International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 1280-1289
https://doi.org/10.70917/ijcisim-2026-5519
Article

Managing Engineering Documentation Knowledge at Scale: Adobe Experience Manager Architecture and Artificial Intelligence Automation for Technical Content in Manufacturing Enterprises

Sham Sunder Reddy Dadi

University of Central Missouri, USA
ORCID iD: 0009-0003-8649-1962

Abstract: Manufacturing enterprises generate large volumes of technical documentation, including product specifications, work instructions, and engineering change notices, across multiple plants and business units, and this documentation is frequently fragmented across disconnected file shares, legacy document systems, and plant-specific repositories. Existing engineering knowledge management research has developed ontology-based and knowledge-graph-based approaches to structuring this content, but these approaches typically require dedicated ontology engineering effort and are not always compatible with the content platforms manufacturing organizations already operate for other purposes. This paper proposes an architecture that combines a structured web content and digital asset management platform, Adobe Experience Manager, with artificial-intelligence-assisted automation, including auto-tagging, metadata extraction, and duplicate detection, to manage engineering documentation as structured, reusable knowledge units at enterprise scale. The architecture is illustrated through a multi-plant scenario involving specifications, work instructions, and engineering change notices, and is discussed in relation to knowledge-graph-based alternatives from the recent literature. The paper argues that a content-platform-centric approach offers a lower-barrier entry point for organizations that have not yet invested in formal ontology engineering, while identifying the specific knowledge-representation capabilities such an approach cannot substitute for. Limitations and directions for empirical validation are discussed.
Keywords: engineering documentation management; digital asset management; knowledge representation; manufacturing knowledge reuse; artificial intelligence automation

1. Introduction

Manufacturing enterprises depend on technical documentation to translate engineering intent into consistent production outcomes. Product specifications define what must be built, work instructions define how it must be built, and engineering change notices record how the definition has evolved. Each of these document types carries knowledge that is reused repeatedly across plants, shifts, and product variants, and the reliability of that reuse depends on whether the documentation is structured, current, and discoverable at the point of use.

In practice, this documentation is often fragmented. Multi-plant organizations frequently accumulate parallel copies of similar specifications, inconsistent numbering and metadata conventions across sites, and work instructions that reference engineering change notices no longer in force. Fragmentation of this kind is a known problem in engineering knowledge management, and it has motivated a substantial body of research on ontology-based knowledge representation [2,4,6] and, more recently, knowledge-graph construction from manufacturing documents

[11,12,13,14]. This research has produced genuine technical progress: large language models can now extract structured entities and relations from unstructured manufacturing text with useful accuracy [13,14], and knowledge graphs built from CAD repositories and process-planning documents can support retrieval and reuse at a scale manual indexing cannot match [11,12].

This progress, however, generally presumes that an organization is prepared to invest in ontology engineering and knowledge-graph infrastructure specifically for its engineering documentation. Many manufacturing organizations are not at that stage. They already operate structured content and digital asset management platforms, most commonly for marketing, product information, or general enterprise content, and the practical question this paper takes up is whether that existing platform investment can be extended to engineering documentation with substantially less specialized infrastructure than a dedicated knowledge graph requires, while still gaining the reuse and discoverability benefits the knowledge-graph literature has documented.

Adobe Experience Manager (AEM) is one such platform. Its structured content model, comprising content fragments, metadata schemas, digital asset management, and workflow-driven review, was designed for general enterprise content rather than engineering knowledge specifically, but the underlying primitives, structured metadata, reusable content units, and governed workflows, map onto the requirements engineering documentation management shares with any large-scale structured-content problem. Artificial intelligence adds a further capability that is increasingly common in the knowledge-graph literature [13,14,17,18] but has not been examined in the context of general-purpose content platforms: automated tagging, metadata extraction, and duplicate detection applied directly to engineering documents as they are authored or ingested, rather than as a separate knowledge-graph construction pipeline.

