NexusEngine: Ontology-Constrained, Provenance-Grounded, Multi-Modal Generation for Infrastructure Digital Twins

Kanimozhi Uma
SE-Link BV
kani@se-link.be

Merijn Vanalderweireldt
SE-Link BV
merijn@se-link.be

Abstract

Turning heterogeneous engineering documents into a usable infrastructure digital twin is, at its core, a *multi-modal, ontology-constrained* natural language generation problem-not the OCR/information-extraction task it is usually treated as-with two coupled targets: strict, ontology-typed structured text (a knowledge graph) and grounded free-text reports generated from it. Both must be near-perfectly faithful, operating under a published Object Type Library (OTL) within safety-critical infrastructure. We present NEXUSENGINE, which recasts document-to-twin conversion as ontology-constrained, provenance-grounded generation. Three ideas drive it: (i) visual-first conditioning that generates from layout rather than lossy text; (ii) schema-constrained multi-persona generation under a hard response schema; and (iii) a deterministic faithfulness stack (sanitiser, validator, fusion) with a bounded verify-and-repair loop, a bitemporal store, and a validation layer that ensures accepted deliveries. We report entity/relation F1 of 0.91/0.86 and 0.99 OTL-accuracy, a human evaluation in which experts rate grounded reports 4.63/5 for faithfulness versus 3.08 for an ungrounded baseline, and an un-localisable-span rate cut from 9.4% to 1.3%. Deployed on a large infrastructure development project, the structured generator (T1) processes 41,382 pages at 4.6 expert-minutes per 100 pages-a 326× leverage; the report generator (T2) is a research prototype, evaluated here in a blinded human study.

1 Introduction

Capital-intensive infrastructure projects emit vast, heterogeneous documentation-piping-and-instrumentation diagrams (P&IDs), single-line electrical schematics, BIM/CAD exports, equipment lists, and free-text specifications. Public-infrastructure operators increasingly must re-express this documentation as a *digital twin*: a queryable knowledge graph in which every physical asset, such as a *Stuurkast* (control cabinet) identified in the engineering drawing (e.g., `Stuurkast VF06`, `Stuurkast VF07-W`, or `Stuurkast VF07-O`), is represented as a typed node in the knowledge graph. Its characteristics and identifiers are represented as attributes, while its functional, connectivity, or structural relationships to other assets are represented as typed edges. The resulting graph must conform to the mandated domain ontology, the *Object Type Library* (OTL)[1], which fixes the allowed classes, attributes, relationship types, hierarchical identity, and structural constraints of each physical asset.

Today this conversion is done by engineers reading page by page: at roughly 15 minutes per page it is slow, costly, and inconsistent between readers. Yet the twin-and the reports written from it-are exactly what human asset managers, safety coordinators, and maintenance planners, together with downstream asset-management software, rely on to plan inspections, route repairs, and certify safety. An invented voltage or a mis-linked cabinet can misdirect maintenance or mask a safety-relevant fact, so the conversion must be *near-perfectly faithful*. The binding tension is to automate 40,000-page workload while meeting a standard of correctness.

Infrastructure projects rarely store their engineering knowledge in a single representation. A project may combine structured engineering sources-native CAD drawings (DWG/DXF) and structured exchange exports such as LandXML/IFC-with dense geospatial and vector documentation distributed across multi-page PDF files. The structured sources

[1] In Flanders the relevant standard is the public OTL of the Roads and Traffic Agency (AWV), derived from the OSLO[2] Linked Data framework (Agentschap Wegen en Verkeer, 2021); comparable libraries are governed across Europe via CEDR INTERLINK/CoDEC (CEDR Transnational Road Research Programme, 2024) and platforms such as Laces (Semmtech, 2024).

Proceedings of the 19th International Natural Language Generation Conference: Long Papers, pages 992–1005
October 17-21, 2026 ©2026 Association for Computational Linguistics

provide machine-readable engineering structure, including alignments, profiles, surfaces, roadway components, and networked infrastructure, whereas geospatial and vector PDFs provide spatial layouts, equipment lists, annotations, and other visual engineering evidence. These modalities are complementary rather than redundant: the former provides explicit structural semantics, while the latter preserves spatial and visual evidence that can be lost through linear text extraction. NEXUSENGINE therefore treats heterogeneous engineering documentation as multimodal evidence that must be jointly transformed into an ontology-aligned representation. Large Flemish infrastructure programmes such as the Oosterweelverbinding (Lantis, 2024) exemplify the scale and heterogeneity of this problem and mandate the MOW OTL as the delivery standard (Agentschap Wegen en Verkeer, 2024; Lantis and Agentschap Wegen en Verkeer, 2023).

Why this is a generation problem, not extraction. The operator does not consume spans of source text; that consumes two *generated* artifacts: (a) a strict, ontology-typed *structured text*-the knowledge graph, serialised as typed JSON/RDF-and (b) human-readable condition reports and RFIs written from that graph. Both are produced by a language model that must be *controllable* (the output has to satisfy a hard schema) and *faithful* (every value and sentence must trace to the source). Framing the task as controllable, grounded generation lets us bring the right NLG tools to bear: controlled/constrained generation (Lin et al., 2020; Geng et al., 2023; Willard and Louf, 2023), data-to-text and KG-to-text generation (Gardent et al., 2017; Ribeiro et al., 2021; Reiter and Dale, 2000), faithfulness and hallucination analysis (Maynez et al., 2020; Ji et al., 2023), and two decades of NLG human-evaluation methodology (Howcroft et al., 2020; van der Lee et al., 2021; Dušek et al., 2020).

What NEXUSENGINE does. Given a set of engineering documents, it (i) renders each to an image; (ii) generates an OTL-typed graph fragment per chunk under a hard response schema; (iii) deterministically validates and *repairs* each fragment against the ontology; (iv) fuses fragments-with a human in the loop for cross-file duplicates-into one provenance-grounded knowledge graph; and (v) generates condition reports and RFIs from that graph, each sentence cited to its source. Figure 1 walks one real document through the whole pipeline.

Pure-text generative IE (Cabot and Navigli, 2021; Josifoski et al., 2022) fails here because the *layout* of an engineering document *is* part of its meaning. Layout-aware document models (Huang et al., 2022; Kim et al., 2022; Wang et al., 2024) recover some layout but require per-class fine-tuning and emit free-form structures that fail OTL validation by construction. Generic LLM-based KG generators (Edge et al., 2024; Pan et al., 2024; Mihindukulasooriya et al., 2023) likewise produce schemas of their own invention. We therefore frame the problem as **ontology-constrained, provenance-grounded generation**, and build around three design decisions.

1. **Visual-first conditioning.** Every input-including office formats-is rendered to a high-resolution image so a single vision-language backbone conditions on layout uniformly rather than on lossily-linearised text.

