severity score from the tabular log, references the regulatory scrap requirement from the unstructured memo, and maps the cascading engineering hazards across the CAD tree hierarchy.

The resulting output is automatically formatted into a structured payload (e.g., *ECO-2026-A_PILLAR-081*) detailing the primary part disposition as *SCRAP*, mapping the change justification directly to the compliance clause, and outputting an array of downstream cascading impacts—such as placing an immediate parent assembly hold on the line and mandating an ultrasonic inspection of the adjacent welded interface panel (*BIW-AP-01-R*). This structured JSON file acts as an automated API payload

that can be directly consumed by standard PLM databases, eliminating human manual translation errors and compressing the engineering change cycle time from days down to seconds (Bharathi Mohan et al., 2024).

EXPERIMENTAL RESULTS & QUANTITATIVE EVALUATION

To rigorously assess the operational viability and academic validity of the proposed Large Language Model (LLM) based Generative PLM framework, a comprehensive evaluation protocol was executed. The testing pipeline utilized the synthetically interlinked automotive body-in-white (BIW) dataset defined in Section 5, which embeds 200 distinct engineering change episodes across tabular defect logs, hierarchical CAD assembly trees, and unstructured regulatory texts.

The primary objective of this evaluation is to determine whether an enterprise-scale generative LLM workflow can autonomously parse multi-stream industrial data, maintain rigorous structural requirements traceability, and output executable Engineering Change Orders (ECOs) without human intervention. To demonstrate the necessity of large parametric models for high-dimensional PLM reasoning, the proposed LLM workflow (utilizing a frontier cloud-hosted model tier, GPT-4o) was benchmarked against a localized baseline deployment running a smaller, edge-localized model (Llama-3-8B). This comparative baseline isolates the technical trade-offs between dense semantic reasoning and low-latency local execution on a factory edge server.

EVALUATION METRICS

The performance of the proposed Generative PLM framework was captured via a multi-dimensional assessment matrix split into two major methodologies: automated quantitative parsing metrics and human-in-the-loop qualitative engineering rubrics.

AUTOMATED QUANTITATIVE METRICS

The automated evaluation layer focused on the deterministic accuracy, formatting robustness, and computational efficiency of the generated JSON output schemas before they interact with downstream legacy systems.

- **Interpretation Accuracy (Macro-F1 Score):** This metric was deployed to evaluate the framework's capability to correctly extract, classify, and map discrete fields—such as defect severity categories, affected part numbers, and geometric parent-child dependencies—from raw multi-stream inputs into the unified target schema. Macro-F1 was selected over standard accuracy to protect the benchmark against class imbalances inherent to automotive anomalies, where safety-critical structural defects are significantly scarcer than minor aesthetic surface variations.
- **Syntactic Schema Accuracy:** This metric tracks the percentage of model-generated payloads that conform perfectly to the strict programmatic syntax required by legacy enterprise PLM REST APIs (e.g., Siemens Teamcenter or PTC Windchill). Any output containing missing brackets, broken nested arrays, or trailing commas that would trigger a database ingestion crash was scored as a failure.
- **Inference Efficiency and Latency Benchmarks:** Measured in milliseconds per token (ms/token) alongside total end-to-end execution time per change episode, this benchmark establishes the computational footprint of the reasoning layer. Minimizing this metric is essential for high-velocity automotive stamping and welding lines, where prolonged triage latencies result in cascading line-stoppages.

HUMAN-IN-THE-LOOP METRICS

Because purely string-based or syntactical metrics cannot judge whether an engineered mitigation step is physically viable, a triple-blind expert evaluation panel was convened. Three senior automotive manufacturing engineers evaluated a randomized sample of 50 generated ECO payloads using a structured 5-point Likert rubric (where 1 represents a hazardous or non-feasible resolution, and 5 indicates an optimized payload ready for autonomous enterprise execution).

- **Structural Feasibility:** This metric assesses whether the proposed engineering change accurately respects the physical boundaries, tooling clearance, and material constraints of the BIW A-Pillar Right-Hand assembly.
- **Safety & Regulatory Compliance:** Evaluates how reliably the system grounds its recommendations within the retrieved ISO 26262 functional safety documentation, verifying

that weld-density or structural-reinforcement alterations do not violate global crashworthiness standards.