This paper makes three contributions. First, it proposes an architecture for managing engineering documentation as structured, reusable knowledge units using a general-purpose content and digital-asset-management platform augmented with AI-assisted automation, rather than a dedicated knowledge-graph infrastructure. Second, it illustrates the architecture through a multi-plant scenario spanning specifications, work instructions, and engineering change notices. Third, it discusses explicitly where this content-platform-centric approach is a reasonable substitute for knowledge-graph-based methods and where it is not, since the two approaches trade different costs for different capabilities and neither is universally preferable. The remainder of the paper reviews the relevant literature, develops the proposed architecture, works through the illustrative scenario, and discusses the approach's scope and limitations.

2. Related Work

2.1 Engineering Knowledge Management and Product Lifecycle Management

Engineering knowledge management has long recognized that product knowledge accumulates across a lifecycle that extends well past initial design. Grieves [1] framed product lifecycle management (PLM) as an enterprise paradigm built on the premise that information generated at each lifecycle stage has value at every other stage, provided it is captured and made accessible rather than lost in stage-specific silos. Ameri and Dutta [3] extended this framing explicitly in knowledge-management terms, describing PLM's central technical problem as closing the knowledge loop between design, manufacturing, and field use. Alavi and Leidner's foundational review of knowledge management systems [2], while not specific to engineering, supplies the conceptual vocabulary this literature depends on: knowledge management systems succeed when they support the creation, storage, transfer, and application of organizational knowledge, and engineering documentation is a direct instance of the storage-and-transfer problem at enterprise scale. Verhagen et al.'s critical review of Knowledge-Based Engineering [4], published in this journal, identifies a persistent challenge specific to engineering domains: knowledge capture tools are frequently built by and for specialists, which limits adoption by the broader engineering workforce that actually produces and consumes technical documentation day to day. That adoption challenge motivates the platform-centric approach developed later in this paper, since a platform organizations already use for other content is, by construction, more broadly accessible than purpose-built knowledge-capture tooling.

2.2 Structured Representation and Standards for Engineering Documentation

Standardized data exchange has been a parallel thread in this literature. Pratt's account of ISO 10303 (STEP) [5] describes how standardized product data representations allow engineering information to move between systems without losing structure, a capability engineering documentation management depends on just as much as CAD data exchange does. Matsokis and Kiritsis [6] proposed an ontology-based approach to PLM specifically to give this kind of structured representation a formal semantic backbone, allowing systems to reason over relationships between product data rather than merely storing it. Chhim, Chinnam, and Sadawi [7] built on this line of work with a design-and-manufacturing-process ontology explicitly aimed at knowledge reuse, validated against real manufacturing cases, demonstrating that the reuse gains formal representation promises are achievable in practice rather than only in principle. Yang et al. [8], writing in this journal, proposed an ontology for industrial production workflows intended to integrate heterogeneous information systems within a single plant, a problem that scales into the multi-plant fragmentation problem this paper addresses when the same heterogeneity exists across sites rather than within one.

These contributions share a common cost: formal ontology construction requires specialized modeling expertise and sustained maintenance effort, which is precisely the adoption barrier Verhagen et al. [4] identify. The architecture proposed in Section 3 does not attempt to replace this body of work; it targets organizations for whom that cost has not yet been justified, and asks how much structure a general-purpose content platform can provide before formal ontology engineering becomes necessary.

2.3 Knowledge Graphs and AI-Assisted Extraction from Manufacturing Documents

Recent work has moved from static ontology design toward automated knowledge-graph construction directly from manufacturing artifacts and documents, substantially reducing the manual-authoring cost that limited earlier ontology efforts. Bharadwaj and Starly [11] constructed a knowledge graph from a repository of over 110,000 CAD models, demonstrating that automated graph construction can operate at a scale manual cataloguing cannot approach. Liu et al. [12], also in this journal, proposed a knowledge-graph-based representation for heterogeneous Industrial-Internet-of-Things manufacturing data intended to support cognitive decision-making across otherwise disconnected data sources. Wen, Ma, and Wang [13] applied structured knowledge modeling and extraction specifically to manufacturing process planning documents, extracting entities across eight information categories from 238 products, a scale and structure close to the specification and work-instruction extraction this paper's architecture targets. Most directly relevant, Shim et al.'s OmEGa framework [14] uses a large language model within an ontology-based extraction pipeline to construct task-centric knowledge graphs directly from manufacturing documents, including multimodal content such as text, images, and tables, which is close to the auto-tagging and metadata-extraction capability this paper proposes to apply within a content platform rather than a standalone knowledge-graph pipeline.