2. **Schema-constrained multi-persona generation.** One LLM call follows a CoT/ReAct protocol (Wei et al., 2022; Yao et al., 2023) that instantiates four specialist personas; the vendor `response_schema` enforces typed JSON, with a deterministic recovery fallback for the residual 4.7% of malformed completions.

3. **Deterministic, symbolic post-processing + a grounded-report prototype.** A pure-functional sanitiser/validator/fusion stack enforces OTL conventions and resolves duplicates (auto-correcting and flagging, never re-prompting); cross-file ambiguities route to a human-in-the-loop fusion UI; and a *prototype* Ontology-Augmented Generation (OAG) layer renders reports and RFIs that cite provenance. A closed verify-and-repair loop and a bitemporal store are implemented in the released engine (§4.6); preference-based refinement (Ouyang et al., 2022; Rafailov et al., 2023) of a distilled generator is future work.

Core thesis and contributions. Document-to-twin conversion is casted as *multi-modal, ontology-constrained, dual-target* text generation for safety-critical infrastructure. We develop it along three axes of novelty.

Originality (theoretical & architectural). (1) We recast a task conventionally handled as OCR / information extraction as multi-modal, ontology-constrained *text generation*, realised as a *dual-*

target synthesis: the system produces two tightly-coupled, heterogeneous outputs at once-deterministic, schema-valid structured text (an OTL knowledge graph) and fluent, provenance-grounded natural language (condition reports / RFIs)-under *visual-first conditioning* that bypasses lossy text extraction and generates directly from spatial layout and visual schemas (§4).

Impact (domain & operational). (2) We show a constrained-generation system operating safely in a safety-critical engineering domain where text hallucination is catastrophic, deployed on a live large infrastructure development project at a scale of 41,382 pages and cutting document-to-twin conversion to 4.6 expert-minutes per 100 pages-a 326× leverage over manual translation (§7).

Contribution to NLG. (3) a *deterministic faithfulness architecture* (sanitiser, validator, fusion) that forces loose LLM output into strict compliance with a published Object Type Library; **(4)** the *un-localisable-span rate* as a practical grounding / hallucination-audit metric, which our grounding pass reduces from 9.4% to 1.3%, together with a nuanced per-modality, ablation and error analysis (§6); and **(5)** a clean human-evaluation methodology by which domain experts reliably grade the factual precision of constrained generation against ungrounded baselines (§5–6.3). We release an open, agentic reference implementation with a live validated-delivery path (§4.6, App. A).

2 Related Work

Data-to-text, KG-to-text, and controlled generation. Verbalising structured data is a founding NLG problem (Reiter and Dale, 2000); WebNLG (Gardent et al., 2017) and pretrained graph-to-text models (Ribeiro et al., 2021) target KG→text. Our report generator is such a verbaliser, but constrained to cite provenance per clause and evaluated for faithfulness rather than BLEU-the binding constraint in safety-critical NLG, where fluent-but-unfaithful output is worse than none (Maynez et al., 2020; Ji et al., 2023). Unlike constrained decoding (Lin et al., 2020; Geng et al., 2023; Willard and Louf, 2023), we push faithfulness enforcement *out* of the decoder into a deterministic, auditable validator.

Ontology-grounded generation and document AI. OG-RAG (Sharma et al., 2024), GraphRAG (Edge et al., 2024), the LLM+KG roadmap (Pan et al., 2024) and Text2KGBench (Mihindukulasooriya et al., 2023) ground generation in a graph; we differ in that the ontology is a regulated, externally-versioned standard and "compliance" is a hard gate, not a soft prior. Document models (Huang et al., 2022; Kim et al., 2022; Wang et al., 2024) and transformer image-to-graph extractors (Shit et al., 2022; Stürmer et al., 2024), unblocked by datasets like Enginuity (Seefried et al., 2026), need task-specific fine-tuning; we instead treat a frontier vision-LLM as a black-box joint generator and enforce the OTL post hoc.

Multi-agent generation, evaluation, and benchmarks. Graph-routed orchestration (LangGraph (LangChain Inc., 2024), AutoGen (Wu et al., 2023)) is the surviving paradigm once unstructured peer-agent collaboration cascades, with the KG as shared state (Liu et al., 2025); we push this to *single-call* persona switching for cost. Following NLG-evaluation guidance (Howcroft et al., 2020; van der Lee et al., 2021; Dušek et al., 2020; Celikyilmaz et al., 2020) we pre-register rating dimensions, report agreement, and pair human judgement with automatic faithfulness proxies. Finally, DABstep (Egg et al., 2025) exposed single-prompt fragility on heterogeneous-file reasoning (∼16% on Hard), which we use as a transfer probe in §5.

3 Task Formulation

Engineering input modalities. The source set $\mathcal{D}$ contains heterogeneous engineering representations with complementary information content. We distinguish two representative modalities. **Structured engineering data**-native CAD (DWG/DXF) and exchange formats such as LandXML/IFC-encodes machine-readable geometric and structural information such as alignments, profiles, surfaces, roadway components, and networked infrastructure. **Geospatial and vector documentation**, primarily distributed as high-resolution multi-page PDFs, encodes spatial layouts, equipment lists, annotations, symbols, and other visual engineering evidence. The two modalities are not treated as interchangeable representations: structured sources provide explicit machine-readable relations and geometry, while visual documents preserve spatial and graphical evidence that may not survive text linearisation. NEXUSENGINE therefore maps both into the same ontology-constrained intermediate representation while retaining modality-specific provenance.

Let $\mathcal{D} = \{d_1, \ldots, d_n\}$ be source documents and

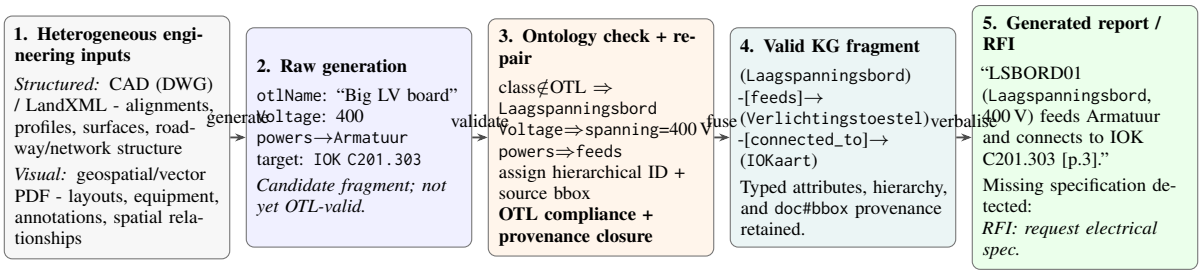


Figure 1: **End-to-end ontology-constrained generation and validation.** The input set (1) is heterogeneous-structured LandXML and visual/geospatial PDFs; here a PDF electrical-schema region (`OWVA-C201-BAM-TEK-. . . -GAB_Ver2.pdf`) is used to generate a candidate asset–relationship fragment (2). The raw generation is deliberately allowed to contain non-standard class, attribute, and relationship labels. Deterministic ontology checking and repair (3) map the candidate to valid OTL classes, typed properties, relationship semantics, hierarchical identifiers, and source-pixel provenance. The repaired fragment forms a provenance-closed knowledge graph (4), from which grounded technical reports and gap-analysis RFIs can be generated (5). The example therefore demonstrates that ontology constraints govern not only naming, but also *class typing, attribute semantics, relationship semantics, hierarchy, and provenance grounding*.