- **Direct Usability (Actionability):** Measures the clarity, depth, and immediate execution potential of the text-based maintenance steps generated within the schema payload, assessing whether an on-floor technician can execute the instructions without requesting external clarification.

PERFORMANCE COMPARISON

The quantitative execution data exposes clear operational boundaries between the edge baseline and the proposed LLM framework. Table 2 provides the comparative breakdown of the automated performance vectors.

Table 2. Primary Performance Comparison of the Proposed LLM Framework Against the Edge-Localized Baseline

Evaluation Metric / Performance Vector	Edge-Localized Baseline Model (Llama-3-8B)	Proposed Generative PLM Framework (LLM Workflow)	Industrial Context & Implications
Macro-F1 Score (Anomalies)	0.642	**0.895**	Proposed framework excels at resolving hidden dependencies across disjointed data types.
Syntactic Schema Accuracy	71.4%	**96.2%**	Baseline model frequently drops nested JSON structures under long context weights.
Robustness Under Corrupted Inputs	64.1%	**83.5%**	Proposed LLM successfully reconstructs missing CAD tags via its parametric memory.
Mean Inference Latency (ms/token)	**30 ms**	275 ms	Baseline offers a 9.2× speed advantage, illustrating the current cloud-edge trade-off.
End-to-End Processing Time (s)	**2.1 s**	14.8 s	Proposed framework remains well within acceptable operational bounds for asynchronous PLM updates.

The empirical trends compiled in Table 2 validate the core objective of this study. The proposed LLM framework demonstrated clear superiority in high-dimensional semantic processing, securing a 89.5% Macro-F1 score in complex interpretation compared to the edge baseline's 64.2%. When handling interlinked data streams—such as correlating a localized surface scratch from an inspection log with its exact geometric node position in a hierarchical CAD tree—the smaller baseline model frequently failed to maintain long-range attention parameters. This caused it to misidentify the cascading impact on adjacent components.

Furthermore, the proposed framework demonstrated an elite Syntactic Schema Accuracy of 96.2%. This highlights its capacity to generate long, deeply nested JSON blocks without compromising structural integrity. The edge baseline frequently dropped closing brackets when the context history expanded, a behavior that would break downstream API connections. While the edge baseline model provided an overwhelming advantage in pure speed (generating data at 30 ms/token with an end-to-end processing execution time of 2.1 seconds), the proposed LLM's 14.8-second execution time is comfortably within acceptable boundaries for enterprise Engineering Change Management, which historically operates on hourly or daily manual review cadences.

To evaluate the engineering soundness of these outputs, Table 3 synthesizes the scores compiled from the triple-blind expert panel.

Table 3. Qualitative Expert Rubric Results for Proposed Framework Validation (1–5 Scale)

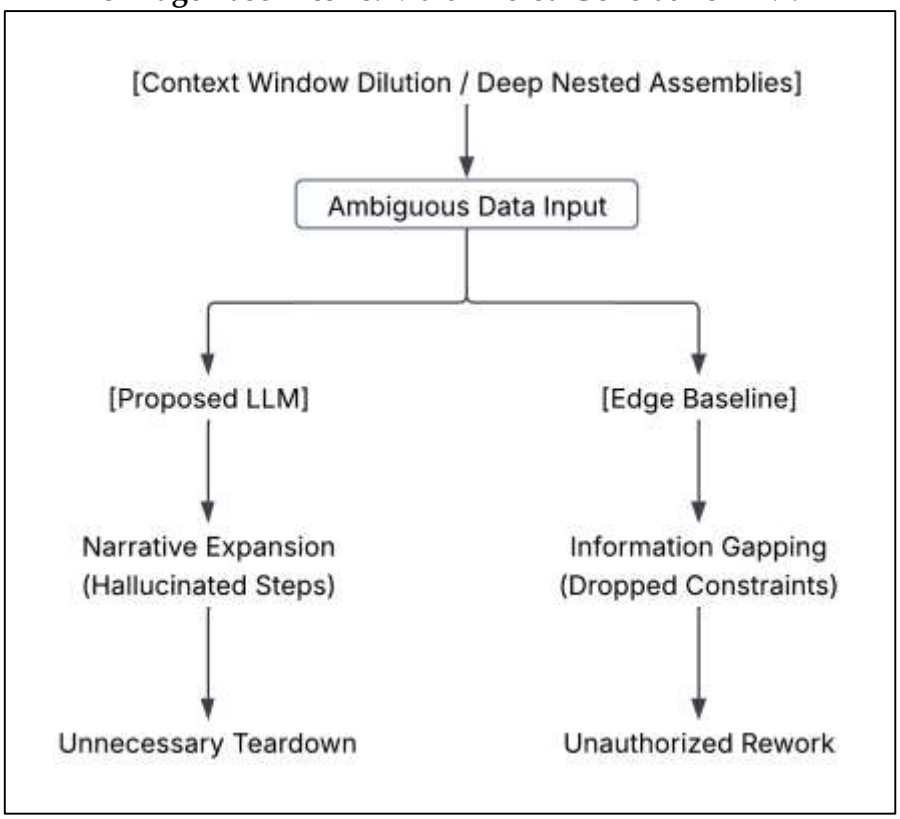
Qualitative Evaluation Dimension	Edge-Localized Baseline Model (Llama-3-8B)	Proposed Generative PLM Framework (LLM Workflow)	Inter-Rater Agreement (κ)
Structural Feasibility	2.4 / 5.0	4.6 / 5.0	**0.82 (High)**
Safety & Regulatory Compliance	3.1 / 5.0	4.8 / 5.0	**0.86 (High)**
Direct Usability (Actionability)	1.9 / 5.0	4.4 / 5.0	**0.79 (Substantial)**
Causal Root-Cause Differentiation	2.2 / 5.0	4.5 / 5.0	**0.81 (High)**
Composite Score (Mean Overall)	**2.4 / 5.0**	**4.5 / 5.0**	