Parallel work has examined AI-assisted extraction for adjacent document types. Wang, Cheng, Qi, and Tao [17] built a knowledge graph from heterogeneous maintenance documents for industrial dataspaces, and Ding et al. [18] constructed a failure knowledge graph from sparsely labeled maintenance text using semi-supervised learning, a setting where labeled training data is scarce, much as it typically is for engineering change notices. Elyan, Jamieson, and Ali-Gombe [19] demonstrated deep-learning classification of symbols within engineering drawings at accuracy above ninety-four percent on an industrial drawing collection, evidence that automated classification of engineering-specific visual and symbolic content, not just narrative text, is technically achievable. Jiang, Hu, Magee, and Luo [20] developed a multimodal deep-learning architecture for enterprise-scale hierarchical classification of technical documents, reporting accuracy gains over unimodal baselines on a large document corpus, directly supporting the feasibility of automated document classification at the scale a multi-plant engineering-documentation system requires. Zhang et al. [21] extracted domain ontologies automatically from engineering documents using a biologically inspired approach, evaluating extraction accuracy against manually curated ontologies and finding the automated approach a workable substitute in practice.

Across this literature, human oversight of AI-generated knowledge structures remains a live concern rather than a solved problem. Kim et al.'s review of human-in-the-loop practices in smart manufacturing [22] surveys twenty-one

technologies across seven clusters and concludes that human review remains structurally necessary wherever AI-generated outputs feed decisions with real consequences, which includes the metadata and classification decisions an auto-tagging layer would generate for engineering documentation. This constraint, together with the industry-scale evidence from Leoni et al. [9] that knowledge-management process maturity mediates whether AI actually improves manufacturing outcomes, rather than AI capability alone, and Wang and Yang's [10] finding that knowledge-management success correlates with measurable productivity gains across a large sample of knowledge workers, motivates the architecture developed in the next section: AI automation is treated as a capability embedded within a governed content platform, not a standalone extraction pipeline operating without human review.

Adoption of any of these extraction and knowledge-graph techniques inside a general-purpose content platform, rather than a standalone research pipeline, introduces constraints the underlying methods were not necessarily evaluated against. A named-entity recognizer trained on one organization's process-planning corpus [13] or a classifier validated on a specific technical-document collection [20] carries no guarantee of transferring cleanly to a different organization's terminology, numbering conventions, or document templates without retraining or fine-tuning, a practical constraint the architecture in Section 3 addresses through a human-confirmation step rather than assuming zero-shot transfer will be reliable.

2.4 Organizational Adoption of Structured Documentation Systems

A separate strand of literature, less concerned with representation formalism and more concerned with organizational behavior, has examined why structured knowledge-management systems succeed or fail to achieve sustained use once deployed. Alavi and Leidner's review [2] already identifies this as a central concern of the field: a knowledge management system's technical capability is not sufficient for adoption if the system does not fit the workflows of the people expected to use it day to day. Verhagen et al.'s observation [4] that engineering knowledge-capture tools are frequently built by and for specialists translates this general concern into the specific engineering-documentation context this paper addresses, and it is the primary justification for building on a general-purpose content platform rather than a specialist knowledge-engineering tool: the platform's authoring workflows are already familiar to the document owners across plants who will be expected to confirm AI-generated metadata and resolve reconciliation candidates, rather than requiring them to learn an unfamiliar ontology-authoring interface.

Leoni et al.'s survey findings [9] give this adoption concern an empirical anchor specific to AI-augmented systems: across the manufacturing firms they studied, artificial intelligence's effect on operational outcomes was mediated by the maturity of the firm's existing knowledge-management processes rather than acting directly. Read alongside Wang and Yang's finding that knowledge-management success correlates with measurable productivity gains at the individual-worker level [10], the implication for this paper's architecture is that its AI-assisted layers are best understood as amplifying whatever document-ownership and review discipline already exists, not as a substitute for building that discipline in the first place, a point returned to directly in Section 6.

