Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 18, No. 2, 2026

Co-Construction of Production-Line AAS Geometric Models and Knowledge Graphs Driven by a Vision-Language Model

Hui Li *, Xiaoyi Zhao, Shasha Wang, Chenxi Yang, Wenxin Huang

School of Engineering, Zhengzhou Sias University, Zhengzhou, Henan, 451100, China

*** Corresponding author:** Hui Li

Abstract: Digital twin systems for a production line usually run on two models at once: an Asset Administration Shell (AAS) that holds asset information under IEC 63278, and a knowledge graph (KG) that records device topology and process semantics. On real projects, though, the two are built apart from each other. Engineers key the AAS geometric model in by hand from drawings and field measurements, while process engineers author the KG separately. The same data therefore gets entered twice, reuse is poor, and sooner or later the two models disagree. This paper presents a co-construction method that removes the duplication at its source. A fine-tuned vision-language model (VLM) reads the engineering sources a line already has, namely 2D/3D CAD drawings, on-site laser point clouds, and equipment technical manuals, and fills a standardized AAS; the KG is then derived from that AAS by fixed rules, so both models share one source. The method is arranged in four layers: multi-source data preprocessing with coarse-to-fine ICP spatial registration; LoRA-based domain adaptation of an open-source VLM supported by industrial prompt templates; automatic population of the AAS geometric, property, and connectivity submodels with conformance-checked .aasx generation on Eclipse BaSyx, plus QEM-based mesh simplification; and same-source KG derivation through fixed field-to-triple mapping rules, backed by a single-source constraint, automatic reconciliation, and incremental refresh with version traceability. We demonstrate the complete pipeline on a photovoltaic cell production line, and a parameterized Monte Carlo simulation of data consistency shows that, for manual single-entry error rates between 1% and 10%, conventional dual-entry modeling produces 4–39 times more data inconsistencies than the proposed single-source derivation, with the gap widening as the error rate grows. The method offers a reusable and extensible path toward production-line digital twin modeling; empirical validation on a physical line is planned as future work.

Keywords: Asset Administration Shell (AAS); Vision-language Model; Knowledge Graph; Digital Twin; Collaborative Modeling; Industry 4.0.

1. Introduction

Digital twins are no longer just a way to visualize a factory. On photovoltaic, automotive, and precision-equipment lines they now support virtual commissioning, daily operation and maintenance, and layout optimization [1]. The idea is easy to state: keep a faithful virtual counterpart of the physical line, and let condition monitoring, simulation-based prediction, and decision making run in a closed loop [2,3]. At the device level, the Asset Administration Shell (AAS) has become the standard container. IEC 63278 defines how the properties, behavior, and service interfaces of a physical asset are packed into submodels, which makes the model readable across systems and vendors [4]; for this reason the AAS is now a common carrier for asset-level digital twin models [5]. Knowledge graphs (KGs) cover the semantic side. They describe device topology and process logic as entity–relation–entity triples, they reason well over relations, and they are widely used for knowledge organization, question answering, and decision support [6,7].

In practice, though, the AAS geometric model and the production-line KG rarely grow together. They are built in two separate, largely manual phases: engineers transcribe AAS submodel fields by consulting CAD drawings and taking field measurements, and process engineers then author the KG, sorting out process sequences and device relations by hand. Working this way creates three problems that keep coming back. Modeling is expensive and slow, because both models feed on scarce engineering labor. Nothing gets reused, since the parameters and connectivity of one device must be keyed in twice, in two different tools and two different formats. And because the models are maintained apart, their versions drift over time, so conflicts in device parameters and connection data surface late and are painful to trace.

Vision-language models (VLMs) point to a way out. CLIP [8] and Segment Anything [9] have shown how much cross-modal understanding, open-vocabulary recognition, and semantic generation a single model can carry. Parsing the unstructured documents that fill engineering workflows, namely drawings, point clouds, and manuals, no longer has to be a manual job [10].

Our work started from the digital twin modeling needs of a photovoltaic (PV) cell manufacturing line. We propose a co-construction method that couples a domain-adapted VLM with the IEC 63278 AAS and KG technology. The pipeline reads "raw engineering data → VLM-based parsing → standardized AAS → same-source KG": both models are generated from one data source and can be updated in an incrementally synchronized way. The contributions are four.

(1) We design a four-layer co-construction architecture that links multi-source data input, VLM-based intelligent parsing, standardized AAS modeling, and same-source KG generation. Structured intermediate data interfaces sit between adjacent layers.

(2) We present a domain-adaptation pipeline for industrial VLMs. It combines LoRA fine-tuning with mechanical-

domain prompt templates; ICP-based spatial registration and QEM-based mesh simplification complete the picture. Together these pieces populate the AAS submodels automatically and produce conformance-checked .aasx files.

(3) We treat the AAS connectivity submodel as the single data source and specify a fixed field-to-triple mapping scheme, together with three consistency-assurance mechanisms: a single-source constraint, automatic post-generation reconciliation, and incremental refresh with version traceability.

(4) We demonstrate the whole pipeline end to end on a worked example. A parameterized Monte Carlo simulation indicates that single-source derivation cuts the item-level data inconsistency rate by a factor of 4.1–39.4 relative to conventional dual-entry modeling over realistic error-rate ranges.

The rest of the paper runs as follows. Section 2 reviews related work. Section 3 covers the preliminary technologies. Section 4 details the co-construction method. Section 5 analyzes it through a process comparison, a worked example, and a Monte Carlo simulation, and states the limitations and validation plan. Section 6 concludes the paper.

2. Related Work

Industry 4.0 and smart manufacturing programs have pushed digital twin modeling beyond geometric visualization, toward multi-source data integration, semantic understanding, and intelligent decision making [11,12]. Three research streams matter most for what follows: AAS-based standardized modeling, industrial knowledge graphs, and the industrial use of vision-language models.