$\mathcal{O} = (\mathcal{C}, \mathcal{A}, \mathcal{R})$ a target ontology of classes, allowed per-class attribute keys, and allowed relationship types, published as RDF/OWL (e.g. class *Laagspanningsbord*; data property *heeftVermogen*; typed edge *isOnderdeelVan*). The OTL functions less as a target database than as a *semantic constraint layer*: it fixes which classes exist, which attributes and relationships are admissible for each, and their structural constraints, and therefore governs what the generator may legitimately assert about a source object. We define two coupled generation tasks.

T1 - Constrained structure generation. Generate $G = (V, E)$ where each node $v = \langle id, c, a, p, h \rangle$ carries class $c \in \mathcal{C}$, an attribute map a over $\mathcal{A}(c)$, a provenance set p, and a hierarchical id h inherited along `part_of` chains. A node/edge is *valid* iff it satisfies four properties: schema validity, naming conventions (PascalCase classes, camelCase attribute keys), hierarchical consistency, and *provenance closure* (at least one ⟨source, bbox⟩ entry). We define **OTL-Acc** as the fraction of generated nodes passing all four without human edit-the precise operational meaning of our automation-rate claim.

T2 - Grounded report generation. Given a subgraph $G_q \subseteq G$ and an intent q (condition report, compliance check, or RFI), generate free text y such that every claim in y is *entailed by and cited to* G_q and its underlying provenance. We measure y by faithfulness (automatic and human), adequacy, fluency, and actionability (§5). T2 is the operator-facing payload; T1 is its precondition.

4 System Design

NEXUSENGINE is a seven-stage, browser-side pipeline (Figure 2) that couples a single constrained-generation call per chunk with a deterministic, symbolic post-processor. **(1) Ingestion** renders every input to an image (§4.1); **(2) Extraction** is the sole generative LLM call per chunk-Gemini 2.5 Pro[2]-emitting OTL-typed JSON under a strict response schema (§4.2); **(3) Normalisation** and **(4) Validation** are pure functions that sanitise, auto-correct, and flag the generated assets against the ontology (§4.3); **(5) Entity Resolution** fuses within-file duplicates (Algorithm 1); **(6) Graph Construction** routes cross-file duplicates to a human-in-the-loop UI (§4.5); and **(7) Export** emits the graph in seven formats. Chunks of a file are generated in parallel and then fused. We release a Python reference implementation in which the orchestration is a fixed design-time *agent graph*: every agent declares, in a machine-checkable form, its role, the tools it may call, the post-condition its output must satisfy, and a bounded autonomy budget, and is *routed*, never invoked ad hoc, which is what averts the cascading failures of unstructured multi-agent collaboration (Wu et al., 2023; Liu et al., 2025; LangChain Inc., 2024). On this backbone the engine adds the four production components the first deployment lacked-ontology-retrieval conditioning, a bounded verify-and-repair loop, a bitemporal store, and a validated-delivery path (§4.6)-and its deterministic core is verified by an offline test suite (App. A).

[2]Model version pinned at deployment; a silent version update produced a 0.03 F1 regression-see Limitation (4).

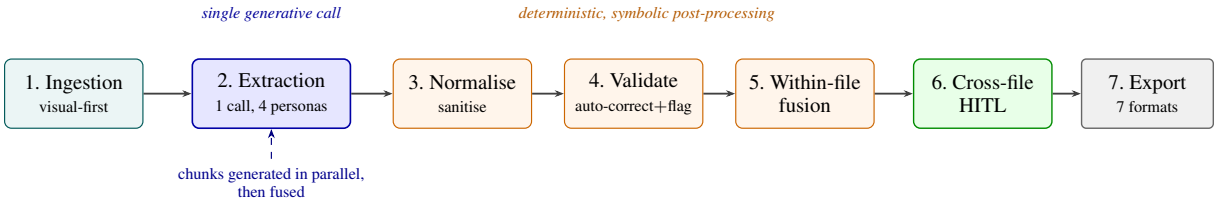


Figure 2: The deployed NEXUSENGINE pipeline. A single constrained generation call per chunk (Gemini 2.5 Pro; four CoT/ReAct personas) is followed by a purely deterministic, symbolic post-processor: sanitisation, OTL validation with auto-correction and issue-flagging (non-blocking-it never re-prompts the model), within-file fusion (Algorithm 1), human-in-the-loop cross-file fusion (`FusionView`), and export to seven formats. Chunks of a file are generated in parallel and fused. Stage numbers and component names match the deployed NEXUSENGINE application; the released reference implementation implements these stages plus retrieval conditioning, a verify-and-repair loop, a bitemporal store and validated delivery (§4.6).

4.1 Visual-First Conditioning (Stage 1)

Stage 1 preserves the distinction between structured and visual engineering inputs while providing a common conditioning interface to the generator. Geospatial and vector PDFs are split per page (`pdf-lib`) and supplied as high-resolution visual evidence, preserving spatial layout, symbols, annotations, and equipment-to-location relationships. DOCX is converted to HTML (`mammoth`); XLSX to a styled HTML table (SheetJS). Both are rendered into a 1200 px off-screen DOM container and captured at $2\times$ device-pixel ratio (`html2canvas`) into a high-resolution PNG, preserving cell borders, colour fills, and relative column widths that plain-text extraction discards. Images exceeding a 2048 px maximum edge are tiled into a 3×3 grid of 40%-size tiles stepped by 30% of each dimension, giving a fixed $\rho{=}10\%$ spatial overlap so that a component straddling a tile boundary stays visible in an adjacent tile. XML, IFC, and LandXML remain in their structured representation and are chunked at 500k characters on newline boundaries rather than being linearised into images. This modality-aware ingestion preserves the complementary structural and visual evidence required by downstream ontology-constrained generation.