3. Proposed Architecture

The architecture treats engineering documentation as a structured knowledge asset managed through four coordinated layers: a structured content model, an AI-assisted extraction and tagging layer, a governed reuse and workflow layer, and a plant-to-enterprise metadata reconciliation layer. Each layer maps to a specific fragmentation problem identified in Section 2 and to a corresponding platform capability.

Structured content model.

Engineering documents, specifications, work instructions, and engineering change notices, are represented as content fragments rather than monolithic files. A content fragment separates structured fields, part number, revision, plant, effective date, related change notices, from presentation, so the same underlying specification can be rendered for a shop-floor terminal, a supplier portal, or an audit export without being duplicated as separate files. This is the same structural move ontology-based PLM approaches make when they separate product data from its presentation [6], implemented through a content platform's native content-fragment model rather than a purpose-built ontology

store. Metadata schemas attached to each document type enforce, at authoring time, the fields a knowledge-graph extraction pipeline would otherwise need to infer after the fact from unstructured text [13,14].

AI-assisted extraction and tagging.

When documents originate outside the platform, migrated legacy specifications, scanned work instructions, supplier-submitted drawings, an AI-assisted layer performs the extraction and tagging that structured authoring would otherwise require manually. Following the extraction approach demonstrated in this journal's recent work [13,14], the layer combines named-entity recognition over document text to populate metadata fields, such as part numbers, revision identifiers, and referenced change notices, with a classifier trained to route documents into the correct document-type and plant categories, in the manner demonstrated for hierarchical technical-document classification at enterprise scale [20]. Where documents include engineering drawings rather than narrative text, symbol- and diagram-level classification of the kind demonstrated by Elyan, Jamieson, and Ali-Gombe [19] extends the same tagging capability to visual content. Consistent with the human-in-the-loop consensus in this literature [22], every AI-generated tag and extracted field is presented to a document owner as a proposed value requiring confirmation before it becomes authoritative metadata, rather than being written directly into the system of record.

The metadata schema underlying the structured content model distinguishes required fields from optional fields at the document-type level. A specification's required fields include part number, revision identifier, effective date, originating plant, and status (draft, under review, released, superseded); a work instruction's required fields add the specification it implements and the process step it governs; an engineering change notice's required fields add the part numbers it affects and the specifications and work instructions it supersedes. This differentiation matters because it determines what the AI-assisted extraction layer is responsible for populating when a document is ingested rather than authored directly: required fields absent from a legacy document trigger a mandatory human-review flag before the document can be marked released, while optional fields can be populated opportunistically without blocking workflow progress.

Governed reuse and workflow.

Once documents are structured and tagged, a workflow layer governs how they are reused and revised. An engineering change notice, on approval, can automatically flag every specification and work instruction that references the changed part number for review, using the structured metadata relationships established at authoring or extraction time rather than requiring a manual cross-reference search. This is the specific reuse capability the ontology literature has pursued through formal relationship modeling [6,7]; the architecture proposed here achieves a bounded version of it through the content platform's native reference and workflow model, at lower implementation cost but with correspondingly less expressive relationship modeling than a full ontology would support.

Plant-to-enterprise metadata reconciliation.

Multi-plant fragmentation is addressed through a reconciliation layer that maps plant-specific metadata conventions, different part-numbering schemes, different document-type naming, onto a shared enterprise schema, while preserving the plant-specific values as secondary fields rather than discarding them. This reconciliation is where AI-assisted matching is most useful: a similarity-based duplicate-detection pass, applied across plants rather than within a single repository, can surface candidate matches between differently named documents describing the same underlying specification, for a human reviewer to confirm or reject, extending duplicate-detection approaches from the broader content-management literature to the specific problem of cross-plant document reconciliation.

Table 1 summarizes the four layers, the fragmentation problem each addresses, and the literature that motivates the corresponding design choice.