2.1. AAS-based Modeling of Production Assets

The AAS is the core information model of the Industry 4.0 reference architecture. It wraps the metadata, submodels, and service interfaces of a physical asset into a standardized form and supports cross-enterprise, cross-system interoperability [4]. Plattform Industrie 4.0 keeps driving the specification forward, and recent studies have packaged AI components and equipment capabilities as AAS instances, validating them at the engineering level in realistic settings [13]; other efforts have examined systematically how OPC UA information models map onto the AAS [14]. What these efforts share is a focus on information modeling and communication design. The geometric submodel is still filled in by hand. In China, Zhou [15] built a flexible distributed motion control component on top of the AAS, encapsulating the states and services of IEC 61499 function blocks as AAS submodels; the result gives shop-floor device states a unified description and supports remote invocation from IT systems, which confirms that the AAS can carry real-time control. Even so, most existing AAS applications stop at “static encapsulation”, and efficient generation and dynamic updating of geometric submodels remain open problems.

2.2. Knowledge Graphs in Manufacturing

Knowledge graphs organize entities and their relations as triples, and manufacturing systems have used them for equipment management, process optimization, and intelligent scheduling [16,17]; comprehensive surveys summarize their application scenarios and maturity in production environments [18]. Cao et al. [19] tackled the “man–machine–material–method–environment” complexity of manufacturing workshops by building a Neo4j-based KG that combines top-down and bottom-up modeling of workshop resources with takt-time and energy models for production scheme optimization. The semantic representation of AAS content has been studied as well: AAS submodels have been mapped to semantic ontologies to support cross-system querying and reasoning [20]. In such approaches, however, the mapping relations are usually defined by hand, which caps the level of automation. There is another gap worth noting. Data in existing industrial KGs are collected independently of standardized asset models such as the AAS, so device parameters and connectivity get entered over and over, and conflicting information is easy to introduce.

2.3. Vision-language Models in Engineering Applications

General-purpose VLMs such as CLIP [8] and Flamingo handle cross-modal image–text understanding well, and Segment Anything [9] has pushed open-scenario image segmentation a long way forward. Engineering uses are appearing. Wang et al. [21] reconstructed 3D parametric models straight from 2D mechanical CAD raster drawings with a vision-language approach, showing that such models can recognize view contours, dimension annotations, and semantic information; multimodal large models have also been tried for the automatic evolution of knowledge graphs [10]. Most of these applications, however, assume well-structured, standardized drawings. How to adapt VLMs to non-standard engineering drawings and industrial equipment manuals is still underexplored. Most methods, moreover, extract a single modality only, and an end-to-end path that systematically fuses drawings, point clouds, and technical documents, from raw engineering data to standardized models and knowledge representation, has not yet been assembled.

To sum up: no integrated scheme yet lets a large model drive the collaborative, same-source generation of the AAS and the KG, and the geometric and attribute information of production assets still depends heavily on manual input. This paper attacks exactly that gap. We study the collaborative automated construction of AAS geometric models and KGs from unstructured engineering data.

3. Preliminaries

3.1. The AAS Architecture of IEC 63278

An AAS consists of an identification block, a set of submodels, and reference objects. The submodel is the unit that carries the geometric parameters, attribute information, and connection relations of a device [4]. Three submodel types concern us here. The geometric submodel stores overall dimensions and three-dimensional placement information. The property submodel records nameplate data such as the manufacturer, product designation, rated power, and weight. The connectivity submodel registers port identifiers, physical interconnections, and upstream/downstream process relations. Throughout the paper we treat the connectivity submodel as the authoritative source from which the KG is derived. The standardized exchange format of an AAS is the .aasx package file.

3.2. Vision-language Models

A VLM couples an image encoder with a text encoder and aligns the two modalities in a shared representation space, which enables image captioning, object detection, and text extraction under natural-language prompts [8]. Once it has

been adapted to industrial drawings and equipment manuals, it can recognize mechanical view contours, dimension annotations, and symbol codes, and pull out textual attributes such as model designations and rated parameters. That is precisely the engine the proposed method needs to turn unstructured engineering data into structured records.

3.3. Knowledge Graphs and QEM Mesh Simplification

A KG stores device topology and process relations as (head entity, relation, tail entity) triples; we use Neo4j for storage and visual querying [6]. On the geometry side, the quadric error metrics (QEM) algorithm assigns each vertex a quadratic error matrix and simplifies a mesh by iteratively contracting the edge with the smallest error, which keeps the geometric error introduced at each step minimal [22]. We use QEM to cut the face count while preserving critical features such as installation interfaces and kinematic structures, so the resulting mesh still meets the real-time rendering requirements of digital twin front ends.

4. The Proposed Co-construction Method

4.1. Overall Architecture

Fig. 1 shows the four-layer architecture: multi-source data input, VLM-based intelligent parsing, standardized AAS modeling, and same-source KG generation. The pipeline runs along the path "data fusion → intelligent extraction → model generation → relation derivation → consistency checking". The input layer takes in CAD drawings, on-site point clouds, and technical manuals. The parsing layer converts images and text into structured records. The AAS layer generates IEC 63278-compliant geometric, property, and connectivity submodels plus a lightweight 3D mesh, and the KG layer derives triples from the connectivity submodel and loads them into the graph database. Adjacent layers exchange data only through structured intermediate tables, never through direct coupling. The consistency-assurance mechanisms of Section 4.6 span the AAS and KG layers so that the two models stay same-sourced.

