Indonesian Journal on Software Engineering (IJSE)
Vol. 12, No. 1, June 2026, page. 59-68 p-ISSN: 2461-0690
Akreditasi KEMENRISTEKDIKTI, No. 225/E/KPT/2022 e-ISSN: 2714-9935

A DETERMINISTIC ONTOLOGY-DRIVEN FRAMEWORK FOR GOVERNANCE-AWARE AUTOMATED SOFTWARE REQUIREMENTS SPECIFICATION GENERATION

Alaa Abd Elazheim*[1], Farid Meziane[2], Omayma Husain[3]

[1]Department of Information Technology, Sudan University of Science and Technology, Khartoum, Sudan
[2]Data Science Research Centre, University of Derby, Derby, UK
[3]Department of Information Technology, Faculty of Mathematical Sciences and Informatics, University of Khartoum, Khartoum, Sudan
Email: [1]alaaabdelazheim@yahoo.com, [2]farid.meziane@gmail.com, [2]omaymahusain@gmail.com
*Corresponding Author

Abstract

Automated generation of Software Requirements Specifications (SRS) remains a persistent challenge in requirements engineering, particularly in governance-intensive environments where specifications must simultaneously preserve structural completeness, integrate domain-specific governance constraints, maintain explicit traceability, and support reproducible generation behaviour. Existing ontology-based, template-based, and AI-based approaches address these requirements only in isolation. This paper proposes and evaluates a deterministic ontology-driven transformation framework that generates SRS artifacts from an OntoUML-derived conceptual model (M), a governance-oriented domain ontology (D), and a declarative governance rule set (R), formalised as S = F(M, D, R). The framework employs a six-layer transformation architecture combining exhaustive baseline structural mapping with governance-aware semantic enrichment through declarative rule-based activation over a selectively merged ontology. A Python prototype using OWLReady2 was implemented and evaluated across five Service Request Management datasets and five FIBO-derived Financial Compliance datasets, using four metrics: Structural Coverage Ratio (SCR), Governance Enrichment Ratio (GER), Traceability Coverage (TC), and Deterministic Reproducibility (DR). Results confirm SCR = 1.00 and TC = 1.00 across all ten datasets, GER progressing from 0.0000 in baseline configurations to 0.1250 and 0.1429 in the two domains respectively, and DR confirmed through ten independent executions per dataset. These findings demonstrate that ontology-driven deterministic transformation provides a viable, extensible, and governance-aware foundation for automated SRS generation, addressing limitations present in existing template-based, ontology-based, and AI-based approaches.

Keywords: software requirements specification, ontology-driven requirements engineering, deterministic transformation, governance-aware generation, rule-based activation, FIBO

1. INTRODUCTION

Software Requirements Specification (SRS) constitutes a foundational artifact in software engineering, providing a formal description of system functionality, constraints, and stakeholder expectations that guides the development lifecycle. Empirical studies consistently identify deficiencies in requirement specifications — ambiguity, inconsistency, incompleteness, and weak traceability as leading causes of software project failure and increased development cost (Schulze and Pretorius, 2013; Fariha, Alwidian and Azim, 2024). Despite this critical role, SRS production remains predominantly manual: even where conceptual models are systematically constructed, transforming structured semantic knowledge into natural language specifications is typically treated as a separate authoring activity, introducing a methodological disconnect between the two.

This disconnect is most pronounced in governance-intensive environments, where requirements must reflect not only system behaviour but also domain-specific governance constraints — service-level agreements, compliance obligations, audit requirements, and escalation mechanisms (Aberkane, Poels and Vanden Broucke, 2021; Khan et al., 2021). Existing practice integrates such semantics through manual or loosely structured processes, producing fragmented, inconsistent, and non-reproducible specifications, compounded by the difficulty of manually satisfying simultaneous constraints from multiple governance sources.

Three principal paradigms address automated SRS generation, each with significant limitations discussed in the Related Work section below: ontology-based approaches provide semantic grounding without generation capability, template-based approaches offer partial generation without deterministic guarantees, and AI-based approaches generate fluent text without determinism or explicit traceability.