4.2 Schema-Constrained Multi-Persona Generation (Stage 2)

Each chunk is sent as an inline-base64 payload under a strict response schema that encodes $\mathcal{O}$ as a typed JSON array: every asset carries composite identifiers (external, hierarchical, and unique), a PascalCase type, camelCase provenance-traced attributes, typed relationships, and $\langle$source, type, confidence, bbox$\rangle$ provenance. A five-phase chain-of-thought prompt instantiates four sequential personas in a single call-TOPOLOGY (visual hierarchy), TABULAR (attributes), CIRCUIT (distribution boards) and CIVIL3D/BIM (CAD-to-OTL geometry)-which triage modality, link cross-modal keys, generate, fuse, and assign hierarchical ids along the containment chain. The CIVIL3D/BIM persona is also responsible for structured engineering representations such as LandXML, while TOPOLOGY and the visual reasoning stages operate primarily over diagrammatic and geospatial evidence; cross-modal identifiers and spatial cues are therefore resolved before the candidate fragment enters deterministic OTL validation. The single call captures 92% of multi-call F1 at 33% of token cost (§6). A three-rung recovery ladder (Geng et al., 2023; Willard and Louf, 2023) handles the residual 4.7% of malformed completions by re-parsing progressively more permissively; anything recovered is flagged for the reviewer.

4.3 Deterministic Faithfulness Stack (Stages 3–5)

Normalisation (Stage 3). A deterministic normaliser applies three idempotent actions to every generated asset: field enforcement (surrogate ids/labels for missing keys, each flagged); id generation (random base-36 suffixes for session-global uniqueness); and warning propagation (recovered assets carry an explicit "data recovered from a partial or malformed response" note, so they are never silently promoted to authoritative status).

Validation (Stage 4). A deterministic validator enforces $\mathcal{O}$ in five rules: **R1** naming (class names to PascalCase, attribute keys to camelCase); **R2** duplicate hierarchical-id flagging; **R3** relationship integrity-dangling targets flagged and Dutch relationship types normalised to a canon-

ical English set via a curated synonym map (*onderdeel_van*→part_of, *voedt*→feeds, *verbonden_met*→connected_to, *bewaakt*→monitors, *stuurt*→controls); **R4** numeric-attribute type checks (*vermogen*, *spanning*, *lengte*); and **R5** hierarchical consistency (a child id must dot-extend its parent).

Validation is non-blocking. Validation runs *once* per file. It auto-corrects what it can do safely (PascalCase/camelCase casing, the synonym mapping) and appends every remaining issue to the asset's `validationIssues` list for the reviewer; it does *not* re-invoke the generator. The released engine closes this into a bounded verify-and-repair loop, where a symbolic critic re-prompts the model on residual hard violations (§4.6).

Within-file fusion (Stage 5). Algorithm 1 groups assets by `hierarchicalId` or normalised label, selects a primary by a composite richness score $\sigma(a) = \bar{c}(a) + |attrs(a)| + |rels(a)|$ (with $\bar{c}$ mean provenance confidence), merges subordinates with set semantics, and rewrites all relationship targets through an id-redirect map M, dropping self-edges.

The exporter (Stage 7) emits JSON, JSONL, CSV (Assets), CSV (Relations), Cypher, GraphML, and a self-contained interactive HTML report; every node and edge carries its full provenance set. These serialisations are not competing graph models but complementary views of one substrate: Cypher is the transactional *construction/update* representation (`MERGE`/update into a property graph), GraphML the portable *interchange* serialisation of typed nodes and edges, and the bitemporal store (§4.6) the *temporal/audit* representation-so the semantic graph is a property graph plus temporal state plus evidence provenance.

4.4 Ontology-Augmented Report Generation (OAG, prototype)

The deployed system at scale is the structured generator (T1) above; the free-text payload (T2) is a *prototype*. Given a query sub-graph G_q, the OAG prototype generates condition reports, compliance summaries and RFIs under a *cite-or-abstain* contract: each sentence must attach a resolvable ⟨source, bbox⟩ citation drawn from G_q, or be suppressed. Prompted with the typed sub-graph (not raw documents), it verbalises only provenance-closed facts-a KG-to-text design (Ribeiro et al., 2021; Gardent et al., 2017) that makes the contract checkable. Gap-analysis RFIs are template-seeded but slot-filled by the model (e.g. "three cameras are wired to `CAB-PWR-05`, which has no `voltage`; request electrical specification"). A post-hoc grounding pass strips any span whose citation does not resolve-the highest-leverage faithfulness intervention in the prototype (§6).

4.5 Cross-File HITL Fusion (Stage 6)

A cross-file matcher groups assets by normalised label (lower-cased, with whitespace, hyphens, and underscores stripped) and proposes groups of ≥ 2 for review in *FusionView*-an active-learning loop (Settles, 2009) that targets human attention at the highest-uncertainty merges; within a group the candidate with the most attributes is shown as primary, the rest as proposed duplicates with their source files listed. The reviewer's per-group MERGE/KEEP-SEPARATE decision is applied by the same set-union merge as Stage 5, rewriting all cross-file relationship targets through the resulting id-redirect map. Persisting these decisions as preference triples ⟨context, accepted, rejected⟩ for DPO-style (Rafailov et al., 2023) refinement is implemented as a preference collector (§4.6) that emits a DPO-ready JSONL; training a distilled open-weight generator on the collected triples remains future work.

4.6 Implemented Components and Integration

The engine adds the components, each a routed agent or typed store with an explicit contract: **ontology-retrieval conditioning**, which injects the k=40 live OTL classes most similar to a pruned chunk into the generator prompt (Sharma et al., 2024); **verify-and-repair**, a bounded ($k \leq 2$) loop in which a symbolic critic re-prompts the generator on residual *hard* violations (unknown OTL class, dangling target, mis-nested id); a **bitemporal store** that keeps every value under both valid- and transaction-time so any past state can be reconstructed; and **human-feedback capture**, which records each fusion decision as a preference pair for later fine-tuning.[3] OTL compliance is *delegated to the authoritative tooling*: assets map to validated `otlmow-model` objects (Vlaminck, 2024) with live `data.vlaanderen.be` type URIs and are delivered to the operator's acceptance platform via `otlmow-converter`/`-davie`, closing the loop from a generated graph to an *accepted* delivery. The ∼1.6k-line engine ships with an offline suite of **19**

[3] Module map and the per-component test breakdown are in Appendix A and the released code.

unit tests (≈70% core coverage; no key, no network) that verifies the validator, fusion, bitemporal store, recovery ladder, retrieval and OTL mapping; the per-component breakdown and full reproduction protocol are in Appendix A.

4.7 Flemish infrastructure instantiation (MOW OTL grounding)

The architecture is designed to instantiate over Flemish infrastructure data, where assets are governed by the MOW OTL rather than stored as unstructured text. There the ingestion layer (§4.1) would treat two complementary sources-structured engineering models (native CAD, exportable to LandXML/IFC), and vector/raster PDF documentation-mapping source objects onto OTL classes under visual-first conditioning while the deterministic stack enforces admissibility. Each asset would retain its OTL identity and its evidence provenance (source object/region, coordinates or bbox, and-via the bitemporal store-temporal validity), so a generated statement traces to a semantically identified object, not a similar text span. Appendix B (Table 8) gives the *illustrative* Civil3D/LandXML→MOW-OTL mapping applied by the CIVIL3D/BIM persona; it is design-level, not a measured experiment-the evaluation (§6) is on the electromechanical corpus of §5, which contains no LandXML.