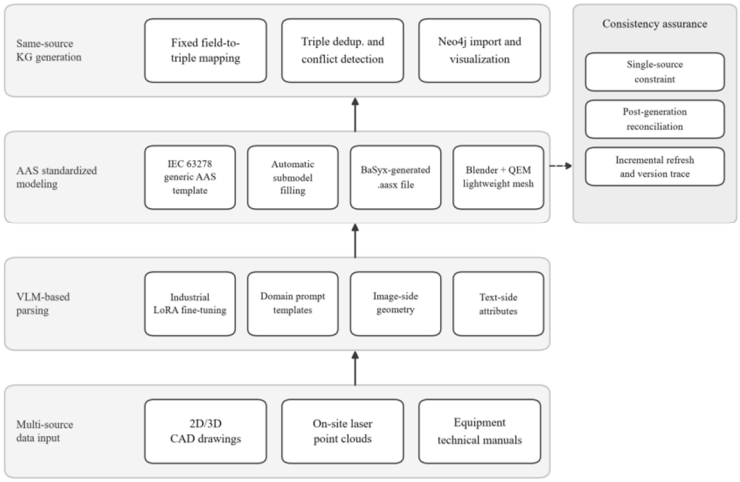


Fig 1. Overall architecture of the proposed co-construction method

4.2. Multi-source Data Preprocessing and Spatial Registration

(1) Ingestion and cleaning. Three classes of raw data come in: (i) 2D/3D CAD design drawings, which describe the geometry and installation dimensions of each device; (ii) laser point clouds of the shop floor, which reflect the as-built layout; and (iii) PDF/Word equipment technical manuals, which carry nameplate information such as power ratings, model designations, and interface parameters. We then apply format conversion, statistical and radius outlier removal for point cloud denoising, character correction, and layout normalization. These steps absorb the heterogeneous drawing formats, point cloud noise, and garbled document text, and leave a raw data set ready for parsing.

(2) Spatial registration. CAD drawings describe the design state; point clouds capture the as-built state. The two differ in coordinate frame and scale, so they must be registered before they can be fused. We adopt a coarse-to-fine strategy. Mounting datum points of the device give the initial pose, which normalizes the scale and roughly aligns the two coordinate frames at once. The iterative closest point (ICP) algorithm then refines the rigid transformation iteratively using feature point correspondences extracted from the point cloud and the CAD model; the iteration stops when the incremental transformation falls below a prescribed threshold ε. We quantify registration accuracy by the root-mean-square error (RMSE) over the correspondence set, and this figure doubles as the acceptance criterion for deciding whether the data of a given placement can move on to geometric modeling.

(3) Structured organization. Every parsing result lands in one unified structured data table. Its fields cover the device identifier, device name, model designation, overall dimensions, mounting coordinates, port identifiers, and upstream/downstream processes. This single table feeds both AAS submodel population and 3D geometric modeling.

4.3. Domain-adaptive VLM Fine-tuning and Information Extraction

(1) Sample construction. The fine-tuning corpus is drawn from the equipment drawings and process manuals of the PV line. We annotated view contours, dimension annotations, mounting points, model designations, rated parameters, and port types. Industrial samples are few and their class distribution is skewed, so we augment them with random

cropping, brightness and contrast perturbation, and layout composition, then split them 8:1:1 into training, validation, and test sets for fine-tuning and for evaluating recognition performance.

(2) Fine-tuning method and prompt templates. An open-source VLM serves as the backbone and is adapted with low-rank adaptation (LoRA) [23]. LoRA updates only a small number of low-rank matrices, which keeps the training cost acceptable for a small industrial corpus. In parallel we wrote prompt templates tailored to mechanical engineering documents; they constrain the output fields and formats so that model outputs can be written straight into the structured data table. How well the fine-tuned model recognizes content, and whether the adaptation helps, will be evaluated in the experiments planned in Section 5.5.

(3) Extraction workflow. The fine-tuned model performs two complementary tasks. On the image side it recognizes CAD view contours, dimension annotations, and mounting points, and outputs geometric coordinate information. On the text side it parses the tables and running text of the equipment manuals, extracting rated parameters, port types, and process predecessor/successor names. Both outputs are emitted as structured records under a predefined field template and written into the unified data table. Fig. 2 shows the pipeline.

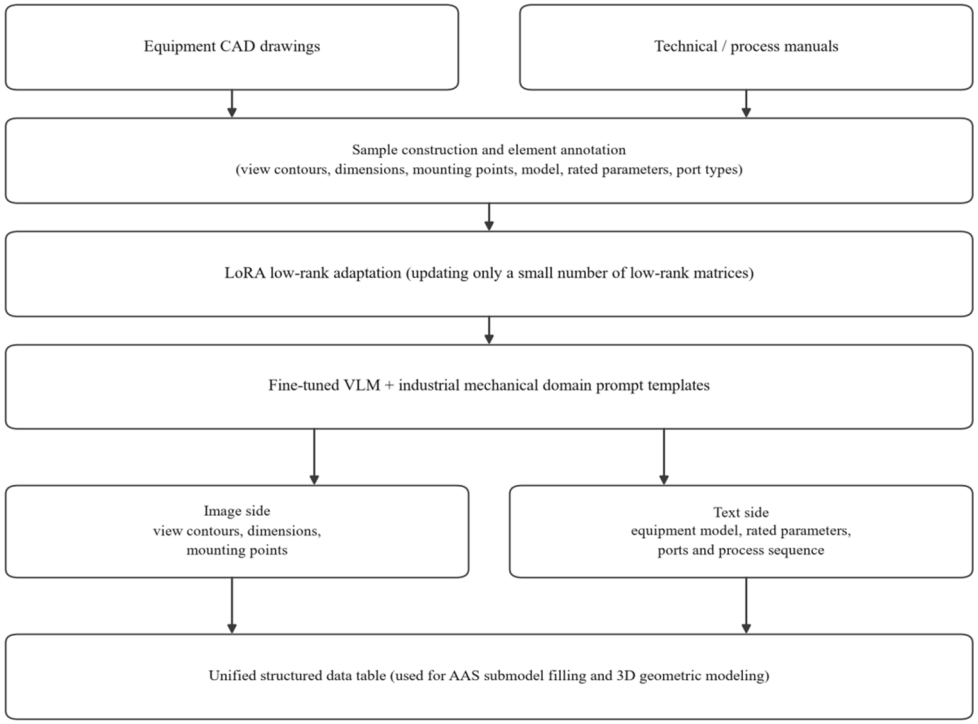


Fig 2. Pipeline of industrial domain adaptation and information extraction with the VLM

(4) Field correspondence. Table 1 defines how the multi-source data correspond to the AAS submodel fields. This correspondence is what makes automatic submodel population possible.

Table 1. Correspondence between multi-source data and AAS submodel fields.