This fragmentation reveals a fundamental gap in the current state of the art: no existing approach simultaneously provides semantic grounding, deterministic transformation, governance-aware enrichment, traceability by construction, and reproducible generation within a unified architecture. Rather than representing a marginal improvement over existing methods, addressing this gap requires a paradigm shift, from treating requirement generation as a document-authoring activity to treating it as a formally defined semantic transformation process. This paper proposes such a shift by formalizing SRS generation as the deterministic transformation function:

$S = F(M, D, R)$ (1)

where M denotes the OntoUML-derived conceptual model, D the governance-oriented domain ontology, and R the declarative governance rule set. F is deterministic — identical (M, D, R) always produce logically equivalent outputs — and integrates a six-layer transformation architecture in which baseline structural mapping and governance-aware semantic enrichment operate within Layer 5.

The framework was empirically evaluated across two domains — Service Request Management (DS1–DS5) and FIBO-derived Financial Compliance (FDS1–FDS5) — confirming SCR = TC = 1.00, systematic GER progression, and DR = true across all ten datasets (Section 3).

The principal contributions of this paper are: (1) a formal deterministic transformation model S = F(M, D, R) that defines SRS generation as an analyzable mapping from structured semantic inputs to requirement outputs; (2) a novel selective materialized merge mechanism that integrates domain ontology elements proportionally to governance content, preventing spurious rule activations; (3) a declarative rule-based activation mechanism that generates governance-aware requirements conditionally based on verifiable semantic trigger conditions; (4) traceability by construction linking every generated requirement to its originating ontology element or activated rule in a machine-recoverable form; and (5) empirical validation across two structurally distinct domains confirming all four formal properties.

The remainder of this paper is organized as follows. The following section reviews related work across three paradigms. Section 2 presents the proposed framework, including the formal model, architecture, formal properties, and prototype implementation. Section 3 presents the empirical evaluation across both domains together with a comparative discussion positioning the framework within the state of the art. Section 4 concludes the paper.

Related Work

Ontology-Based and Model-Driven Approaches

Ontology-based approaches establish formal vocabularies enabling domain knowledge to be represented precisely. Alrumaih, Mirza and Alsalamah (2020) propose a domain ontology for requirements classification, reducing ambiguity and improving consistency. Alsanad, Chikh and Mirza (2019) present an ontology for requirements change management in global software development, supporting evolution and dependency modelling. Mosquera, Ruiz and Pastor (2025) introduce an ontology-based NLP tool for tracing requirements and conceptual models across industrial case studies. Mokos et al. (2022) apply OWL-based reasoning to improve the semantic precision of existing specifications. Complementing these, Gupta, Poels and Bera (2022) show that conceptual models preserve domain knowledge in agile development, while Gillani, Niaz and Ullah (2022) propose architecture-based integration linking system architecture to requirement derivation. Despite these contributions, such approaches remain analytical rather than generative: they support classification, validation, and traceability but do not operationalise semantic knowledge into executable generation processes, nor do they use governance constraints to trigger requirement generation — the gap this paper addresses.

Template-Based Generation Approaches

Darif et al. (2023) propose a model-driven, template-based approach (MD-RSuT) combining controlled natural language templates with domain models for semi-formal specification and validation. Under criteria requiring structured input, an explicit transformation mechanism, and the capability to produce requirement artifacts as output, this is the only reviewed study satisfying all three conditions — itself evidence of how scarce generation-oriented approaches are. However, requirement reformulation and template selection require direct user involvement, introducing variability and limiting reproducibility; the transformation is not formalised as a deterministic function; and domain models serve mainly for validation rather than as active semantic drivers. Governance integration, end-to-end traceability, and deterministic reproducibility are not addressed — the methodological boundaries the proposed framework is designed to transcend.

AI-Based and LLM-Based Approaches

Recent LLM-based approaches automate aspects of requirements engineering from a fundamentally different paradigm. Shafikuzzaman et al. (2025) investigate zero-shot and few-shot classification of requirement artifacts. Korn, Gorsch and Vogelsang (2025) propose LLMREI for conversational requirement elicitation. Quattrocchi et al. (2025) examine LLM-based user story generation from unstructured input. Voria et al. (2025) propose RECOVER for generating requirements from stakeholder conversations. Beg, O'Donoghue and Monahan (2025) investigate formalisation of requirements using LLMs, and Krishna et al. (2024) conduct an empirical evaluation of LLM-based SRS generation comparable in quality to entry-level engineers. Despite this generative strength, three limitations constrain applicability in governance-intensive contexts: semantic grounding is implicit, captured through training data rather than formal ontologies; determinism is structurally absent, with probabilistic inference producing variable outputs across executions as confirmed by (Krishna et al., 2024; Beg, O'Donoghue and Monahan, 2025); and traceability is unavailable by design, since generated statements are not explicitly linked to formal semantic sources.