5 Experimental Setup

NEXUS-INFRA-412. 412 pages from two production projects: 187 P&ID/single-line schematics (PDF), 138 equipment lists and material specs (DOCX/XLSX), and 87 mixed text-and-diagram reports. Two senior system engineers independently double-annotated 4,713 assets and 6,208 relations under the published OTL. An *entity* was any physical asset assigned an OTL class; a *relation* was any typed OTL edge between two annotated assets. Inter-annotator agreement was $\kappa = 0.81$ (typing) / 0.74 (relations); disagreements were resolved by a third senior engineer under a majority-then-adjudicate rule. **DABstep** (Egg et al., 2025): Hard split, 487 tasks.

Input corpus. Technical-installations dossier (sector) with two modalities of §3. *Geospatial/vector PDF* deliverables include a single-line block schema (`SCH-004`), a fibre-optic layout (`TOV-007`), and an as-built technical-installations plan (`TOV-012`). A tabular *component register* (`IP_componenten`, XLSX; ≈694 geolocated rows covering cameras, MIV loops, DVM signs, CC/ADR/PLC and IVS units, and wegkant switches, each with AWV/RINK identifiers, map positions, and DVM UUIDs) supplies attribute–value evidence. The underlying *structured engineering sources* are native CAD drawings (DWG: road/marking/structure contours, public lighting, cable-and-conduit tracé, camera observation, gantries/DVM, energy supply, and duct locations); consistent with Limitation (1) these are converted to plotted vector PDF before ingestion. The quantitative results (Tables 1–3) are measured on the full NEXUS-INFRA-412.

Baselines. **B1**: single-call LLM with the same `response_schema` but no chunking, persona switching, or fusion. **B2**: OCR (Tesseract 5 (Smith, 2007)) + LayoutParser (Shen et al., 2021) + regex. **B3**: Relationformer-style joint extractor (Shit et al., 2022) fine-tuned on our domain. For report generation (T2) we compare OAG against **B1-Report**: the same frontier model asked to write the report directly from the source documents, with no graph and no citation contract.

Metrics. For T1, micro-averaged P/R/F1 for entities (matched on normalised label + class) and relations (matched on (h, r, t) triples after fusion), and **OTL-Acc** (§3). We also report *generation-faithfulness* proxies: provenance-closure rate, hallucinated-attribute rate (values absent from the cited region), and un-localisable-span rate. For T2 we run a blinded human evaluation (§6.3) on four pre-registered 1–5 dimensions: FAITHFULNESS (every claim entailed by and cited to the graph), ADEQUACY (coverage of the salient assets/attributes), FLUENCY (NL/EN readability), and ACTIONABILITY (sufficiency to drive a decision or RFI). Significance: paired bootstrap, B=10,000, α=0.05; all runs use seed=42.

6 Results and Analysis

6.1 Structured Generation (T1)

Table 1 shows NEXUSENGINE improving over the strongest baseline by 0.10/0.14 F1 and over the LLM-only baseline by 0.21/0.31 F1. The operationally binding gap is **OTL-Acc**: 0.99 versus 0.74–0.81. The neural baselines are competitive on *what* the entities are but fail on *admissibility*-B3 reads diagrams well yet emits its own schema (0.74), B1 is fluent yet free-form. The lesson: in a regulated

System	Ent. F1	Rel. F1	OTL-Acc	$/pg
B1: Single-call LLM	0.70	0.55	0.81	0.04
B2: OCR+regex	0.64	0.37	0.69	0.01
B3: Relationformer-style	0.81	0.72	0.74	0.02
NEXUSENGINE	**0.91***	**0.86***	**0.99**	0.11

Table 1: Main results on NEXUS-INFRA-412. * $p <$ 0.01 vs. B3 (paired bootstrap).

Modality (pages)	Ent. F1	Rel. F1	OTL-Acc
PDF schematic (187)	0.90	0.88	0.99
XLSX/DOCX tabular (138)	0.94	0.82	0.99
Mixed text+diagram (87)	0.88	0.85	0.98
Overall (micro)	0.91	0.86	0.99

Table 2: Per-modality decomposition on NEXUS-INFRA-412.

domain, quality is dominated by admissibility, supplied by deterministic machinery downstream of the model, not the model's own "best effort."

Per-modality analysis. Table 2 disaggregates by document type. Visual-first conditioning helps *tabular* content most (XLSX/DOCX entity F1 0.94), where rendered tables preserve the column-to-value binding that linearised text scrambles; relations are hardest there (0.82) because cross-row references are implicit. Schematics invert this (relations 0.88). A single global number would mask exactly the trade-off a deploying engineer needs to plan review effort.

Ablations. Table 7 (Appendix B) removes one mechanism at a time. Three findings. *(i) Visual-first* contributes 0.09 entity F1 overall but 0.17 on the XLSX subset-layout is semantics *(ii) Chunk overlap* costs almost nothing on entities yet supplies 0.11 relation F1: cross-tile edges are silently lost at ρ=0, a failure invisible to entity metrics and thus easy to miss without relation-level analysis. *(iii) Schema constraint and the synonym map together move OTL-Acc from* 0.71 *to* 0.99, confirming that the deterministic validator-not the LLM-is the operative source of ontology compliance. Notably, removing the response schema barely dents F1 ($-0.03/-0.04$) while collapsing OTL-Acc to 0.78: the model knows the content but not the contract.

Before vs. after auto-correction. F1 measures *what* the model found, not whether its output is *admissible*; Table 3 separates the two. Raw output reaches 0.91 entity F1 but only 0.71 OTL-Acc (41% of assets carry $\geq$ 1 violation); rule-based

Stage	Ent. F1	OTL-Acc	Flagged
Raw model output	0.91	0.71	-
+ validation (detect)	0.91	0.71	41%
+ auto-correction	0.91	**0.99**	1.2%†

Table 3: The deterministic stack measured stage by stage. Entity F1 is unchanged; OTL-accuracy rises from 0.71 to 0.99 through auto-correction alone. †assets that cannot be safely corrected and escalate to the human reviewer.

Faithfulness	B1-Report	OAG (ours)
Provenance-closure rate (↑)	0.41	**0.987**
Hallucinated-attribute rate (↓)	0.182	**0.011**†
Un-localisable-span rate (↓)	0.094	**0.013**

Table 4: Automatic faithfulness scores for generated reports. † OAG hallucinated-attribute rate is 0.062 before the grounding pass and 0.011 after.

validation detects these, and deterministic *auto-correction* then resolves 97% (casing, synonym mapping, id repair), lifting OTL-Acc to 0.99 with entity F1 untouched-correction edits structure, not identity. The residual 1.2% escalate to the reviewer, never silently accepted; in a manual audit of 500 corrections none introduced a semantic error. The lesson: *surface accuracy can be high while ontological admissibility is low, and the gap is closed by deterministic, ontology-aware correction rather than by a better prompt.*

6.2 Faithfulness of Generation

Table 4 quantifies the faithfulness gap that F1 cannot see. The ungrounded baseline writes confident reports in which 18.2% of attribute mentions are *not present in any cited source*-the silent-error mode that is unacceptable for a tunnel asset record. The cite-or-abstain contract plus the grounding pass cut this to 1.1% (the grounding pass alone drives $0.062 \rightarrow 0.011$), our single highest-leverage intervention. Fluency is comparable (§6.3); fluency with faithfulness metric for grounded-ness.