Submodel	Field (idShort)	Data source	Extraction method
Property	ManufacturerName	Equipment technical manuals	VLM text extraction
Property	ManufacturerProductDesignation	Technical manuals, drawing title blocks	VLM text extraction
Property	RatedPower / Weight	Equipment technical manuals	VLM text extraction
Geometric	BoundingBox (length/width/height)	CAD dimension annotations	VLM image recognition
Geometric	InstallationPoint	Point cloud registration result and drawings	Spatial registration and coordinate transformation
Geometric	MeshFile (lightweight mesh reference)	CAD geometry and point cloud	Blender API and QEM
Connectivity	PortName	Drawing interface annotations	VLM image recognition
Connectivity	ConnectedTo	Drawing annotations and on-site verification	VLM recognition with rule-based checking
Connectivity	ProcessPredecessor / Successor	Equipment technical manuals	VLM text extraction

4.4. Automatic AAS Generation and Conformance Validation

(1) AAS template design. We designed a generic device-level AAS template for the PV line with reference to IEC 63278. It predefines the fields of the geometric, property, and connectivity submodels. Fig. 3 shows the template structure and the file generation workflow.

(2) .aasx file generation. Following the correspondence of Table 1, the structured extraction results are written automatically into the corresponding AAS submodel fields, and the Eclipse BaSyx open-source toolchain packages the result into a standard-compliant .aasx file.

(3) Geometric model generation and simplification. For each device, a 3D mesh is generated automatically with the Blender Python API from the CAD geometry and the registered point cloud, then simplified with the QEM algorithm. The simplification must preserve installation interfaces, kinematic structures, and other critical features, so that the mesh still meets the real-time rendering requirements of the digital twin front end. The simplified mesh is attached to the geometric submodel.

(4) Conformance validation. No .aasx file enters the model repository without passing a conformance check first. The check covers the integrity of the package structure (the references among the AAS environment, submodels, and resource files), the completeness of mandatory submodel elements (idShort, semanticId, kind, etc.), the conformity of identifiers and namespaces, and the cross-submodel reference consistency between the geometric and connectivity submodels. We use AASX Package Explorer as the validation tool. A file that fails is returned for correction with its validation log attached, and re-validated before acceptance.

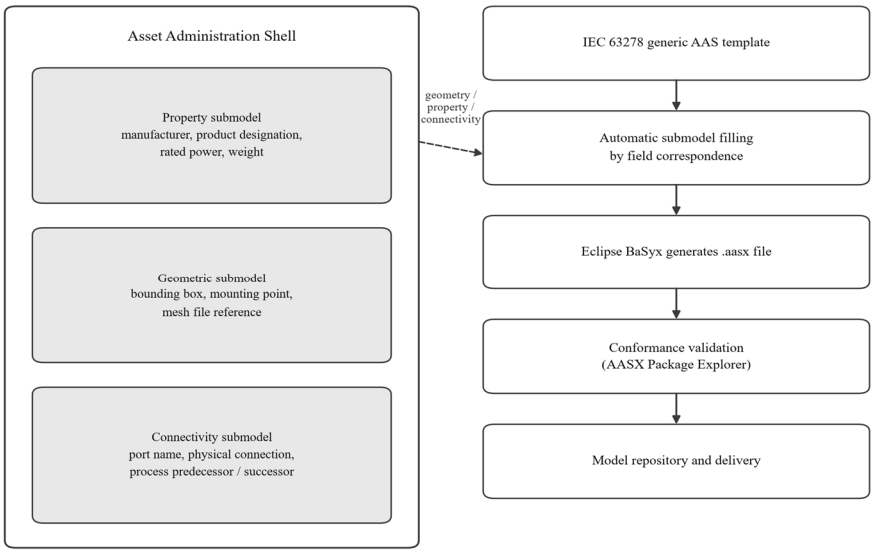


Fig 3. AAS submodel structure and the .aasx generation and validation workflow

4.5. AAS-driven Same-source KG Construction

A fixed set of field-to-triple rules, listed in Table 2, maps the port-level connections and process precedence relations stored in the AAS connectivity submodel onto KG triples. Device names and identifiers become entities. The physical-connection and process-precedence fields become relation predicates. Port identifiers ride along as relation attributes, which keeps multiple connections between the same pair of devices distinguishable.

Table 2. Field-to-triple mapping rules from the AAS connectivity submodel to the KG.

No.	AAS source field	Triple element	Generation rule
1	Device name (property submodel)	Head / tail entity	Device name as entity label; unique device code as entity ID
2	Model designation (property submodel)	Entity attribute	Written as the node attribute model in Neo4j
3	ConnectedTo (physical connection)	Predicate	Relation physicalConnection, directed from the source device to the target device
4	ProcessPredecessor	Predicate	Relation processPredecessor, directed from the upstream device to the downstream device
5	ProcessSuccessor	Predicate	Relation processSuccessor, inverse direction, for reverse queries
6	PortName	Relation attribute	Written as the port attribute to distinguish parallel connections between the same pair of devices

Mapping and loading run in three steps. A Python script parses the .aasx files in batch and locates the port and connection entries of the connectivity submodels. Triples are then generated according to the rules of Table 2. After deduplication and conflict detection, the triples are bulk-loaded into Neo4j, where the KG can be explored visually. Fig. 4 traces the mapping from the connectivity submodel to the resulting triples.