Positioning of the Proposed Framework

The preceding analysis reveals consistent fragmentation: ontology-based approaches provide semantic grounding without generation capability; template-based approaches offer partial generation without formal transformation guarantees or governance integration; AI-based approaches provide powerful generation without semantic grounding, formal traceability, or reproducibility. No paradigm simultaneously addresses all dimensions required for automated governance-aware SRS generation. Table 1 summarises this comparative positioning, using qualitative labels rather than binary scores to reflect the partial capabilities of existing approaches relative to the proposed framework.

Table 1. Comparative Positioning of Existing and Proposed Approaches

Dimension	Ontology-Based	Template-Based	AI-Based	Proposed Framework
Semantic Grounding	Strong	Partial	Implicit	**Formal**
Automated Generation	None	Partial	Partial	**Systematic**
Formal Transformation	None	Partial	None	Explicit
Governance Integration	Implicit	None	None	Systematic
Traceability	Model-level	None	None	End-to-end
Determinism	N/A	Limited	None	Guaranteed
Evaluation Evidence	Partial	Partial	Partial	Quantitative

This fragmentation reflects the absence of a unifying theoretical foundation bridging semantic modelling, transformation logic, and governance representation. The proposed research addresses this directly by integrating formal semantic modelling, deterministic rule-based transformation, selective ontology integration, and traceability by construction within the unified function S = F(M, D, R), with empirical confirmation of SCR = 1.00, TC = 1.00, and DR = true across two domains evidencing that this integration is operationally realisable.

2. RESEARCH METHODS

Datasets

The framework was evaluated using ten controlled datasets across two application domains. The primary evaluation employed five datasets within the Service Request Management (SRM) domain (DS1–

DS5), constructed following a controlled semantic escalation strategy in which governance semantics are progressively introduced while the underlying conceptual model is held constant, isolating the effect of governance rule activation. A secondary evaluation employed five Financial Compliance datasets (FDS1–FDS5) derived from the Financial Industry Business Ontology (FIBO) (EDM Council, 2026), executed through the identical pipeline without code modification to validate domain-agnostic extensibility. Table 4 (Section 3) summarises the entity, relationship, and governance-rule profile of each dataset.

Design

The proposed framework formalises SRS generation as the deterministic transformation function S = F(M, D, R), where M denotes an OntoUML-derived conceptual model encoded as structured JSON, D a governance-oriented domain ontology encoded in OWL/RDF, and R a declarative governance rule set. For any fixed triple (M, D, R), the output S is uniquely and invariably determined. The framework is organised into six functionally distinct layers (Table 2): semantic input representation, system ontology construction, selective materialized merge, semantic reconciliation, governance-aware generation, and traceability and realisation. The architecture is strictly sequential and non-adaptive, with no feedback loop or stochastic variation, guaranteeing that the pipeline output is uniquely determined by its inputs.

Table 2. Framework Architecture: Layers and Functions

Layer	Name	Primary Function
L1	Semantic Input Representation	Encode conceptual model and domain ontology in machine-processable form
L2	System Ontology Construction	Transform the structured conceptual model into an OWL system ontology
L3	Selective Materialized Merge	Integrate domain semantics proportionally with the system ontology
L4	Semantic Reconciliation	Establish formal cross-ontology class correspondence
L5	Governance-Aware Generation	Generate baseline and governance requirements
L6	Traceability and Realization	Produce a structured SRS with explicit traceability

Conceptual entities, attributes, and relationships from M are mapped to OWL classes, datatype properties, and object properties respectively. Domain governance semantics from D are then integrated through a novel selective materialized merge mechanism, which applies a three-level relevance filter — matching class names, then object properties whose domain and range overlap with the system ontology, then datatype properties whose domain overlaps with it — rather than importing D in its entirety, preventing spurious governance rule activations. Semantic reconciliation subsequently links corresponding system and domain ontology classes via owl:equivalentClass declarations.

Generation operates through two composed mechanisms. Baseline structural mapping exhaustively transforms all extracted ontology elements into requirement statements through three fixed rules (Table 3), guaranteeing structural completeness by construction. Declarative rule-based activation conditionally generates governance-aware requirements when semantic trigger conditions (object property, datatype property, or compound AND conditions) are satisfied against the merged ontology; the SRM catalogue comprises seven such rules spanning SLA governance, priority management, compliance linking, audit trail, and escalation triggering, each externally configurable via YAML/JSON without modifying the core pipeline. The two requirement sets are combined through monotonic composition ($R\_total = R\_baseline \cup R\_rule$), ensuring governance requirements extend rather than replace the structural baseline.