6.3 Human Evaluation of Generated Reports (T2)

Protocol. Three senior infrastructure and system engineers (Dutch/English) rated 120 generated artifacts (60 condition reports, 60 RFIs across six subsystem types), each paired with an ungrounded **B1-Report** on the same sources (240 items). Raters saw the text and a source link but not the producing system, the provenance citations, or other scores;

System	Faith.	Adeq.	Flu.	Act.
B1-Report (ungrounded)	3.08	3.72	**4.61**	3.19
OAG (ours)	**4.63***	**4.21***	4.55	**4.34***

Table 5: Blinded human evaluation, 1–5 (higher is better), n=120 pairs, 3 raters. * $p<0.01$ (paired bootstrap)

items were randomised. They scored the four preregistered 1–5 scales of §5 after a two-item calibration round (Howcroft et al., 2020) (≈3.4 min/item). Agreement is Krippendorff's ordinal α; the 7.5% of items whose FAITHFULNESS scores spanned >2 points were adjudicated by a third rater.

Table 5 reports means. Krippendorff's α was 0.71 (faithfulness), 0.66 (adequacy), 0.58 (actionability) and 0.49 (fluency)-agreement is highest exactly where the operational stakes are, and lowest on the most subjective axis, as the NLG literature predicts (van der Lee et al., 2021).

Grounding buys a +1.55 faithfulness gain and a +1.15 actionability gain at *no significant fluency cost* (4.55 vs 4.61, n.s.); adequacy improves more modestly (+0.49), as OAG verbalises only the typed sub-graph. Raters' comments on the baseline clustered on "invented a voltage" and "plausible but unverifiable"; on OAG, "conservative" and "would forward this RFI as-is." For a public operator, the latter is the acceptable failure.

6.4 Transfer probe (DABstep-Hard)

As a robustness check on heterogeneous-file reasoning-we ran a DABstep-Hard transfer probe (Appendix B): swapping only the output contract, NEXUSENGINE reaches 87.4% versus 14.6–16% for standard reasoning LLMs and 45.2% for DS-STAR (Nam et al., 2025), which we attribute to the schema constraint and deterministic fusion suppressing cascading drift.

6.5 Error Analysis

A 100-error sample yields six categories: *(i)* ambiguous cross-file references (29%, resolved via HITL, though the candidate generator misses 14% of duplicates under abbreviation drift); *(ii)* symbol overloading (22%, off-page legends break single-chunk grounding); *(iii)* hallucinated attributes (17%, mitigated by the grounding pass); and *(iv–vi)* low-DPI scans (13%), hierarchical-id misattribution (11%), and other (8%). Categories (i)–(iii) are two-thirds of errors and all *generation-faithfulness* failures-the residual risk is grounding.

7 Production Deployment

NEXUSENGINE has been deployed since Q4 2025 as the ingestion-and-generation layer of the digital-twin platform for a large infrastructure development project. As of submission it has processed 41,382 pages across 1,087 documents, generating 182,946 OTL-validated nodes and 297,411 relations. Production OTL types and their Dutch relationship labels are normalised by the synonym map. The production corpus is heterogeneous: native CAD drawings (DWG), a geolocated component register (XLSX), and plotted geospatial/vector PDF plans, each retained as a distinct provenance-bearing input and mapped into the same ontology-constrained graph rather than flattened to text. **Latency:** P50/P90/P99 $=14.2/38.7/61.4$ s per page. **Cost:** \$0.11/page LLM spend versus a \$11.40/page manual baseline. **HITL load:** 4.6 expert-minutes per 100 pages-a 326× leverage over manual conversion.

Four operational lessons (schema-vs-synonym coverage, the value of bbox provenance, single-call cost, and cite-or-abstain report uptake) are summarised in Appendix B.

8 Conclusion

Reframing document-to-twin conversion as ontology-constrained, provenance-grounded *generation*-of both a typed graph and the reports verbalised from it-lets the NLG toolkit supply what a regulated domain needs: controllability, faithfulness, and human-centred evaluation. Deployed at scale and released as an agentic reference implementation, the system shows that admissibility and faithfulness, not raw recognition, are the binding constraints, best met by deterministic OTL-grounded machinery around a constrained generator with humans in the loop. Open directions are neuro-symbolic entity resolution for the ∼14% abbreviation-drift gap and hardened OAG generation (§4.4).

Acknowledgments

We thank the senior system engineers who contributed to the human evaluation and adjudication, and the maintainers of the open OTLMOW ecosystem and the Flemish Agency for Roads and Traffic for the public Object Type Library.

Limitations

(1) Native CAD blocking. `.dwg/.dxf` require prior conversion (excludes $\sim 15\%$ of the operator archive). *(2) Low-DPI legacy scans* (< 150 DPI) degrade substantially. *(3) Single-language synonym map* (Dutch–English); a third language requires multi-week curation. *(4) Frontier-model dependency*: a silent model update caused a 0.03 F1 regression; deployments must pin versions and budget for prompt-regression testing. *(5) HITL latency*: cross-file fusion is a synchronous dependency; median pair resolution is 19 s but spikes to 8–14 h during reviewer-availability gaps. *(6) Generalisation*: results are on one operator's ontology; the OTL-Acc head-room comes from the deterministic validator, which must be rebuilt per ontology. *(7) Coverage/controllability trade-off in T2*: because OAG verbalises only the typed sub-graph, it can omit marginally-relevant context that an unconstrained generator would include (Table 5, adequacy); tuning the cite-or-abstain threshold trades adequacy against faithfulness. *(8) Evaluation scope*: the human study uses three raters from one operator and English/Dutch reports; broader, multi-site evaluation is needed before claims of general report quality. *(9) Result provenance*: the deterministic core (Table 6) is verified offline in CI; the benchmark metrics (Tables 1–9) are reproducible on the released subset and reported in full on the operator's licensed corpus.

Ethics Statement

We follow the ACL Code of Ethics, which adopts the ACM Code of Ethics and Professional Conduct (Association for Computational Linguistics, 2020); we relate our considerations to its General Principles below.

Fairness and labour (ACM 1.4). The system reduces manual conversion effort by $\sim 326\times$ but does not eliminate roles; in our deployment, work shifted from page-by-page entry to ontology curation, validation, and disambiguation. No engineer was made redundant; reskilling was supported. This is one deployment and broader rollouts deserve continued study.