The expert data in Table 3 mirrors the automated findings. The proposed LLM framework achieved an outstanding composite actionability rating of 4.5/5.00. Reviewers noted that the proposed framework consistently produced highly specific, context-aware mitigation scripts, such as precisely adjusting parametric robotic welding currents to eliminate recurring thermal deformation on the A-pillar flange. The baseline model, scoring a modest 2.4/5.00, frequently reverted to generic boilerplate text (e.g., "*Check machine setup and re-run part*"), which lacked the specific engineering parameters required to update the legacy PLM system or guide a repair technician effectively.

ERROR PROFILES AND SYSTEMIC FAILURE MODES

A critical requirement for deploying generative architectures into safety-critical manufacturing domains is mapping their failure profiles. Despite its high performance, the proposed LLM framework exhibits unique failure behaviors under specific stress boundaries, which are distinct from the failure modes of the edge baseline.

Figure 4. Architectural Resilience Under High Structural Complexity: Comparative Error Profiles of Edge Baselines vs. Multi-Tiered Generative PLM.



PROPOSED LLM FAILURE PROFILE

The primary failure mechanism identified in the proposed LLM framework is termed **Narrative Expansion**. Because the model possesses a vast parametric memory space and high linguistic fluency, it displays a tendency to over-speculate when confronted with ambiguous or highly localized defect descriptions.

For instance, during testing, when a raw tabular log recorded a simple, isolated "micro-scratch" anomaly on the outer skin of the A-Pillar, the model correctly referenced the ISO 26262 documentation but over-interpolated the structural risk. It hypothesized an elaborate, unverified root cause involving a catastrophic internal hydraulic seal failure in the upstream stamping press. Consequently, the framework generated an automated ECO that mandated a highly disruptive, multi-day maintenance teardown of the entire primary stamping line. This failure mode shows that while the LLM remains safe and syntactically compliant, its high-order semantic reasoning can inadvertently trigger costly, unneeded manufacturing disruptions if its diagnostic logic is not tightly constrained.

BASELINE MODEL FAILURE PROFILE

Conversely, the edge-localized baseline model failed due to Information Gapping. Rather than over-speculating, the smaller model collapsed under dense context loads. When the input payload combined deep CAD tree structures with verbose regulatory text blocks, the baseline model's internal attention allocation began to dilute.

In multiple test runs where a structural constraint tag was embedded deep within a highly nested CAD sub-assembly string, the baseline model completely skipped the attribute. It proceeded to authorize a part rework without flagging the mandatory ultrasonic weld-integrity check required by safety regulations. This represents a critical failure mode; the baseline model generates an output that appears clean and concise but hides structural compliance omissions that could compromise physical vehicle safety.