Table 3. Baseline Structural Mapping Rules

Ontology Element	Mapping Rule	Generated Requirement Pattern
OWL Class	R1	"The system shall store and manage {ClassName} information."

Vol. 12, No. 1, June 2026, page. 59-68 p-ISSN: 2461-0690
Akreditasi KEMENRISTEKDIKTI, No. 225/E/KPT/2022 e-ISSN: 2714-9935

Datatype Property	R2	"The system shall record {propertyName} as an attribute of {domainClass}."
Object Property	R3	"The system shall maintain the relationship {propertyName} between {domainClass} and {rangeClass}."

The framework further defines four formal correctness properties — Structural Completeness, Monotonic Governance-Aware Enrichment, Traceability by Construction, and Deterministic Reproducibility — each guaranteed architecturally by the mechanisms above and operationalised through the metrics defined below.

Measures

Four quantitative metrics operationalise the framework's formal properties, computed programmatically from execution artifacts (the metrics JSON and traceability CSV) without manual estimation. Structural Coverage Ratio (SCR) is the proportion of structural ontology elements reflected in at least one baseline requirement; SCR = 1.00 confirms Structural Completeness. Governance Enrichment Ratio (GER) is the proportion of generated requirements that are governance-derived (GER = |R_rule| / |R_total|); GER = 0.000 confirms the absence of governance activation in baseline configurations, and its theoretical maximum, GER = n/(n+b) for n activated rules and b baseline requirements, is architecturally bounded below 1.00 since baseline mapping is always applied exhaustively first. Traceability Coverage (TC) is the proportion of requirements with an explicit traceability mapping; TC = 1.00 confirms Traceability by Construction. Deterministic Reproducibility (DR) is confirmed when four logical indicators and the SHA-256 hash of the traceability CSV are identical across ten independent executions per dataset, providing byte-level equivalence verification beyond simple count matching.

Procedures

The framework was implemented as an executable Python prototype using the OWLReady2 library for ontology construction and Python's xml.etree.ElementTree for ontology merging, extraction, and IRI conflict resolution — a deliberate choice over OWL reasoning engines, whose inference mechanisms can introduce non-determinism through variable materialisation order. The prototype comprises five modules mirroring the layered architecture (build_ontology, merge_ontologies, ontology_extractor, a rules evaluation engine, and run_pipeline), producing five output artifacts per execution: the generated SRS text, the merged ontology, a traceability CSV, an evaluation-metrics JSON, and an execution log.

Deterministic execution is guaranteed by three design decisions: all ontology elements are extracted via Python's sorted() function before downstream processing, eliminating serialisation-order variability; the governance rule catalogue is evaluated in a fixed sequence against deterministically ordered elements; and all transformation templates use fixed-string, deterministic placeholder substitution with no probabilistic or adaptive components. Determinism was verified by executing each dataset ten independent times and comparing the SHA-256 hash of the resulting traceability CSV across all runs; identical hashes confirm byte-level equivalence of content, ordering, and trace-link assignment. Domain-agnostic operation was validated by substituting the domain ontology and governance rule catalogue at runtime (--domain and --rules parameters) without modifying the core pipeline, executing the identical prototype against both the SRM and FIBO-derived Financial Compliance datasets.

3. RESULTS AND DISCUSSION

Evaluation Design

The evaluation adopts a property-driven validation strategy in which correctness is assessed through the measurable structural and semantic guarantees defined in Section 2 (Measures) rather than subjective or linguistic criteria; the objective is to verify that the transformation process correctly preserves and operationalises the structural and semantic knowledge encoded in the input artifacts, not to assess linguistic quality.