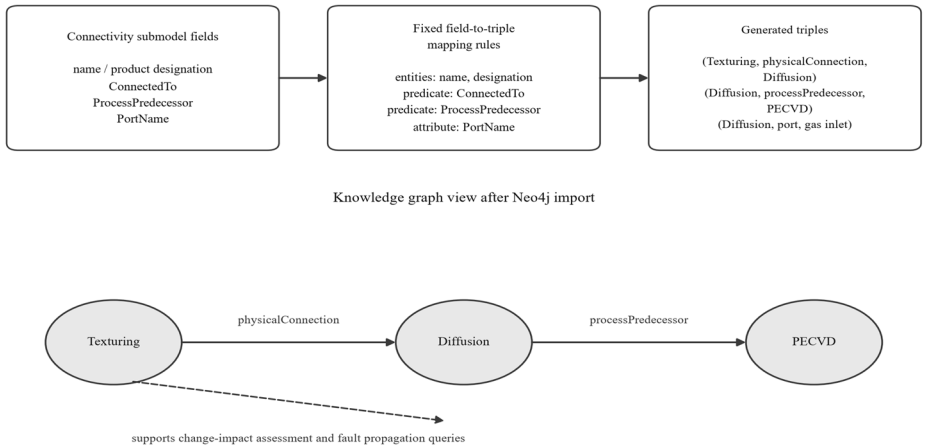


Fig 4. Mapping from AAS connectivity submodel fields to KG triples

4.6. Consistency Assurance Mechanisms

Three mechanisms keep the AAS and the KG consistent.

(1) Single-source constraint. Every KG entity and relation is derived from the AAS connectivity and property submodels. Direct data entry on the KG side is simply not allowed. Dual-entry divergence is thus eliminated by construction, not by checking.

(2) Automatic post-generation reconciliation. After each import, a reconciliation script reads the entities and relations back from Neo4j and compares them item by item with the AAS source fields; mismatches are collected into a list. The unique device identifier serves as the comparison key. The compared items include the device name, model designation, port identifier, connection relation, and process order.

(3) Incremental refresh with version traceability. When an AAS parameter changes, the script finds the affected triples by device identifier and refreshes them with a delete-then-insert step instead of a full rebuild. The AAS version number and the KG refresh timestamp are recorded, so every change can be traced.

5. Analysis and Discussion

We implemented the method as an end-to-end script chain over open-source tools. Quantitative validation on the physical line is still in progress, so this section analyzes the method at three complementary levels: a process-level comparison with conventional manual modeling (Section 5.1), a worked example that walks through the complete pipeline and its mapping rules (Sections 5.2 and 5.3), and a parameterized Monte Carlo simulation of the data consistency behavior (Section 5.3). Read these results as mechanism-level and simulation-level evidence of feasibility and benefit. They are not measured field performance; the protocol for empirical validation is given in Section 5.5.

5.1. Process-level Comparison with Conventional Manual Modeling

Table 3. Step-by-step comparison between conventional manual modeling and the proposed method.

Step	Conventional manual scheme	Proposed method	Underlying reason for the improvement
Data preparation and drawing parsing	Drawings and manuals consulted one by one; dimensions and designations transcribed by hand	VLM recognizes drawings and manuals in batch and emits structured records	Parameter entry shifts from manual lookup to model recognition, cutting repetitive labor
AAS submodel population	Field-by-field manual filling per template	Automatic filling from the structured data table per Table 1	Uniform, reusable filling rules suppress transcription errors
3D modeling and simplification	Manual modeling followed by experience-based mesh reduction	Automatic generation and QEM simplification via the Blender API	Geometry is driven by the registration result; simplification becomes controllable
Device connectivity modeling	Process engineers manually sort out and enter connection relations	Connections entered once in the AAS connectivity submodel	A single copy of the connectivity data is created and maintained
KG construction	Separate data collection and independent graph authoring	KG derived from the connectivity submodel per Table 2	Single-source derivation removes dual-entry divergence structurally
Model change and maintenance	Two models modified separately; versions easily diverge	Incremental refresh initiated from the AAS side synchronizes the KG	A single change entry point, combined with version traceability

Under the conventional scheme, the AAS geometric information and the production-line KG are built in two separate phases. Engineers transcribe AAS submodel fields from drawings and field measurements, while process

engineers author the KG and sort out process sequences and device relations by hand. The proposed method changes neither the modeling object nor the modeling result; what it changes is how each step is carried out. Table 3 compares the two schemes step by step.

Table 3 shows where the efficiency actually comes from. No single step gets dramatically shorter. What changes is the structure: parameter entry moves from "manual lookup and transcription" to "model recognition and rule-based filling"; connectivity data go from "entered twice" to "entered once and derived"; and maintaining the two models drops from two rounds to one. The port-identifier and process-order errors that duplicate entry breeds in the manual scheme then have no second entry point to arise from, and data consistency stops being an a posteriori checking problem and becomes a property guaranteed by mechanism. Section 5.3 quantifies these effects through a worked example and a parameterized simulation, and the protocol for empirical validation is given in Section 5.5.

5.2. A Worked Mapping Example

Take two adjacent processes on the PV cell line, texturing and diffusion, with device identifiers PV-A01 and PV-A02. Table 4 lists the relevant entries of their connectivity submodels and the triples generated under the rules of Table 2.

Table 4. Worked example of the mapping from connectivity submodel entries to KG triples (illustrative).

Connectivity submodel entry (illustrative)	Generated triple	Relation attribute
Port P1 of the texturing device (PV-A01) is physically connected to port P1 of the diffusion device (PV-A02)	(PV-A01, physicalConnection, PV-A02)	port = P1
The process successor of the texturing device is the diffusion device	(PV-A01, processSuccessor, PV-A02)	—
The process predecessor of the diffusion device is the texturing device	(PV-A02, processPredecessor, PV-A01)	—

Note: the device and port identifiers in the table are illustrative values, included only to explain the mapping rules

The derivation introduces no new semantic decisions. The head and tail entities, the predicate, and the attributes of every triple come straight from AAS fields, which is what makes scripted batch execution possible. When several connections link the same pair of devices, the port identifier is preserved as a relation attribute, so connection granularity survives triple deduplication. The rules are process-agnostic: adapting the method to another line means reconfiguring the sources of device names and process fields, not the rules themselves. That is the basis for the cross-line reusability of the method.

5.3. End-to-end Walkthrough and Consistency Simulation

Is the pipeline executable end to end, and what do the consistency mechanisms actually buy? This section answers those questions at two levels: (i) a complete walkthrough from raw engineering documents to AAS submodels and KG triples, using a diffusion device as the example, and (ii) a parameterized Monte Carlo simulation comparing the inconsistency risk of dual-entry modeling and single-source derivation. Both results come from a walkthrough and a simulation. They demonstrate feasibility and the order of magnitude of the benefit; they are not measured field performance, which will be validated under the protocol of Section 5.5.