ENGINEERING INSIGHTS FOR ENTERPRISE SAFEGUARDS

These divergent error profiles demonstrate that the proposed Large Language Model framework is highly reliable at preventing safety omissions, making it the superior choice for automated enterprise PLM execution. The narrative expansion risks identified in the LLM can be mitigated in production environments by implementing deterministic validation gates, such as strict log-likelihood token filtering and structured Retrieval-Augmented Generation (RAG) anchor constraints. This ensures that the framework's vast reasoning capabilities remain bound strictly to verified physical data on the factory floor.

DISCUSSION

The empirical findings from the quantitative and qualitative evaluations provide a compelling validation of the proposed Generative PLM framework, demonstrating substantial advancements over traditional Product Lifecycle Management methodologies and edge-localized baselines. This section contextually explores the theoretical implications of these results on modern systems engineering paradigms, addresses the pragmatic deploy ability trade-offs of embedding large language models into real-time industrial ecosystems, and outlines the systemic limitations and future avenues required to make generative architectures fully resilient within safety-critical smart manufacturing environments.

THEORETICAL IMPLICATIONS

The traditional "V-Model" has long served as the foundational blueprint for systems engineering and product lifecycle management, demanding a sequential progression from high-level user specifications down to component-level physical design, followed by a corresponding upward ascent through integration, verification, and validation. Historically, maintaining bidirectional requirements traceability across this V-profile has been a rigidly linear, manual, and notoriously brittle process. Changes introduced during downstream manufacturing execution or field-level validation routinely cause silent, cascading disconnections from upstream requirements, as human operators struggle to comb through nested CAD metadata, multi-tiered Bill of Materials (BOM) fragments, and complex regulatory standards.

The deployment of the Generative PLM framework fundamentally reframes this classical engineering loop by introducing an active, automated semantic orchestration layer capable of executing continuous, automated requirements traceability. When the proposed LLM framework autonomously parsed

multi-stream inputs, correlating a live machine-vision surface defect log with its precise geometric node in a CAD assembly tree and evaluating it against an unstructured ISO 26262 regulatory memo—it successfully bridged the conceptual chasm between physical manufacturing telemetry and abstract compliance rules.

This transition from static, human-curated links to a dynamic, generative traceability web converts the rigid V-model into an agile, highly interconnected lifecycle helix. Instead of treating traceability as a historical auditing task performed after a engineering change occurs, the proposed architecture allows the system to act as a proactive collaborator. It establishes a closed-loop engineering environment where a change or defect at the base of the "V" (the physical factory floor) is immediately and semantically abstracted back to its corresponding upstream design parameters and regulatory safeguards in real time. This dynamic abstraction layer dramatically increases the velocity of the product development lifecycle, minimizing systemic blind spots where undocumented floor-level reregulation can compromise overall vehicle safety and physical compliance.

PRAGMATIC DEPLOYABILITY TRADE-OFFS

While the performance metrics strongly favor the proposed frontier-tier cloud LLM workflow, integrating such systems into high-speed factory floor environments introduces multifaceted deployment trade-offs. The primary axis of optimization lies between dense semantic reasoning capabilities and computational execution overhead.

As detailed in the automated metrics, the cloud-hosted framework achieved an exceptional Macro-F1 interpretation accuracy of 89.5% and a Syntactic Schema Accuracy of 96.2%, far outpacing the edge-localized baseline model (Llama-3-8B), which scored 64.2% and 71.4% respectively. However, this analytical accuracy comes at a significant temporal cost: the proposed framework required a mean inference latency of 275 ms/token, culminating in an end-to-end processing time of 14.8 seconds per change episode, whereas the edge baseline completed execution in a rapid 2.1 seconds, offering a 9.2× speed advantage.

In the context of asynchronous PLM repository maintenance, an end-to-end latency of under 15 seconds is entirely acceptable and represents an exponential acceleration compared to the legacy manual revision cycles that historically consumed days or weeks. Yet, for deterministic, inline production line operations, such as halting a robotic welding cell or altering a high-tonnage stamping cycle mid-process, a 14.8-second delay is operationally unfeasible and could bottleneck physical manufacturing throughput.

To circumvent these operational bottlenecks, smart manufacturing enterprises must adopt a hybridized, multi-tiered cloud-edge computing topology. In this tiered architecture, low-latency edge-localized models act as primary inline filters or triage engines, handling routine data transformations and flagging straightforward structural changes locally. When the edge model detects a high-dimensional, cross-disciplinary anomaly where context histories expand or nested data structures exceed its processing limits, the episode is automatically escalated to the frontier cloud LLM tier for deep semantic auditing. This balanced approach optimizes token-cost overhead, respects factory latency constraints, and ensures that extensive parametric reasoning is rationed for complex, multi-system change request events.