Table 4. Dataset Profile: Entities, Relationships, and Governance Rules Activated

Domain	Dataset	Governance Focus	Entities	Relationships	Rules Activated
SRM	DS1	Structural baseline	4	3	0

SRM	DS2	SLA and priority governance	6	5	3
SRM	DS3	Workflow and routing governance	7	7	3
SRM	DS4	Resource and skills allocation	7	7	3
SRM	DS5	Composite governance	10	11	7
Financial	FDS1	Structural baseline	3	2	0
Financial	FDS2	KYC and AML compliance	4	2	2
Financial	FDS3	Credit risk assessment	4	2	1
Financial	FDS4	Approval workflow and audit	4	2	2
Financial	FDS5	Composite governance	11	7	7

Evaluation Results

Table 5. Requirement Generation Results: SRM and Financial Compliance Domains

Domain	Dataset	Total Reqs	Baseline	Rule-Based	Trace Links	SCR	GER	TC	DR
SRM	DS1	22	22	0	22	1.00	0.0000	1.00	Yes
SRM	DS2	33	30	3	33	1.00	0.0909	1.00	Yes
SRM	DS3	41	38	3	41	1.00	0.0732	1.00	Yes
SRM	DS4	40	37	3	40	1.00	0.0750	1.00	Yes
SRM	DS5	56	49	7	56	1.00	0.1250	1.00	Yes
Financial	FDS1	11	11	0	11	1.00	0.0000	1.00	Yes
Financial	FDS2	15	13	2	15	1.00	0.1333	1.00	Yes
Financial	FDS3	16	15	1	16	1.00	0.0625	1.00	Yes
Financial	FDS4	16	14	2	16	1.00	0.1250	1.00	Yes
Financial	FDS5	49	42	7	49	1.00	0.1429	1.00	Yes

SCR achieves 1.00 across all five datasets, confirming that every structural ontology element extracted from the merged ontology is systematically reflected in at least one generated baseline requirement. This result is not a statistical tendency but a formal guarantee of the exhaustive mapping design confirmed empirically across all governance configurations.

TC achieves 1.00 across all five datasets, confirming that every generated requirement, whether derived from a structural ontology element or from governance rule activation, maintains an explicit and machine-recoverable traceability link to its semantic origin.

GER progresses from 0.0000 in DS1 to 0.0909 in DS2, reflecting the activation of three governance rules, RG_SLA_ASSOCIATION, RG_SLA_BREACH_RISK, and RG_PRIORITY_ASSIGNMENT. The GER values in DS3 (0.0732) and DS4 (0.0750) are slightly lower than DS2 (0.0909) despite retaining the same three activated rules. This reflects denominator growth: additional structural entities in DS3 and DS4 expand the total requirement count while the numerator remains constant at three rule-based requirements, confirming that governance enrichment responds to semantic content rather than structural size. GER reaches its maximum of 0.1250 in DS5, where all seven governance rules are simultaneously activated across compliance, audit, and escalation dimensions.

DR is confirmed through overall_exact_logical_match = true across ten independent executions per dataset, with SHA-256 hashes of all traceability CSV artifacts identical across every run, establishing the framework as a fully deterministic transformation system whose outputs are exclusively governed by semantic inputs and transformation logic.

SCR = 1.00 and TC = 1.00 across all five Financial Compliance datasets confirm that structural completeness and traceability by construction are domain-independent properties of the transformation design; they hold regardless of domain characteristics, entity structures, or governance patterns.

GER progresses from 0.0000 in FDS1 to 0.1429 in FDS5, the highest enrichment ratio across both domain sequences. FDS2 activates two governance rules covering KYC and AML compliance, producing GER = 0.1333. FDS3 activates one rule for credit risk assessment, producing GER = 0.0625, reflecting a smaller numerator over a growing baseline. FDS4 activates two rules covering approval workflow and audit event generation, producing GER = 0.1250. FDS5 simultaneously activates all seven financial governance rules covering KYC, AML, credit risk, approval workflow, audit event generation, regulatory reporting, and dispute escalation, producing GER = 0.1429, the highest value observed across both domain sequences (EDM Council, 2026).

DR = true is confirmed through overall_exact_logical_match = true across all FDS executions. The identical pipeline producing correct domain-specific outputs across both domains without code modification provides direct empirical evidence that domain-agnostic extensibility is an operational characteristic of the implemented framework rather than a theoretical design claim.