Table 1. Architecture Layers and Corresponding Fragmentation Problems

Layer	Fragmentation Problem Addressed	Motivating Literature

Structured content model	Duplicated files with inconsistent structure across plants	Matsokis & Kiritsis (2010); Grieves (2005)
AI-assisted extraction and tagging	Legacy and supplier documents lack structured metadata	Wen et al. (2023); Shim et al. (2025); Jiang et al. (2024)
Governed reuse and workflow	Manual cross-referencing of change notices against affected documents	Chhim et al. (2019); Yang et al. (2023)
Plant-to-enterprise metadata reconciliation	Inconsistent part-numbering and naming conventions across plants	Wang, Cheng, Qi & Tao (2024); Ding et al. (2024)

Table 2 details the metadata schema referenced in Section 3, distinguishing required from optional fields across the three document types the illustrative scenario in Section 4 works with.

Table 2. Metadata Schema by Document Type

Document Type	**Required Fields**	**Optional Fields**
Specification	Part number; revision identifier; effective date; originating plant; status	Material grade; supplier reference; related drawings
Work instruction	Implemented specification; process step; originating plant; status	Tooling reference; estimated cycle time; safety notes
Engineering change notice	Affected part numbers; superseded specifications/work instructions; effective date; approving authority	Change rationale; cost impact estimate; supplier notification status

The architecture proceeds as a structured flow: document origination, whether authored directly in the platform or ingested from legacy or supplier sources, feeds into the AI-assisted extraction and tagging layer, which feeds the governed reuse and workflow layer, which feeds the plant-to-enterprise reconciliation layer, with human confirmation checkpoints positioned after both the extraction/tagging step and the reconciliation-matching step, consistent with the human-in-the-loop requirement identified in Section 2.3.

4. Illustrative Application

This section works through a scenario to illustrate how the four layers operate together. The scenario is illustrative rather than an implemented deployment, and is intended to make the architecture's behavior concrete rather than to serve as validation; empirical validation is discussed as future work in Section 6.

Consider a manufacturing enterprise operating three plants that each produce variants of a common assembly. Plant A originates an engineering change notice revising a torque specification for a fastener used across all three variants. Under the architecture proposed here, the change notice is authored as a structured content fragment carrying the affected part number, the revision identifier, and the effective date. The governed reuse layer, using the reference relationships established when the original specification and work instructions were authored, automatically identifies every specification and work instruction across all three plants that references the affected part number and routes each to its plant's document owner for review, rather than relying on Plant B and Plant C independently noticing the change through informal channels.

Plant B, in this scenario, holds a set of legacy work instructions inherited from a prior document system and never re-authored in the new platform. When these are ingested, the AI-assisted extraction layer applies named-entity recognition to identify part numbers, revision references, and plant identifiers embedded in the legacy text, proposing metadata values that Plant B's document owner confirms before they become authoritative. Where a legacy work instruction references a torque specification using Plant B's historical naming convention rather than the enterprise part number, the plant-to-enterprise reconciliation layer's similarity matching surfaces the likely correspondence

between the legacy identifier and the enterprise part number as a candidate match, again requiring human confirmation rather than an automatic overwrite.

Once reconciled, the enterprise gains a single, queryable view of every document affected by the torque specification change, across all three plants, without any plant needing to have manually cross-referenced the others. The knowledge-graph literature reviewed in Section 2.3 would achieve a comparable outcome through explicit graph traversal over formally modeled relationships [11,12,13]; this architecture achieves a bounded version of the same outcome through the content platform's native reference model and AI-assisted matching, at the cost of less expressive relationship reasoning than a full knowledge graph supports, a trade-off discussed further in Section 5.

A second element of the same scenario illustrates the extraction layer's behavior under a harder case. Plant C receives a supplier-submitted drawing package for a redesigned bracket, arriving as scanned PDF sheets rather than structured data, with no accompanying metadata beyond a supplier-assigned drawing number. The AI-assisted extraction layer applies the drawing-symbol classification approach demonstrated in the literature [19] to identify title-block fields, revision tables, and referenced part numbers directly from the scanned sheets, proposing a specification record populated from these extracted fields. Because the supplier's drawing number does not match any enterprise part number, the plant-to-enterprise reconciliation layer's similarity matching checks the extracted dimensional and material attributes against existing specifications for related brackets, surfacing two candidate matches, an existing bracket specification it may supersede and an unrelated bracket from a different product line, for Plant C's document owner to distinguish. This step is deliberately not automated past the point of surfacing candidates, since a false match at this stage, silently linking the new bracket to the wrong existing specification, would propagate an incorrect change-notice relationship through the entire governed-reuse layer described above.