(1) Worked walkthrough. Table 5 walks through one complete modeling cycle for the diffusion device PV-A02, following the workflow of Section 4.

Table 5. Worked walkthrough from raw engineering documents to the AAS model and the KG.

Step	Input	Processing	Output (illustrative)
Drawing parsing	General assembly drawing of the diffusion device (2D CAD)	Image recognition by the fine-tuned VLM	Overall dimensions 3,200 mm × 2,100 mm × 2,400 mm
Manual parsing	Technical manual of the diffusion device	Text extraction by the fine-tuned VLM	Model designation, rated power 85 kW, weight 4.2 t
Spatial registration	On-site laser point cloud of the installation position	Coarse alignment + ICP refinement	Mounting coordinates written to the geometric submodel
AAS field population	Structured data table	Automatic filling per Table 1	Fields of the property, geometric, and connectivity submodels
File generation	Generic AAS template + structured data	Packaging with Eclipse BaSyx	.aasx file of the diffusion device
Mesh simplification	CAD geometry and registered point cloud	Blender API + QEM	Lightweight mesh attached to the geometric submodel
KG derivation	Connectivity submodel entries	Rules of Table 2 (example in Table 4)	Triples such as (PV-A02, physicalConnection, PV-A03)

Note: all values in the table are illustrative figures that demonstrate the workflow; they are not measured data.

Three points emerge from the walkthrough. The output of each step is exactly the input of the next. The whole chain can be stitched together by scripts and open-source tools. And manual intervention is confined to two nodes, document preparation and result review. Information changes hands only through the structured data table, so there is no point anywhere in the chain where data must be re-transcribed by hand. That is the precondition the consistency mechanisms of

Section 4.6 rely on.

(2) Consistency simulation model. The simulated line matches the scale of the application scenario: N = 12 devices and M = 15 connections, each connection carrying 4 information items (port identifier, connected device, process predecessor, process successor). That gives M × 4 = 60 comparison items between the two models. The two modeling modes are modeled as follows.

(i) Dual-entry manual mode (conventional). Every item of every connection is entered once on the AAS side and once on the KG side. Each entry is erroneous with probability p, and the erroneous value is drawn at random. Whenever the two sides disagree, the item counts as inconsistent.

(ii) Single-source derivation mode (proposed). Here the KG is derived deterministically from the connectivity submodel, so the two sides agree by construction. The only possible source of inconsistency is an information item outside the coverage of the mapping rules, which occurs with probability q.

The simulation is a Monte Carlo experiment with 10,000 repetitions per parameter setting. It reports two rates: the item-level inconsistency rate (inconsistent items over total items) and the connection-level inconsistency rate (the share of connections containing at least one inconsistent item). Table 6 lists the parameter settings.

Table 6. Parameter settings of the consistency simulation.

Parameter	Value	Remark
Number of devices N	12	Matches the scale of the application line
Number of connections M	15	Mostly chained process connections with a few branches
Items per connection	4	Port ID, connected device, process predecessor, process successor
Total comparison items	60	M × 4
Manual single-entry error rate p	1%–10%, baseline 3%	Assumed; covers the range commonly reported for manual data entry
Mapping-rule non-coverage probability q	0.5%	The only inconsistency source of the proposed mode; the case q = 0 is included in the sensitivity analysis
Monte Carlo repetitions	10,000	—

(3) Results and sensitivity analysis. Table 7 compares the two modes; Fig. 5 plots the item-level inconsistency rate against p.

Table 7. Simulation results of dual-entry modeling versus single-source derivation.

Error rate p	Manual mode, item level / %	Manual mode, connection level / %	Proposed mode, item level / %	Ratio (item level)
1%	1.99	7.72	0.48	4.1
2%	3.99	15.04	0.48	8.3
3% (baseline)	5.92	21.66	0.48	12.3
5%	9.67	33.41	0.48	20.0
10%	19.01	57.08	0.48	39.4

Note: the proposed mode assumes q = 0.5%; with q = 0, both its item-level and connection-level inconsistency rates are zero.

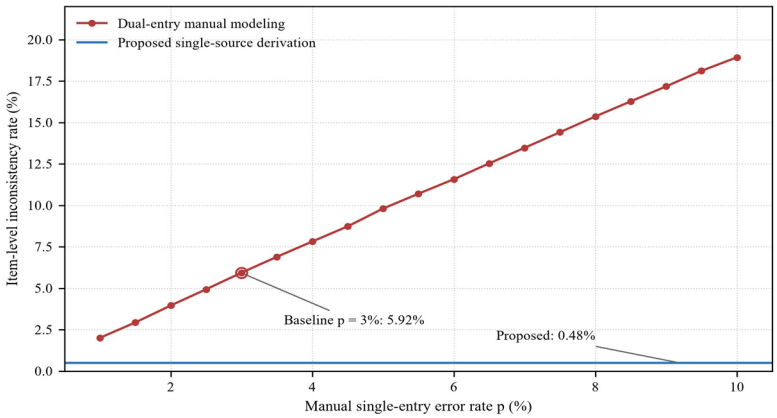


Fig 5. Sensitivity of the item-level inconsistency rate to the manual entry error rate p (Monte Carlo simulation, 10,000 repetitions)

Three observations follow. On the mechanism side, the probability that the two sides of an item disagree in the dual-entry mode is about 2p, and since a connection aggregates four items and goes inconsistent as soon as any one of them

disagrees, errors amplify level by level from items to connections: at the baseline p = 3%, the item-level rate is 5.92% while the connection-level rate already reaches 21.66%. On the sensitivity side, across the whole range p = 1%–10% the item-level inconsistency rate of the manual mode runs 4.1–39.4 times that of the proposed mode, and the gap widens with p, so the conclusion that single-source derivation beats dual entry does not hinge on any particular error-rate assumption. Finally, when q = 0 the inconsistency rate of the proposed mode falls to exactly zero. In the single-source mode, consistency is governed entirely by how completely the mapping rules cover the data. That is exactly why the reconciliation mechanism of Section 4.6 exists: the script verifies coverage after every derivation and confines the effect of q to a detectable, correctable range.