Qualitative Assessment of Representative Generated Requirements

To complement the quantitative metrics, a qualitative assessment examined three representative subsets - the 22 baseline requirements from DS1, the seven governance-derived requirements from DS5, and the seven financial governance requirements from FDS5 — against three criteria (Rodrigues da Silva, 2021): correctness of semantic content, structural conformance to recognisable SRS patterns using the modal verb "shall", and traceability interpretability. Baseline requirements produced through Rules R1–R3 consistently satisfied all three criteria, correctly reflecting their originating classes, properties, and relationships (e.g., "The system shall store and manage ServiceRequest information"). Governance-derived requirements demonstrated higher semantic specificity: the requirement generated by RG_SLA_BREACH_RISK correctly integrated a compound trigger condition involving two semantic elements, and the financial requirement generated by RF_KYC_VERIFICATION reflected the regulatory intent of the KYC dimension in a form recognisable to compliance practitioners. These observations confirm that generated requirements are structurally complete, traceable by construction, and linguistically recognisable to practitioners, though a formal human-centred evaluation remains a priority for future work.

Comparative Discussion

Comparison with Template-Based Approaches

The closest methodological counterpart is Darif et al. (2023), which applies controlled natural language templates through user-driven reformulation. However, its transformation is not formalised as a deterministic function: requirement reformulation and template selection depend on human judgment, limiting reproducibility, and domain models serve mainly for validation rather than as active generation drivers. The proposed framework differs in three respects: (1) inputs are formally structured (OWL ontologies, JSON conceptual models) rather than natural language; (2) the transformation S = F(M, D, R) is fully automated with no user intervention; and (3) governance integration is systematic, triggered by verifiable semantic patterns rather than manual inspection. These design choices allow the framework to achieve SCR = 1.00, TC = 1.00, and DR = true by construction — guarantees that template-based approaches cannot offer, since their reliance on human judgment at the reformulation stage inherently limits structural completeness, traceability, and reproducibility.

Comparison with Ontology-Based Approaches

Ontology-based approaches, represented here by Alsanad, Chikh and Mirza (2019) and Mosquera, Ruiz and Pastor (2025), provide strong semantic grounding but are fundamentally analytical rather than generative: they classify, validate, and trace existing requirements rather than deriving new ones from semantic representations. This is an architectural boundary, not an oversight — these approaches define what requirements mean without defining how they should be generated. The proposed framework bridges this boundary by operationalising the same type of semantic knowledge through the declarative rule-based activation mechanism, which uses governance constraints as generation triggers rather than purely for reasoning and validation. This is confirmed empirically through SCR = 1.00 across all ten datasets, demonstrating that ontological knowledge can be systematically transformed into complete SRS artifacts rather than used solely for post-hoc analysis.

Comparison with AI-Based Approaches

AI-based approaches (Shafikuzzaman et al., 2025; Korn, Gorsch and Vogelsang, 2025; Quattrocchi et al., 2025) demonstrate powerful generative capabilities but exhibit three structural limitations.

First, traceability is absent by design: LLM-based generation relies on autoregressive token prediction with no explicit symbolic link between output and source, unlike the proposed framework, where every requirement carries a traceability record (e.g., OWL:Class:X, OWL:GovernanceRule:RuleCode) at the moment of generation. Second, non-determinism is intrinsic to probabilistic inference; even at temperature = 0, outputs can vary across model versions and hardware, as confirmed by Beg, O'Donoghue and Monahan (2025) and Krishna et al. (2024), whereas the proposed framework's determinism is guaranteed by sorted extraction, fixed rule sequencing, and deterministic templates, verified via SHA-256 hash comparison across ten runs per dataset. Third, semantic grounding is implicit, captured through training data rather than formal ontologies. Rather than competing paradigms, the two approaches are complementary: AI-based methods suit early-stage elicitation and informal refinement, while the proposed framework provides formal, traceable, and reproducible specification at the transformation stage.

Contribution to the State of the Art

The framework contributes to the state of the art in four ways. First, the formal transformation model S = F(M, D, R) offers a level of analysability absent from all three reference paradigms, none of which define an explicit, verifiable transformation function. Second, the selective materialized merge mechanism addresses a previously unaddressed gap — integrating domain ontology elements proportionally to semantic relevance rather than wholesale — confirmed empirically by GER = 0.000 in all baseline datasets despite full domain ontology availability. Third, governance-aware enrichment through declarative rule activation extends generation beyond structural representation, with GER progressing systematically from 0.0000 to 0.1250 (SRM) and 0.1429 (Financial Compliance) as governance complexity increases. Fourth, domain-agnostic extensibility is empirically verified rather than theoretically claimed: the identical pipeline produced correct outputs across a purposefully constructed SRM ontology and the externally sourced FIBO standard ontology (EDM Council, 2026) without code modification. A criteria-based comparison, rather than a shared-metric benchmark, is used throughout since the three paradigms require fundamentally incompatible input formats, and forcing a common format would introduce information loss or artificial enrichment that would invalidate the comparison.