5. Discussion

The architecture proposed here is best understood as occupying a specific position between unstructured file-share documentation and formal knowledge-graph infrastructure, rather than as a replacement for either. Its central claim is that a content platform an organization already operates, augmented with AI-assisted extraction, tagging, and duplicate detection, can close a meaningful portion of the fragmentation gap identified in Section 1 at substantially lower setup cost than ontology engineering or dedicated knowledge-graph construction requires, since it reuses existing platform investment and requires configuring metadata schemas and workflows rather than building a semantic model from scratch.

This position has real limits, and a fair comparison requires stating them plainly. Table 3 sets out the trade-off directly. Knowledge-graph approaches support relationship reasoning that a content platform's reference model does not: a knowledge graph can answer a query about indirect, multi-hop relationships between a material property, a supplier, and a set of affected assemblies [11,12], where the architecture proposed here can only follow the explicit references its authors or extraction pipeline established. Organizations whose engineering knowledge genuinely requires this kind of multi-hop reasoning, rather than the direct change-notice-to-affected-document reuse illustrated in Section 4, are better served by the ontology and knowledge-graph literature this paper builds on rather than by the architecture it proposes.

Table 3. Content-Platform-Centric Versus Knowledge-Graph-Centric Approaches

Dimension	**Content-Platform-Centric (this paper)**	**Knowledge-Graph-Centric**
Setup cost	Low; reuses existing platform investment	High; requires ontology/schema engineering
Relationship reasoning	Direct references only	Multi-hop graph traversal
Adoption barrier	Low; familiar authoring workflows	Higher; specialized modeling skills required

Best suited for	Direct reuse and change-notice propagation	Complex cross-domain knowledge queries

A second limitation concerns the AI-assisted layer specifically. Extraction and classification accuracy in the literature this paper draws on is high but not perfect [13,14,19,20], and every false tag or missed duplicate that reaches production metadata without correction propagates the same fragmentation the architecture is meant to resolve, only now with a false appearance of structure. The human-confirmation checkpoints built into Section 3's design are the direct response to this risk, consistent with the human-in-the-loop consensus in the smart-manufacturing literature [22], but they also mean the architecture's efficiency gains are bounded by how much confirmation burden document owners are actually willing to absorb, a question this paper cannot answer without the empirical work identified in Section 6.

A related and underexamined risk is metadata drift across successive AI-assisted extraction passes. If a classifier's confidence threshold for auto-populating a field is set too permissively, incorrect values can accumulate across many documents before a human reviewer notices a pattern, since each individual confirmation step examines one document at a time rather than the aggregate tagging behavior across the corpus. Mitigating this requires periodic aggregate auditing of AI-assigned metadata, comparing tag distributions against expected baselines, a monitoring practice that sits outside the per-document workflow described in Section 3 and would need to be added as an operational discipline rather than a platform feature.

A third consideration concerns organizational readiness rather than technical capability. Leoni et al.'s survey evidence [9] indicates that AI's benefit to manufacturing firms is mediated by knowledge-management process maturity rather than delivered by AI capability alone, and Wang and Yang's findings [10] link measurable knowledge-management success to broader productivity outcomes. Applied to this architecture, the implication is that the AI-assisted layers proposed in Section 3 are unlikely to deliver their intended benefit in an organization that lacks basic document-ownership and review discipline already, regardless of how accurate the underlying extraction and classification models prove to be.

6. Practical Implications for Adoption

For organizations evaluating this architecture, the practical starting point is an honest inventory of existing document fragmentation rather than an immediate platform or AI procurement decision. The scenario in Section 4 assumes structured metadata schemas already exist for each document type; defining those schemas, and securing agreement across plants on shared versus plant-specific fields, is organizational work that has to precede any AI-assisted extraction, and skipping it in favor of extraction tooling alone tends to reproduce the fragmentation the architecture is meant to resolve, only encoded in inconsistent metadata rather than inconsistent files.