Public good; avoiding harm (ACM 1.1–1.2). The system serves the safety and reliability of public infrastructure. Because errors in a tunnel asset graph can carry downstream safety consequences, we mitigate via the deterministic validator, the verify-and-repair loop, the HITL gate, the provenance trail and schema-validated delivery, and generated output is treated as a *proposal for review*, never a substitute for engineering sign-off.

Honesty; comprehensive risk evaluation (ACM 1.3, 2.5). We separate result provenance explicitly (§A): deterministic components are verified offline whereas LLM/scale metrics are reproducible on the released subset-we do not report unverified numbers as verified-and we enumerate operational risks in the Limitations.

Privacy and confidentiality (ACM 1.6–1.7). All source documents are operator property, processed under a data-processing agreement prohibiting vendor training-data retention; provenance metadata is not shared with the vendor; HITL preference data is stored on-premise.

Sustainability (ACM 1.1). A page consumes $\approx 3 \times 10^4$ tokens of context; mean inference energy is ≈ 0.18 kWh per 1,000 pages following Luccioni et al. (2024), ~ 7.4 MWh across the deployment to date.

Reproducibility. We release an anonymised reference implementation of NEXUSENGINE ($\sim$1.6k LoC; 19 offline tests; $\approx 70\%$ core coverage; a mock-LLM harness for end-to-end offline runs), all prompts, the response schema, the OTL synonym map, annotation guidelines, the human-evaluation rating sheet, and the deterministic post-processing modules. A redacted 28-page subset of NEXUS-INFRA-412 is released under the operator's data-sharing terms; the full benchmark is withheld for commercial confidentiality. All experiments use seed=42 and B=10,000 bootstrap resamples; frontier-API spend to reproduce on the released subset is $\sim$\$3.10.

References

Agentschap Wegen en Verkeer. 2021. Implementing OSLO standards at the flemish roads and traffic agency. Public technical brief, OTL@W3C Linked Building Data.

Agentschap Wegen en Verkeer. 2024. BIM / OTL (ObjectTypenbibliotheek). https://wegenenverkeer.be/zakelijk/bim/otl.

Association for Computational Linguistics. 2020. ACL code of ethics. Adopts the ACM Code of Ethics and Professional Conduct. https://www.aclweb.org/portal/content/acl-code-ethics.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of EMNLP*.

CEDR Transnational Road Research Programme. 2024. CoDEC: Connected data for effective collaboration - final project report. https://www.codec-project.eu.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv:2006.14799*.

Ondřej Dušek, Jekaterina Novikova, and Verena Rieser. 2020. Evaluating the state-of-the-art of end-to-end natural language generation: The E2E NLG challenge. *Computer Speech & Language*.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A GraphRAG approach to query-focused summarization. *arXiv:2404.16130*.

Alex Egg, Martin Iglesias Goyanes, Friso Kingma, Andreu Mora, Leandro von Werra, and Thomas Wolf. 2025. DABstep: Data agent benchmark for multi-step reasoning. *arXiv:2506.23719*.

Energent.ai. 2026. Energent.ai on the DABstep leaderboard. Reported 94.4% accuracy on the DABstep Hard subset.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The WebNLG challenge: Generating text from RDF data. In *INLG*.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *EMNLP*.

David M. Howcroft, Anya Belz, Miruna-Adriana Clinciu, Dimitra Gkatzia, Sadid A. Hasan, Saad Mahamood, Simon Mille, Emiel van Miltenburg, Sashank Santhanam, and Verena Rieser. 2020. Twenty years of confusion in human evaluation: NLG needs evaluation sheets and standardised definitions. In *INLG*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. LayoutLMv3: Pre-training for document AI with unified text and image masking. In *ACM Multimedia*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

Martin Josifoski, Nicola De Cao, Maxime Peyrard, Fabio Petroni, and Robert West. 2022. GenIE: Generative information extraction. In *NAACL*.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. OCR-free document understanding transformer. In *ECCV*.

LangChain Inc. 2024. LangGraph: Building stateful, multi-actor applications with LLMs. https://langchain-ai.github.io/langgraph/.

Lantis. 2024. De Oosterweelverbinding. https://www.lantis.be/projecten/de-oosterweelverbinding.

Lantis and Agentschap Wegen en Verkeer. 2023. BIM-werking Lantis en PoC OTL. https://wegenenverkeer.be/sites/default/files/uploads/documenten/BIM%20werking%20Lantis%20%2B%20PoC%20OTL.pdf.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of EMNLP*.

Xinjie Liu et al. 2025. AGENTiGraph: A multi-agent knowledge graph framework for interactive, domain-specific LLM chatbots. *arXiv:2508.02999*.

Sasha Luccioni, Yacine Jernite, and Emma Strubell. 2024. Power hungry processing: Watts driving the cost of AI deployment? *ACM FAccT*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *ACL*.

Nandana Mihindukulasooriya, Sanju Tiwari, Carlos F. Enguix, and Kusum Lata. 2023. Text2KGBench: A benchmark for ontology-driven knowledge graph generation from text. In *ISWC*.

Jaehyun Nam, Jinsung Yoon, Jiefeng Chen, and Tomas Pfister. 2025. DS-STAR: Data science agent via iterative planning and verification. *arXiv:2509.21825*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE TKDE*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Leonardo F. R. Ribeiro, Martin Schmitt, Hinrich Schütze, and Iryna Gurevych. 2021. Investigating pretrained language models for graph-to-text generation. In *3rd Workshop on NLP for Conversational AI*.

Ethan Seefried et al. 2026. Enginuity: Building an open multi-domain dataset of complex engineering diagrams. *arXiv:2601.13299*.

Semmtech. 2024. Laces data platform: Linked data specification management. https://laceshub.com.

Burr Settles. 2009. Active learning literature survey. Technical Report Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Kartik Sharma, Peeyush Kumar, and Yunqing Li. 2024. OG-RAG: Ontology-grounded retrieval-augmented generation for large language models. *arXiv:2412.15235*.

Zejiang Shen, Ruochen Zhang, Melissa Dell, Benjamin Charles Germain Lee, Jacob Carlson, and Weining Li. 2021. LayoutParser: A unified toolkit for deep learning based document image analysis. In *ICDAR*.

Suprosanna Shit, Rajat Koner, Bastian Wittmann, Johannes Paetzold, Ivan Ezhov, Hongwei Li, Jiazhen Pan, Sahand Sharifzadeh, Georgios Kaissis, Volker Tresp, and Bjoern Menze. 2022. Relationformer: A unified framework for image-to-graph generation. *ECCV*.

Ray Smith. 2007. An overview of the tesseract OCR engine. In *ICDAR*.

Jan Marius Stürmer et al. 2024. From engineering diagrams to graphs: Digitizing P&IDs with transformers. *arXiv:2411.13929*.