Limitations

The work is subject to several limitations. First, evaluation used controlled datasets — the SRM datasets were purposefully constructed, and the Financial Compliance datasets derived from the FIBO standard ontology (EDM Council, 2026) — rather than live industrial deployments; generalisation to large-scale ontologies with deep inheritance hierarchies and incomplete specifications requires further investigation. Second, the absence of a formal human-centred evaluation is the most significant limitation: the qualitative assessment presented earlier in this section offers initial evidence of semantic coherence but does not constitute a rigorous usability study involving domain experts. Third, domain-agnostic extensibility, while empirically supported across two domains, is bounded by four structural prerequisites: the domain ontology must be serialised in standard RDF/XML; class names must overlap between domain and system ontologies for the selective merge to identify relevant elements; governance rules must be expressible as object/datatype property conditions or compound AND conditions (excluding negation, quantification, or temporal reasoning); and the conceptual model must be a structured JSON artifact conforming to one of the two supported field-naming conventions. Fourth, the declarative rule-based activation mechanism evaluates rules against static ontology snapshots and does not support forward chaining, conflict resolution, or dynamic rule interaction, limiting its applicability to governance scenarios requiring multi-step reasoning. Fifth, the transformation algorithm operates in approximately $O(n + n \cdot r)$ time ($n = |E\_structural|$, $r = |R|$); while this exhibits near-linear growth under bounded governance-rule conditions, empirical performance profiling on industrial-scale ontologies (n in the thousands) remains an open question.

4. CONCLUSION

This paper presented a deterministic ontology-driven transformation framework for automated SRS generation, formalising the process as S = F(M, D, R) and treating determinism, traceability, and governance integration as first-class architectural properties. Three technical mechanisms support this: selective materialized merge, which proportionally incorporates domain ontology elements and prevents spurious rule activation (confirmed by GER = 0.000 in all baseline datasets); declarative rule-based activation, which generates governance-aware requirements from substitutable rule catalogues without pipeline modification; and traceability by construction, linking every requirement to its originating element or rule at the moment of generation.

Empirical evaluation across two domains confirmed all four formal properties: SCR = 1.00 and TC = 1.00 across all ten datasets, GER progressing systematically to 0.1250 (SRM) and 0.1429 (Financial Compliance), and DR = true confirmed via SHA-256 hash verification across ten executions per dataset. The consistency of results across a purposefully constructed SRM ontology and the independently developed FIBO standard ontology provides strong evidence that these properties are architectural characteristics of the design rather than domain-specific outcomes. Comparative analysis against template-based (Darif et al., 2023), ontology-based (Alsanad, Chikh and Mirza, 2019; Mosquera, Ruiz and Pastor, 2025), and AI-based (Korn, Gorsch and Vogelsang, 2025; Quattrocchi et al., 2025; Shafikuzzaman et al., 2025) approaches confirmed that the framework addresses limitations present across all three paradigms, occupying a complementary position to AI-based methods rather than competing with them.

The broader significance lies in demonstrating that structured semantic representations can function as active drivers of formal requirement synthesis rather than merely descriptive artifacts, advancing requirements engineering as a transformation discipline. Future work will extend the framework toward large-scale industrial ontologies, human-in-the-loop validation, more expressive governance reasoning (forward chaining, conflict resolution), further domain-agnostic validation (e.g., HL7 FHIR, ArchiMate), and integration within end-to-end model-driven development toolchains.

Declarations

CRediT authorship contribution statement: Alaa Abd Elazheim: Conceptualization, Methodology, Software, Investigation, Data curation, Formal analysis, Writing – original draft. Farid Meziane: Supervision, Methodology, Writing – review & editing. Omayma Husain: Conceptualization, Supervision, Validation, Project administration, Writing – review & editing.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process: During the preparation of this work, the author(s) used Claude (Anthropic) to assist with editing and reviewing the manuscript text. Subsequent to the use of this tool, the author(s) reviewed and edited the content as necessary and take full responsibility for the content of the publication.

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.

Funding statement: This research received no external funding.