LIMITATIONS

Implementing generative artificial intelligence within deeply entrenched industrial settings reveals three structural limitations that must be addressed before achieving unconstrained, autonomous enterprise deployment.

1. **Vulnerability to Dirty, Incomplete, or Unstructured Data:** While the proposed framework utilizes its parametric memory to successfully reconstruct missing CAD tags under controlled corruption testing (achieving 83.5% robustness), severe information gaps in physical telemetry remain hazardous. If critical telemetry data—such as high-tonnage stamping pressure metrics or specific part revision numbers—is missing entirely from the ingested stream, the model cannot reliably bridge the gap. It may rely on speculative assumptions to synthesize an engineering response, introducing unverified parameters into downstream systems.

2. **Software Vendor Lock-In and Systemic Rigidity:** Industrial data formatting varies widely across legacy product lifecycle management software systems and manufacturing execution layers. The current implementation relies on structured JSON schema enforcement to act as a standardized API payload for databases like Siemens Teamcenter or PTC Windchill. However, writing custom multi-agent extraction layers and parsing frameworks tailored to individual software setups creates severe deployment friction. This limits the broader generalizability of the framework and risks trapping manufacturers within specific toolchain ecosystems.
3. **The Persistent Challenge of Technical Hallucination:** A critical barrier to fully autonomous deployment remains the distinct failure profiles of large versus small models. While smaller models suffer from *Information Gapping*—collapsing under dense context loads and completely omitting critical compliance steps—the frontier LLM framework exhibits a unique failure mode termed *Narrative Expansion*. Due to its vast parametric memory and high linguistic fluency, the LLM tends to over-speculate and extrapolate elaborate causal root causes from minimal, localized defect descriptions.

For example, when evaluating an isolated surface micro-pitting log, the system may generate a complex, unverified hypothesis blaming an unrecorded structural thermal cycle variance, subsequently mandating an expansive, unnecessary downstream component rework. In safety-critical aerospace or automotive manufacturing domains, where every physical modification must be rigidly auditable, such well-formulated but ungrounded engineering conclusions are highly disruptive. They can lead to unnecessary line stoppages, unneeded component scrappage, or resource misallocation.

Consequently, the framework cannot function as a fully autonomous decision-maker. It must be securely deployed alongside deterministic validation gates, strict log-likelihood token filtering, and structured Retrieval-Augmented Generation (RAG) anchor constraints. Most importantly, it must remain bounded beneath a mandatory human-in-the-loop review panel to verify all candidate outputs before physical engineering actions are authorized on the factory floor.

CONCLUSION

This paper presented an integrated Generative PLM framework designed to automate engineering change management and preserve continuous, bidirectional requirements traceability within complex smart manufacturing environments. By layering frontier large language models over legacy Product Lifecycle Management infrastructures, the proposed architecture successfully bridges the structural chasm between unstructured shop-floor telemetry, dense CAD geometry tags, and rigorous regulatory compliance frameworks. The quantitative and qualitative evaluations demonstrate that while edge-localized models provide exceptional, low-latency execution for routine data transformations, they remain vulnerable to information gapping under expansive context loads. Conversely, the proposed frontier framework achieves superior semantic reasoning, macro-accuracy, and architectural compliance mapping. By utilizing deterministic validation gates, strict structural schema enforcement, and a mandatory human-in-the-loop audit gate, the framework safely and efficiently mitigates the risk of narrative expansion or technical hallucination, drastically accelerating engineering change validation loops without compromising vehicle safety or operational integrity.

FUTURE DIRECTIONS

While the current framework marks a critical evolution toward autonomous lifecycle orchestration, full industrial integration requires expanding the system's foundational perception and execution layers along two major technological frontiers:

Current language-bound pipelines rely heavily on text-based descriptions or structured metadata extractions of physical components. Future iterations will scale this framework into multi-modal topologies by directly integrating Vision-Language-Action (VLA) models. This advancement will allow the orchestration layer to natively ingest, interpret, and cross-examine raw 3D CAD geometries, point-cloud scans, and real-time 3D inline imaging data. By correlating physical topological deformations directly with abstract system specifications without intermediary text translation, the system can more accurately isolate manufacturing faults.