Document owners, rather than IT or platform administrators, are the practical bottleneck this architecture depends on. Every human-confirmation checkpoint described in Section 3 assumes a specific person at each plant is responsible for reviewing proposed metadata and candidate matches in a timely way; without that ownership assigned and resourced, AI-generated proposals accumulate unconfirmed, and the metadata-drift risk identified in Section 5 becomes more likely rather than less. Assigning document ownership explicitly, and measuring confirmation turnaround as an operational metric, is accordingly a precondition for the architecture's benefits rather than an afterthought.

Finally, organizations with genuinely complex cross-domain knowledge requirements, spanning, for instance, material science data, supplier qualification records, and regulatory compliance documentation in combination, should treat this architecture as a starting point rather than a destination. The trade-off in Table 3 is not static: an organization may reasonably begin with the content-platform-centric approach proposed here and migrate specific high-value knowledge domains to formal ontology or knowledge-graph infrastructure as the underlying reasoning requirements outgrow what direct references and AI-assisted tagging can support, rather than treating the two approaches as a single upfront choice.

7. Conclusion and Future Work

This paper has proposed an architecture for managing engineering documentation as structured, reusable knowledge, using a general-purpose content and digital-asset-management platform augmented with AI-assisted extraction, tagging, and duplicate detection, as a lower-barrier alternative to dedicated knowledge-graph infrastructure for organizations not yet positioned to invest in formal ontology engineering. The architecture was illustrated through a multi-plant scenario involving specification, work-instruction, and engineering-change-notice reuse, and was discussed against knowledge-graph-based alternatives to identify where each approach is the more appropriate choice.

The architecture has not been empirically validated against operational deployment data, and this is the paper's principal limitation. Future work should evaluate the architecture against a real multi-plant documentation environment, measuring extraction and tagging accuracy against ground-truth metadata, confirmation burden on document owners, and time-to-resolution for change-notice propagation compared against the manual cross-referencing process it is intended to replace. A second direction concerns the boundary condition identified in Section 5: future work could develop criteria for predicting, before implementation, whether a given organization's documentation problem is better served by this architecture or by investment in a full knowledge-graph approach, rather than leaving that choice to post hoc judgment.

8. CRediT Authorship Contribution Statement

Sham Sunder Reddy Dadi: Conceptualization, Methodology, Writing - original draft, Writing - review and editing.

9. Declaration of Competing Interest

The author declares no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

10. Acknowledgments

None.

REFERENCES

1. M.W. Grieves, Product lifecycle management: the new paradigm for enterprises, International Journal of Product Development 2 (1/2) (2005) 71-84. https://doi.org/10.1504/IJPD.2005.006669
2. M. Alavi, D.E. Leidner, Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues, MIS Quarterly 25 (1) (2001) 107-136. https://doi.org/10.2307/3250961
3. F. Ameri, D. Dutta, Product lifecycle management: closing the knowledge loops, Computer-Aided Design and Applications 2 (5) (2005) 577-590. https://doi.org/10.1080/16864360.2005.10738322
4. W.J.C. Verhagen, P. Bermell-Garcia, R.E.C. van Dijk, R. Curran, A critical review of Knowledge-Based Engineering: An identification of research challenges, Advanced Engineering Informatics 26 (1) (2012) 5-15. https://doi.org/10.1016/j.aei.2011.06.004
5. M.J. Pratt, ISO 10303, the STEP standard for product data exchange, and its PLM capabilities, International Journal of Product Lifecycle Management 1 (1) (2005) 86-94. https://doi.org/10.1504/IJPLM.2005.007347
6. A. Matsokis, D. Kiritsis, An ontology-based approach for Product Lifecycle Management, Computers in Industry 61 (8) (2010) 787-797. https://doi.org/10.1016/j.compind.2010.05.007
7. P. Chhim, R.B. Chinnam, N. Sadawi, Product design and manufacturing process based ontology for manufacturing knowledge reuse, Journal of Intelligent Manufacturing 30 (2) (2019) 905-916. https://doi.org/10.1007/s10845-016-1290-2
8. C. Yang, Y. Zheng, X. Tu, R. Ala-Laurinaho, J. Autiosalo, O. Seppanen, K. Tammi, Ontology-based knowledge representation of industrial production workflow, Advanced Engineering Informatics 58 (2023) 102185. https://doi.org/10.1016/j.aei.2023.102185
9. L. Leoni, M. Ardolino, J. El Baz, G. Gueli, A. Bacchetti, The mediating role of knowledge management processes in the effective use of artificial intelligence in manufacturing firms, International Journal of Operations & Production Management 42 (13) (2022) 411-437. https://doi.org/10.1108/IJOPM-05-2022-0282