Workload shrinks too. The 15 connections comprise 60 items, which the dual-entry mode must enter 120 times; the proposed method needs only 60 entries on the AAS side, and the VLM fills those automatically. Manual entry effort for connectivity data is halved, and duplicated entries disappear altogether.

One caveat is in order. The simulation assumes independent errors and does not model systematic version desynchronization, such as updating only one side after a change. In engineering practice such errors push the real inconsistency rate above the simulated one, so the manual-mode results in Table 7 should be read as a lower bound of that scheme's inconsistency risk.

5.4. Applicability and Limitations

(1) Applicability. The method targets automated lines with relatively homogeneous equipment and reasonably complete basic documentation, especially lines that already hold 2D/3D CAD design data and line layout data. Legacy lines are a different story: where documentation is incomplete or exists only on paper, drawing digitization and parameter supplementation must come first, and that eats into the efficiency advantage.

(2) Limitations. (i) The recognition stability of the VLM depends on how representative the samples are; its performance on non-standard drawings, handwritten annotations, and low-resolution scans still needs to be examined on larger corpora. (ii) Point cloud registration leans on the mounting datum points for the initial pose; accuracy degrades when datum points are missing or severely occluded. (iii) The AAS conformance validation currently covers structural and naming rules at the tool level; semantic completeness (e.g., uniqueness of semantic identifiers and unit consistency) still requires manual review. (iv) The AAS template and the field-to-triple mapping rules are coupled to the process of the specific line; cross-industry deployment means reconfiguring both. (v) The project schedule confined this work to method design, pipeline implementation, and simulation; no quantitative comparison on the physical line has been run yet, so the efficiency and consistency benefits reported here are mechanism- and simulation-level findings awaiting empirical confirmation. (vi) The consistency simulation assumes independent errors and does not cover systematic version desynchronization; its results should be read as a lower bound of the risk of the manual scheme.

5.5. Validation Plan

The simulation of Section 5.3 gives the order of magnitude of the consistency benefit. Whether it holds on a real line is an empirical question, and it must be settled under the protocol below. Table 8 defines the evaluation metrics.

Table 8. Evaluation metrics for the planned validation.

Metric	Definition	Unit
Modeling duration	Total time from data preparation to repository acceptance of the AAS model and the KG	working day
Efficiency improvement	(manual duration − proposed duration) / manual duration × 100%	%
Data inconsistency rate	Inconsistent items / sampled items × 100%	%
Face reduction rate	(faces before simplification − faces after) / faces before × 100%	%
Critical structure deviation	Maximum dimensional deviation of critical mounting holes and interfaces before and after simplification	mm
Recognition precision	Correctly recognized elements / recognized elements × 100%	%
Recognition recall	Correctly recognized elements / ground-truth elements × 100%	%

The protocol runs as follows. (i) Modeling efficiency is measured in working days for the complete construction of the AAS models and the KG. Both the conventional and the proposed schemes are repeated several times and averaged, with the scope fixed as "from data preparation to repository acceptance"; the proposed scheme includes VLM fine-tuning and point cloud registration but excludes the manual annotation effort for the fine-tuning corpus. (ii) Data consistency is checked by sampling devices by identifier and comparing every AAS source field with the corresponding entity and relation in the KG. The sampling unit is the parameter item: one field of one device, or one connection, counts as one item. Two researchers adjudicate all detected inconsistencies independently, and a third reviewer resolves disagreements, which keeps the judging standard uniform. The measured rate is then compared with the simulated values of Table 7 to assess how well the simulation model predicts reality. (iii) Simplification quality is assessed jointly by the face reduction rate and the critical structure deviation. (iv) Extraction quality is assessed by precision and recall, with an ablation control in which the VLM runs without domain fine-tuning, so that the effect of the adaptation can be isolated.

6. Conclusion and Future Work

This paper tackled two deficiencies of production-line digital twin modeling: the heavy manual burden of AAS geometric modeling, and the data fragmentation between the AAS and the KG. Four contributions are worth restating. We designed a four-layer co-construction architecture, running from multi-source data input through VLM-based intelligent parsing and standardized AAS modeling to same-source KG generation, with structured interfaces between adjacent layers. We built an industrial domain-adaptation and extraction pipeline (LoRA fine-tuning supported by mechanical-domain prompt templates), combined it with the field correspondence of Table 1 to populate AAS submodels automatically and generate conformance-checked .aasx files, and used ICP spatial registration and QEM simplification to produce lightweight geometric models. Taking the AAS connectivity submodel as the single data source, we specified fixed field-to-triple mapping rules and three consistency-assurance mechanisms: a single-source constraint, automatic reconciliation, and incremental refresh with version traceability. Finally, we demonstrated the complete pipeline end to end on a worked example, and a parameterized Monte Carlo simulation showed that, for manual single-entry error rates between 1% and 10%, the dual-entry mode produces 4–39 times more data inconsistencies than the proposed single-source derivation, with the gap widening as the error rate grows.

The core move is to replace “two independently maintained models” with “one AAS from which the KG is derived”. That removes the structural cause of dual-entry divergence and halves the manual effort for connectivity data. The present work sits deliberately at the method-design and simulation level. No quantitative field trial has been run, and whether the simulation conclusions transfer to a physical line remains to be verified.

Four lines of future work are planned: (i) an empirical study on the PV cell line following the metrics and protocol of Section 5.5, in which the measured inconsistency rate will calibrate the simulation model and drive iterative refinement of the AAS templates and mapping rules; (ii) extending the fine-tuning corpus with equipment from other industries, to obtain a more generally applicable VLM; (iii) adding semantic reasoning on top of the graph, to support fault propagation analysis and change-impact assessment; and (iv) connecting the models to MES/SCADA real-time data for dynamic and continuous updating.