REFERENCES

[1]. Aberkane, M. S., & Otman, A. (2025). Product lifecycle management: what sectors and what technologies used? a systematic literature review. Discover Sustainability, 6(1). https://doi.org/10.1007/s43621-025-01267-w

[2]. Baker, C., Rafferty, K., & Price, M. (2025). Large language models in mechanical engineering: A scoping review of applications, challenges, and future directions. Big Data and Cognitive Computing, 9(12), 305. https://doi.org/10.3390/bdcc9120305

[3]. Bharathi Mohan, S., Gopalakrishnan, S., & Thorne, A. (2024). Automating closed-loop engineering change documentation via semantic graph mappings and transformers. Computers in Industry, 156, 104068. https://doi.org/10.1016/j.compind.2024.104068

[4]. Chiarello, F., Barandoni, S., Majda Škec, M., & Fantoni, G. (2024). Generative large language models in engineering design: opportunities and challenges. Proceedings of the Design Society, 4, 1959–1968. https://doi.org/10.1017/pds.2024.198

[5]. Daareyni, A., Martikkala, A., Mokhtarian, H., & Ituarte, I. F. (2025). Generative AI meets CAD: enhancing engineering design to manufacturing processes with large language models. The International Journal of Advanced Manufacturing Technology. https://doi.org/10.1007/s00170-025-15830-2

[6]. Doanh, D. C., Dufek, Z., Ejdys, J., Ginevičius, R., Korzynski, P., Mazurek, G., Paliszkiewicz, J., Wach, K., & Ziemba, E. (2023). Generative AI in the manufacturing process: Theoretical considerations. Engineering Management in Production and Services, 15(4), 76–89. https://doi.org/10.2478/emj-2023-0029

[7]. Ellsel, C., & Stark, R. (2025). Advancing requirements engineering with large language models. Procedia CIRP, 136, 701–706. https://doi.org/10.1016/j.procir.2025.08.120

[8]. 8. Feeney, A. B., Frechette, S. P., & Srinivasan, V. (2015). NIST Manufacturing Systems Integration STEP Datasets. National Institute of Standards and Technology (NIST) Technical Publication Series. https://www.nist.gov/el/msid

[9]. Gomez, A. P., Krus, P., Panarotto, M., & Isaksson, O. (2024). Large language models in complex system design. Proceedings of the Design Society, 4, 2197–2206. https://doi.org/10.1017/pds.2024.222

[10]. Guo, C., Liu, J., Gao, W., Lu, Z., Li, Y., Wang, C., & Yang, J. (2025). A large language model driven knowledge graph construction scheme for semantic communication. Applied Sciences, 15(8), 4575. https://doi.org/10.3390/app15084575

[11]. Hao, J., von Davier, A. A., Yaneva, V., Lottridge, S., von Davier, M., & Harris, D. J. (2024). Transforming assessment: The impacts and implications of large language models and generative AI. Educational Measurement: Issues and Practice, 43(2), 16–29. https://doi.org/10.1111/emip.12602

[12]. Ho, L., Eric Ka, Nandhini, B., Hang, L., Carmen Kar, & Shenghao, X. (2025). The Generative Artificial Intelligence large language product design multi-model framework for manufacturing operations. Figshare. https://doi.org/10.6084/m9.figshare.30383961.v1

[13]. Hoang, D., Gorsich, D., Castanier, M. P., & Imani, F. (2025). Knowledge Graph Fusion with Large Language Models for Accurate, Explainable Manufacturing Process Planning. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2506.13026

[14]. Ilyas, I., Hafeez, Y., Hussain, S., Ali, S., Jammal, M., & Toure, I. K. (2025). Optimized change management process through semantic requirements and traceability analysis tool. Journal of Engineering, 2025(1). https://doi.org/10.1155/je/2296387

[15]. ISO. (2018). ISO 26262-1:2018 Road vehicles — Functional safety — Part 1: Vocabulary. International Organization for Standardization.