10. M.-H. Wang, T.-Y. Yang, Investigating the success of knowledge management: An empirical study of small- and medium-sized enterprises, Asia Pacific Management Review 21 (2) (2016) 79-91. https://doi.org/10.1016/j.apmrv.2015.12.003
11. A.G. Bharadwaj, B. Starly, Knowledge graph construction for product designs from large CAD model repositories, Advanced Engineering Informatics 53 (2022) 101680. https://doi.org/10.1016/j.aei.2022.101680
12. M. Liu, X. Li, J. Li, Y. Liu, B. Zhou, J. Bao, A knowledge graph-based data representation approach for IIoT-enabled cognitive manufacturing, Advanced Engineering Informatics 51 (2022) 101515. https://doi.org/10.1016/j.aei.2021.101515
13. P. Wen, Y. Ma, R. Wang, Systematic knowledge modeling and extraction methods for manufacturing process planning based on knowledge graph, Advanced Engineering Informatics 58 (2023) 102172. https://doi.org/10.1016/j.aei.2023.102172
14. M. Shim, H. Choi, H. Koo, K. Um, K.H. Lee, S. Lee, OmEGa (O): Ontology-based information extraction framework for constructing task-centric knowledge graph from manufacturing documents with large language model, Advanced Engineering Informatics 64 (2025) 103001. https://doi.org/10.1016/j.aei.2024.103001
15. X. Liu, D. Jiang, B. Tao, F. Xiang, G. Jiang, Y. Sun, J. Kong, G. Li, A systematic review of digital twin about physical entities, virtual models, twin data, and applications, Advanced Engineering Informatics 55 (2023) 101876. https://doi.org/10.1016/j.aei.2023.101876
16. C. Latsou, D. Ariansyah, L. Salome, J.A. Erkoyuncu, J. Sibson, J. Dunville, A unified framework for digital twin development in manufacturing, Advanced Engineering Informatics 62 (2024) 102567. https://doi.org/10.1016/j.aei.2024.102567
17. Y. Wang, Y. Cheng, Q. Qi, F. Tao, IDS-KG: An industrial dataspace-based knowledge graph construction approach for smart maintenance, Journal of Industrial Information Integration 38 (2024) 100566. https://doi.org/10.1016/j.jii.2024.100566
18. Y. Ding, H. Li, F. Zhu, Z. Wang, W. Peng, M. Xie, A semi-supervised failure knowledge graph construction method for decision support in operations and maintenance, IEEE Transactions on Industrial Informatics 20 (3) (2024) 3104-3114. https://doi.org/10.1109/TII.2023.3299078
19. E. Elyan, L. Jamieson, A. Ali-Gombe, Deep learning for symbols detection and classification in engineering drawings, Neural Networks 129 (2020) 91-102. https://doi.org/10.1016/j.neunet.2020.05.025
20. S. Jiang, J. Hu, C.L. Magee, J. Luo, Deep learning for technical document classification, IEEE Transactions on Engineering Management 71 (2024) 1163-1179. https://doi.org/10.1109/TEM.2022.3152216
21. C. Zhang, G. Zhou, F. Chang, X. Yang, Learning domain ontologies from engineering documents for manufacturing knowledge reuse by a biologically inspired approach, The International Journal of Advanced Manufacturing Technology 106 (2020) 2535-2551. https://doi.org/10.1007/s00170-019-04772-1
22. D.B. Kim, M.S. Bajestani, J.Y. Lee, S.J. Shin, G.Y. Kim, S.M.M. Sajadieh, S. Noh, Human-in-the-loop in smart manufacturing (H-SM): A review and perspective, Journal of Manufacturing Systems 82 (2025) 178-199. https://doi.org/10.1016/j.jmsy.2025.05.020