Declaration of Competing Interest

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

Acknowledgments

This work was supported by the General Research Program of Zhengzhou Sias University [grant number 2025XKD090], project "Research on the Co-construction of Production-Line AAS Geometric Models and Knowledge Graphs Based on Large Models".

References

[1] Tao, F., Zhang, H., Liu, A., & Nee, A. Y. C. (2019). Digital twin in industry: State-of-the-art. IEEE Transactions on Industrial Informatics, 15(4), 2405–2415. https://doi.org/10.1109/TII.2018.2873186.

[2] Tao, F., Liu, W. R., Zhang, M., Hu, T. L., Qi, Q. L., Zhang, H., Sui, F. Y., Wang, T., Xu, H., Huang, Z. G., Ma, X., Zhang, L. C., Cheng, J. F., Yao, N. K., Yi, W. M., Zhu, K. Z., Zhang, X. S., Meng, F. J., Jin, X. H., Li, Y., Lv, Q., & Luo, Y. M. (2019). Five-dimension digital twin model and its ten applications. Computer Integrated Manufacturing Systems, 25(1), 1–18. [In Chinese]. https://doi.org/10.13196/j.cims.2019.01.001.

[3] Tao, F., Cheng, Y., Cheng, J. F., Zhang, M., Qi, Q. L., & Zhang, H. (2017). Theory and technology of cyber-physical fusion in digital twin shop floors. Computer Integrated Manufacturing Systems, 23(8), 1603–1611. [In Chinese].

[4] International Electrotechnical Commission. (2023). IEC 63278-1:2023. Asset administration shell for industrial applications—Part 1: Asset administration shell structure. IEC.

[5] Tao, J., Dai, Y. C., & Wei, R. (2017). Research on production life cycle based on digital thread and digital twin. Aeronautical Manufacturing Technology, 60(21), 26–31. [In Chinese].

[6] Hogan, A., Blomqvist, E., Cochez, M., d’Amato, C., de Melo, G., Rigotti, C., Saleem, M., & Vrandecic, D. (2021). Knowledge graphs. ACM Computing Surveys, 54(4), Article 71, 1–37. https://doi.org/10.1145/3447772.

[7] Liu, X. B., Wang, S. T., & Bai, C. Y. (2022). Visualization analysis of digital twin development based on a scientometric knowledge map. Computer Integrated Manufacturing Systems, 28(6), 1673–1684. [In Chinese].

[8] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In Proceedings of the 38th International Conference on Machine Learning (pp. 8748–8763). PMLR. https://doi.org/10.48550/arXiv.2103.00020.

[9] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W. Y., Dollár, P., & Girshick, R. (2023). Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 4015–4026). IEEE. https://doi.org/10.1109/ICCV51070.2023.00371.

[10] Wang, S. B. (2025). Research on knowledge graph evolution based on multimodal large models. China High-Tech, (2), 25–27. [In Chinese].

[11] Liu, Q. W., & Yang, G. (2025). Application research and exploration of digital twins in the machinery field under Industry 4.0. Intelligent Manufacturing, 32(3), 95–100. [In Chinese].

[12] Guo, L., Zhang, H. Q., Cheng, W. S., & Liu, Y. (2023). Research status and prospects of data-driven digital twin technology. Machinery, 50(7), 1–10. [In Chinese].

[13] Rauh, L., Reichardt, M., & Schotten, H. D. (2022). AI asset management: A case study with the asset administration shell (AAS). In 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation. IEEE. https://doi.org/10.1109/ETFA52439.2022.9921705.

[14] Cavalieri, S., & Salafia, M. G. (2020). Insights into mapping solutions based on OPC UA information model applied to the Industry 4.0 asset administration shell. Computers, 9(2), 28. https://doi.org/10.3390/computers9020028.

[15] Zhou, H. T. (2024). Design and implementation of a flexible distributed motion control component based on AAS [Master’s thesis]. Huazhong University of Science and Technology. [In Chinese].

[16] Tao, F., Liu, W. R., Liu, J. H., Liu, X. J., Liu, Q., Qu, T., Hu, T. L., Zhang, Z. N., Xiang, F., Xu, W., & Zhang, H. (2018).

Digital twin and its application exploration. Computer Integrated Manufacturing Systems, 24(1), 1–18. [In Chinese].

[17] Banerjee, A., Dalal, R., Mittal, S., & Joshi, S. (2017). Generating digital twin models using knowledge graphs for industrial production lines. In Workshop on Industrial Knowledge Graphs, co-located with the 9th ACM Web Science Conference.

[18] Buchgeher, G., Gabauer, D., Martinez-Gil, J., & Bergner, K. (2021). Knowledge graphs in manufacturing and production: A systematic literature review. IEEE Access, 9, 55537–55554. https://doi.org/10.1109/ACCESS.2021.3070395.

[19] Cao, Y. Y., Xu, Z., Tian, Z. G., & Zhang, L. (2024). Research on a digital twin knowledge graph for manufacturing workshops. Mechanical Science and Technology for Aerospace Engineering, 43(12), 2089–2095. [In Chinese]. https://doi.org/10.13433/j.cnki.1003-8728.20230128.

[20] Elshafei, B., Martínez-Arellano, G., Chaplin, J. C., & Ratchev, S. (2024). Semantic knowledge representation in asset administration shells: Empowering manufacturing utilization. In Flexible Automation and Intelligent Manufacturing (FAIM 2024) (pp. 282–296). Springer. https://doi.org/10.1007/978-3-031-74482-2_32.

[21] Wang, X., Zheng, J., Hu, Y., Li, Y., & Zhang, H. (2025). From 2D CAD drawings to 3D parametric models: A vision-language approach. Proceedings of the AAAI Conference on Artificial Intelligence, 39(8), 7961–7969. https://doi.org/10.1609/aaai.v39i8.32858.

[22] Garland, M., & Heckbert, P. S. (1997). Surface simplification using quadric error metrics. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (pp. 209–216).

[23] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. In Proceedings of the 10th International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2106.09685.