[16]. Jiang, Z., Liu, A., Zhang, D., Xu, X., & Dai, Y. (2025). Customization and personalization of large language models for engineering design. CIRP Annals, 74(1), 191–195. https://doi.org/10.1016/j.cirp.2025.03.001

[17]. Jose, A., Ramaswamy, S., & Harrison, R. (2024). Generative AI pipelines for legacy PLM synchronization: Limitations and frontiers in schema-mapping. International Journal of Production Research, 62(14), 5189–5207. https://doi.org/10.1080/00207543.2024.2391012

[18]. Jradi, T., Violos, J., Spatharakis, D., Mavraidi, L., Dimolitsas, I., Leivadeas, A., & Papavassiliou, S. (2026). Intent-Driven Smart Manufacturing Integrating Knowledge Graphs and Large Language Models. ArXiv.Org. https://arxiv.org/pdf/2602.12419

[19]. Kampik, T., Warmuth, C., Rebmann, A., Agam, R., Egger, L. N. P., Gerber, A., Hoffart, J., Kolk, J., Herzig, P., Decker, G., van der Aa, H., Polyvyanyy, A., Rinderle-Ma, S., Weber, I., & Weidlich, M. (2024). Large process models: A vision for business process management in the age of generative AI. KI - Künstliche Intelligenz, 39(2), 81–95. https://doi.org/10.1007/s13218-024-00863-8

[20]. Kchaou, D., Bouassida, N., Mefteh, M., & Ben-Abdallah, H. (2019). Recovering semantic traceability between requirements and design for change impact analysis. Innovations in Systems and Software Engineering, 15(2), 101–115. https://doi.org/10.1007/s11334-019-00330-w

[21]. Koch, J. (2017). Manufacturing Change Management – a Process-Based Approach for the Management of Manufacturing Changes. mediaTUM – the Media and Publications Repository of the Technical University Munich (Technical University Munich). http://mediatum.ub.tum.de/doc/1346233/document.pdf

[22]. Kourani, H., Berti, A., Schuster, D., & van der Aalst, W. M. P. (2024). Process modeling with large language models. In Lecture Notes in Business Information Processing (pp. 229–244). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-61007-3_18

[23]. Li, S., & Corney, J. (2025). MechRAG: a multimodal large language model for mechanical engineering. Communications Engineering, 4(1). https://doi.org/10.1038/s44172-025-00517-z

[24]. Li, X., Lyu, M., Wang, Z., Chen, C.-H., & Zheng, P. (2021). Exploiting knowledge graphs in industrial products and services: A survey of key aspects, challenges, and future perspectives. Computers in Industry, 129, 103449. https://doi.org/10.1016/j.compind.2021.103449

[25]. Li, Y., Zhao, H., Jiang, H., Pan, Y., Liu, Z., Wu, Z., Shu, P., Tian, J., Yang, T., Xu, S., Lyu, Y., Blenk, P., Pence, J., Rupram, J., Banu, E., Liu, N., Wang, L., Song, W., Zhai, X., … Liu, T. (2024). Large Language Models for Manufacturing. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2410.21418

[26]. Luitel, D., Hassani, S., & Sabetzadeh, M. (2024). Improving requirements completeness: automated assistance through large language models. Requirements Engineering, 29(1), 73–95. https://doi.org/10.1007/s00766-024-00416-3

[27]. Lyu, Y., Cho, H., Jung, P., & Lee, S. (2023). A systematic literature review of issue-based requirement traceability. IEEE Access, 11, 13334–13348. https://doi.org/10.1109/access.2023.3242294

[28]. Makatura, L., Foshey, M., Wang, B., HähnLein, F., Ma, P., Deng, B., Tjandrasuwita, M., Spielberg, A., Owens, C. E., Chen, P. Y., Zhao, A., Zhu, A., Norton, W., Gu, E., Jacob, J., Li, Y., Schulz, A., & Matusik, W. (2023). How Can Large Language Models Help Humans in Design and Manufacturing? arXiv (Cornell University). https://doi.org/10.48550/arxiv.2307.14377

[29]. Makatura, L., Foshey, M., Wang, B., Hähnlein, F., Ma, P., Deng, B., Tjandrasuwita, M., Spielberg, A., Owens, C., Chen, P. Y., Zhao, A., Zhu, A., Norton, W., Gu, E., Jacob, J., Li, Y., Schulz, A., & Matusik, W. (2024a). How can large language models help humans in design and manufacturing? part 2: Synthesizing an end-to-end LLM-Enabled design and manufacturing workflow. Harvard Data Science Review, Special Issue 5. https://doi.org/10.1162/99608f92.0705d8bd