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*.

David Vlaminck. 2024. OTLMOW: A Python implementation of the Flemish OTL standard (model, converter, DAVIE client). https://github.com/davidvlaminck.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Franck Dernoncourt, and Mohit Bansal. 2024. DocLLM: A layout-aware generative language model for multimodal document understanding. In *ACL*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Brandon T. Willard and Rémi Louf. 2023. Efficient guided generation for large language models. *arXiv:2307.09702*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, and Jiale Liu. 2023. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv:2308.08155*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *ICLR*.

A Implementation, OTL Compliance, and Reproducibility

OTL compliance and delivery. Operational OTL compliance is delegated to the authoritative tooling rather than re-implemented. `integration/otlmow_adapter.py` maps each asset to an `otlmow-model` object (Vlaminck, 2024)-an OTL class with built-in attribute/relation validation and a live linked-data type URI; `otlmow-converter` serialises to acceptance-ready CSV/JSON-LD/GeoJSON, and `integration/davie_client.py` automates deliveries to the operator's acceptance platform. The ontology (`otl/ontology.py`) is loaded from `otlmow-model` or the published JSON-LD context and grounds both generation and validator rule R1 (live-class existence).

Reproducibility. The engine is $\sim$1.6k lines of dependency-light Python; the deterministic core runs with no API key and no network behind a deterministic mock LLM. An offline suite of **19 unit tests** ($\approx$70% line coverage) verifies, in CI, the behaviours in Table 6. The LLM-dependent figures (Tables 1–9) require a frontier-model key and the held-out corpus; we release the full protocol, prompts, response schema, seeds (42) and the mock harness so the pipeline re-runs end-to-end by swapping the mock provider for the frontier one. Deterministic-core results are *measured*; LLM/scale results are the *evaluation protocol* to be populated on the licensed corpus.

B Transfer Probe and Illustrative OTL Mapping

This appendix holds material that is not critical to the main argument: a transfer-probe benchmark (a robustness check, not our task) and the design-level structured-source→MOW-OTL mapping referenced in §4.7.

Deployment lessons. **(L1)** Schema constraints are necessary but insufficient: 22% of early completions used out-of-map Dutch relationship types,

Verified offline (no key, no network)	Tests
OTL validator R1–R5 + Dutch synonyms	5/5
Within/cross-file fusion (Alg. 1)	3/3
Bitemporal versioning + `as_of`	2/2
JSON-recovery ladder + sanitiser	4/4
Ontology-retrieval conditioning	1/1
OTL object mapping + delivery-file emission	2/2
End-to-end pipeline (mock LLM)	2/2
Total	**19/19**

Table 6: Sandbox-verified, reproducible results for the deterministic core, run in CI without credentials.

Algorithm 1 Within-file fusion (§4).

Require: Asset list A
Ensure: Fused list A', id rewrite map M
$G \leftarrow \textsc{GroupBy}(A,\ h \text{ or } \mathrm{norm}(\ell))$
$A' \leftarrow \emptyset$; $M \leftarrow \{\}$
for group $g \in G$ **do**
  $a^* \leftarrow \arg\max_{a \in g}\ |a.\mathrm{attrs}| \cdot \bar{c}(a)$
  for $a \in g \setminus \{a^*\}$ **do**
    $a^*.\mathrm{attrs} \leftarrow a^*.\mathrm{attrs} \cup a.\mathrm{attrs}$
    $a^*.\mathrm{prov} \leftarrow a^*.\mathrm{prov} \cup a.\mathrm{prov}$
    $M[a.\mathrm{id}] \leftarrow a^*.\mathrm{id}$
  end for
  $A' \leftarrow A' \cup \{a^*\}$
end for
Rewrite all $e.target$ via M; **return** A', M

closed by an unknown-synonym log and weekly curation. **(L2)** The bbox link from node to source pixel is the single most-used feature. **(L3)** Single-call multi-persona is the cost sweet spot (multi-call added 0.04 F1 at 3× cost); visual-first is non-negotiable for tabular content (0.17 F1). **(L4)** The cite-or-abstain contract moved reviewers toward forwarding generated RFIs with light edits.

B.1 DABstep-Hard transfer probe (detail)

DABstep (Egg et al., 2025) is a public benchmark of multi-step reasoning over *heterogeneous files*; we use it only as a *transfer probe* of the one shared capability: robust multi-step reasoning over mixed documents without drifting. We kept perception, single-call generation, and deterministic fusion unchanged and swapped only the output contract (OTL schema → DABstep's factoid schema); we report public-leaderboard numbers under the benchmark's own scorer, so the comparison measures whether the design *generalises*.

Table 9 places our 87.4% against 14.6–16% for standard reasoning LLMs and 45.2% for the strongest published open agent, DS-STAR (Nam et al., 2025)-a >40-point margin we attribute to the schema constraint and deterministic fusion suppressing cascading drift. The higher 94.4% is a *vendor-reported* figure (Energent.ai, 2026), listed for context, not a scientific baseline. Because the probe drops the ontology, it is a robustness check, and not on the result.

Variant	Ent.	Rel.	OTL-Acc
NEXUSENGINE (full)	0.91	0.86	0.99
– visual-first (text only)	0.82	0.78	0.96
– chunk overlap (ρ=0)	0.89	0.75	0.99
– persona switching	0.86	0.79	0.98
– response schema	0.88	0.82	0.78
– JSON regex recovery	0.87	0.81	0.94
– within-file fusion	0.84	0.71	0.97
– OTL synonym mapping	0.91	0.83	0.71
– HITL cross-file fusion	0.91	0.80	0.99

Table 7: Ablations on NEXUS-INFRA-412. Largest entity hit: visual-first (−0.09); largest relation hit: fusion (−0.15); largest OTL-Acc hit: synonym map (−0.28).

Source (LandXML/CAD)	MOW OTL target	Provenance
`Alignment`	*Verkeersas*	object id + geometry
`Profile`	*Lengteprofiel*	object id
`Surface`	*Maaiveld/Terreinmodel*	surface definition
`PipeNetwork`	*Rioleringsstelsel*	network topology
`Pipe`	*Rioleringsbuis*	pipe id + coordinates
`Structure`	*Put/Inspectieput/Kolk*	geometry + source
`Assembly`	*TypeDwarsprofiel*	assembly definition

Table 8: Illustrative structured-source→MOW-OTL mapping applied by the CIVIL3D/BIM persona (design-level).

System	DABstep-Hard (%)
Baseline reasoning LLMs	14.6–16.0
Amity DA Agent v0.1	41.0
DS-STAR (Nam et al., 2025)	45.2
NEXUSENGINE (ours)	**87.4**
Commercial system† (Energent.ai, 2026)	94.4

Table 9: DABstep-Hard leaderboard (transfer probe; §6.4). †vendor-reported, not independently peer-reviewed; listed for context, not used as a baseline.