Data availability: The experimental datasets and code supporting the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] Aberkane, A.-J., Poels, G., & Vanden Broucke, S. (2021). Exploring automated GDPR-compliance in requirements engineering: A systematic mapping study. *IEEE Access, 9*, 66542–66556. https://doi.org/10.1109/ACCESS.2021.3076921

[2] Alrumaih, H., Mirza, A., & Alsalamah, H. (2020). Domain ontology for requirements classification in requirements engineering context. *IEEE Access, 8*, 110300–110312. https://doi.org/10.1109/ACCESS.2020.2993838

[3] Alsanad, A. A., Chikh, A., & Mirza, A. (2019). A domain ontology for software requirements change management in global software development environment. *IEEE Access, 7*, 49352–49361. https://doi.org/10.1109/ACCESS.2019.2909839

[4] Beg, A., O'Donoghue, D., & Monahan, R. (2025). Formalising software requirements with large language models. *arXiv*. https://doi.org/10.48550/arXiv.2506.10704

[5] Darif, I., Politowski, C., El Boussaidi, G., Benzarti, I., & Kpodjedo, S. (2023). A model-driven and template-based approach for requirements specification. In Proceedings of the 26th ACM/IEEE

International Conference on Model Driven Engineering Languages and Systems (MODELS). IEEE. https://doi.org/10.1109/MODELS58315.2023.00018
[6] EDM Council. (2026). *Financial Industry Business Ontology (FIBO): Financial compliance module*. https://spec.edmcouncil.org/fibo/
[7] Fariha, A., Alwidian, S., & Azim, A. (2024). A systematic literature review on requirements engineering and maintenance for embedded software. *IEEE Access, 12*, 114263–114276. https://doi.org/10.1109/ACCESS.2024.3443271
[8] Gillani, M., Niaz, H. A., & Ullah, A. (2022). Integration of software architecture in requirements elicitation for rapid software development. *IEEE Access, 10*, 56158–56173. https://doi.org/10.1109/ACCESS.2022.3177659
[9] Gupta, A., Poels, G., & Bera, P. (2022). Using conceptual models in agile software development: A possible solution to requirements engineering challenges in agile projects. *IEEE Access, 10*, 128716–128742. https://doi.org/10.1109/ACCESS.2022.3221428
[10] Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2021). Systematic mapping study on security approaches in secure software engineering. IEEE Access, 9, 19139–19160. https://doi.org/10.1109/ACCESS.2021.3052311
[11] Korn, A., Gorsch, S., & Vogelsang, A. (2025). LLMREI: Automating requirements elicitation interviews with LLMs. *arXiv*. https://arxiv.org/abs/2507.02564
[12] Krishna, M., Gaur, B., Verma, A., & Jalote, P. (2024). Using LLMs in software requirements specifications: An empirical evaluation. arXiv. https://doi.org/10.48550/arXiv.2404.17842
[13] Mokos, K., Nestoridis, T., Katsaros, P., & Bassiliades, N. (2022). Semantic modeling and analysis of natural language system requirements. IEEE Access, 10, 84094–84109. https://doi.org/10.1109/ACCESS.2022.3197281
[14] Mosquera, D., Ruiz, M., & Pastor, O. (2025). Ontology-based NLP tool for tracing software requirements and conceptual models: An empirical study. *Requirements Engineering, 30*, 341–369. https://doi.org/10.1007/s00766-025-00447-4
[15] Quattrocchi, G., Pasquale, L., Spoletini, P., & Baresi, L. (2025). Can LLMs generate user stories and assess their quality? IEEE Transactions on Software Engineering. Advance online publication.
[16] Rodrigues da Silva, A. (2021). Linguistic patterns, styles, and guidelines for writing requirements specifications: Focus on use cases and scenarios. *IEEE Access, 9*, 143506–143530. https://doi.org/10.1109/ACCESS.2021.3120004
[17] Schulze, S., & Pretorius, L. (2013). An exploratory case study on the requirements business processes of a typical South African high technology systems engineering company. *SAIEE Africa Research Journal, 104*(1), 11–21.
[18] Shafikuzzaman, M., Islam, M. R., Zaman, S., Ma, A., & Sifat, A. I. (2025). On the effectiveness of zero-shot and few-shot pretrained language models for software requirement classification. IEEE Access, 13, 159439–159448. https://doi.org/10.1109/ACCESS.2025.3607813
[19] Voria, G., Casillo, F., Gravino, C., Catolino, G., & Palomba, F. (2025). RECOVER: Toward requirements generation from stakeholders' conversations. IEEE Transactions on Software Engineering. Advance online publication.