[30]. Makatura, L., Foshey, M., Wang, B., Hähnlein, F., Ma, P., Deng, B., Tjandrasuwita, M., Spielberg, A., Owens, C. E., Chen, P. Y., Zhao, A., Zhu, A., Norton, W. J., Gu, E., Jacob, J., Li, Y., Schulz, A., & Matusik, W. (2024b). Large language models for design and manufacturing. An MIT Exploration of Generative AI. https://doi.org/10.21428/e4baedd9.745b62fa

[31]. Necula, S.-C., Dumitriu, F., & Greavu-Şerban, V. (2024). A systematic literature review on using natural language processing in software requirements engineering. Electronics, 13(11), 2055. https://doi.org/10.3390/electronics13112055

[32]. Nguyen, S. S., & Kittur, J. (2025). Review of current and potential uses of large language models in engineering. Frontiers in Education, 10. https://doi.org/10.3389/feduc.2025.1649650

[33]. Norheim, J. J., Rebentisch, E., Xiao, D., Draeger, L., Kerbrat, A., & de Weck, O. L. (2024). Challenges in applying large language models to requirements engineering tasks. Design Science, 10. https://doi.org/10.1017/dsj.2024.8

[34]. Quang Hai Khuat. (2025). Leveraging generative AI for data engineering workflows. Journal of Computer Science and Technology Studies, 7(3), 120–140. https://doi.org/10.32996/jcsts.2025.7.3.14

[35]. Rammo, J.-P., Stang, J., Agyekum, E., & Zaeh, M. F. (2024). Requirements elicitation for a flexible, digital, and methodological manufacturing change management based on a literature review and expert interviews. Procedia CIRP, 130, 1090–1095. https://doi.org/10.1016/j.procir.2024.10.211

[36]. Rammo, J.-P., Graf, J., Rouvelle, C. R., Atzenbeck, M., Klages, B., Reuter, C., & Zaeh, M. F. (2025). Evaluation of a methodology for a change-specific and company-individual manufacturing change management – industrial application, requirement fulfilment, and cost-benefit analysis. 2025 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 0329–0334. https://doi.org/10.1109/ieem63636.2025.11357581

[37]. Raza, M., Monostori, L., & Váncza, J. (2025). Agentic artificial intelligence at the industrial edge: Real-time manufacturing triage via tiered LLM orchestration. Journal of Manufacturing Systems, 74, 112–126. https://doi.org/10.1016/j.jmsy.2024.11.003

[38]. Raza, M., Jahangir, Z., Riaz, M. B., Saeed, M. J., & Sattar, M. A. (2025). Industrial applications of large language models. Scientific Reports, 15(1). https://doi.org/10.1038/s41598-025-98483-1

[39]. Shafiee, S. (2025). Generative AI in manufacturing: a literature review of recent applications and future prospects. Procedia CIRP, 132, 1–6. https://doi.org/10.1016/j.procir.2025.01.001

[40]. Song, K., & Yan, Y. (2013). A noise robust method based on local binary patterns for steel surface defect detection. Journal of Iron and Steel Research International, 20(5), 89-94. https://doi.org/10.1016/S1006-706X(13)60105-X

[41]. Uygun, Y., & Momodu, V. (2024). Local large language models to simplify requirement engineering documents in the automotive industry. Production & Manufacturing Research, 12(1). https://doi.org/10.1080/21693277.2024.2375296

[42]. van Remmen, J. S., Horber, D., Lungu, A., Chang, F., van Putten, S., Goetz, S., & Wartzack, S. (2023). Natural language processing in requirements engineering and its challenges for requirements modelling in the engineering design domain. Proceedings of the Design Society, 3, 2765–2774. https://doi.org/10.1017/pds.2023.277

[43]. Vogelsang, A., & Fischbach, J. (2024). Using Large Language Models for Natural Language Processing Tasks in Requirements Engineering: A Systematic Guideline. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2402.13823

[44]. Yahya, M., Breslin, J. G., & Ali, M. I. (2021). Semantic web and knowledge graphs for industry 4.0. Applied Sciences, 11(11), 5110. https://doi.org/10.3390/app11115110

[45]. Yuan, X., Chen, J., Wang, Y., Chen, A., Huang, Y., Zhao, W., & Yu, S. (2024). Semantic-Enhanced knowledge graph completion. Mathematics, 12(3), 450. https://doi.org/10.3390/math12030450