

Does Concept-Structured Text Decomposition Improve LLM-enabled Domain Modeling? An Exploratory Study of Its Decomposition Patterns

Zixin Wu^{1,*}

¹University of Duisburg-Essen, Forsthausweg 2, 47057 Duisburg, Germany

Abstract

Domain modeling from natural-language descriptions is expertise-intensive, and large language models (LLMs) automate it only partially, producing output that is syntactically valid but weakens on relationships and structural constraints as the domain grows. Recent work decomposes the modeling task, yet along axes that do not come from the domain, either the element types of the target notation or the paragraphs of the source document. This doctoral research investigates a third axis, *concept-structured decomposition*, which organizes the source by the domain concepts it describes and the relationships among them, approximated by an extracted knowledge graph (KG). It proceeds in two phases. The first is exploratory and asks whether a structure taken from the domain's concepts guides LLM-enabled modeling more reliably than one taken from the target notation or the document layout. It is instantiated as KG-Trav-DM (KG-guided concept traversal for domain modeling), in three stages: KG extraction with entity canonicalization and sentence grounding, incremental synthesis integrating one KG entity and its grounding sentences per hop, and global refinement. The traversal order is itself a design variable, and the alternatives form a family of concept-structured decomposition patterns. The second phase is design-oriented and contingent on the first. Only if conceptual structure proves effective is it worth designing around the concept-level steps it exposes, each traceable to its source sentences, as points for a modeler to intervene. A preliminary study on UML class diagram tasks, reported here, indicates that the axis merits further investigation and that the traversal pattern affects the outcome. The research therefore aims to establish whether decomposing by conceptual structure guides LLM-enabled modeling more reliably than the target-notation and document-layout alternatives, under which traversal patterns it does so, and ultimately where a modeler can most sensibly enter the collaboration.

Keywords

LLM-enabled Domain Modeling, Knowledge Graphs, Large Language Models, UML Class Diagram, Human-AI Collaboration

1. Introduction

Domain modeling captures a domain's entities, relationships, and constraints in a structured form, and plays an essential role in software engineering [1] and information systems engineering [2]. The output, typically a UML [3] class diagram or an entity-relationship diagram, anchors subsequent design decisions and serves as a primary artifact through which modelers and domain experts negotiate a shared understanding. Producing it from informal source material is iterative and expertise-intensive [4], and the result is judged against criteria such as completeness, correctness, minimality, and understandability [5]. Because downstream artifacts such as database schemas and test scaffolding rest on this model, domain modeling is both demanding to perform and disproportionately costly to get wrong.

This labor and expertise burden has motivated a long-standing interest in automation, from rule-based extraction over controlled natural language and ontology learning pipelines [6, 7] to more recent AI-assisted modeling tools [8]. Large language models (LLMs) have made it technically possible to generate a domain model directly from a natural-language description in a single prompt. The quality of these outputs, however, remains uneven. Empirical evaluations of LLM-generated UML class

ER2026: Companion Proceedings of the 45th International Conference on Conceptual Modeling: Doctoral Consortium, Tutorials, Posters and Demos, October 5-8, 2026, St. John's, NL, Canada.

*Corresponding author.

✉ zixin.wu@uni-due.de (Z. Wu)

ORCID 0009-0006-4383-3127 (Z. Wu)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

diagrams [9, 10, 11] consistently report that LLMs handle class and attribute identification adequately while struggling on relationships and structural constraints between classes.

This unreliability is not a uniform “noise floor” that scale alone will lift [12]. Once an erroneous element enters a generated model, downstream elements are constructed in reference to it, mirroring the cascading errors documented for free-form generation [13]. Beyond cascades, structural inconsistencies in cardinalities, generalizations, and relationships disproportionately damage correctness and understandability, dimensions that distinguish a usable conceptual model from a merely syntactically valid one. A second difficulty compounds the first. The output carries no trace of which decisions rest on textual evidence, complicating audit and revision. Both difficulties share one setting, holistic single-prompt synthesis, in which the full source is presented at once and no intermediate state is inspectable. Recent work has accordingly begun to decompose the generation process.

Two recent approaches illustrate this reframing, each decomposing along a distinct axis. Silva et al. [14] decompose along the meta-structure of the target notation, processing each UML element type in a separate step. Reinhartz-Berger et al. [15] decompose along the surface structure of the input text, contrasting whole-text with paragraph-by-paragraph processing. Both demonstrate their methods on UML class diagrams, and both show that how the task is broken up affects the quality of the result. Neither axis is drawn from the domain itself, since element types belong to the target notation and paragraphs to the source document. To our knowledge, a decomposition guided by the domain’s conceptual structure has not been previously explored.

This doctoral research proposes a further decomposition axis, organizing the source text by extracted concepts, the domain entities named in the source, rather than by output element types or surface paragraphs. A knowledge graph (KG) is extracted from the source as a preliminary step, with each KG entity defining one unit of integration. The source sentences grounding the triples in which that entity appears as source form the evidence visible at that step, and KG centrality yields a natural integration order from core concepts outward. We refer to the resulting decomposition as concept-structured decomposition, in contrast to the element-type and input-granularity decompositions in the literature.

Whether this axis is the better choice is an empirical question: *can the conceptual structure of the domain a source text describes guide an LLM through domain modeling more reliably than a structure imposed by the target notation or by the document layout, and if so, under which traversal patterns?* The research approaches it in two phases. The first phase is exploratory and is the subject of this paper, which instantiates the axis as KG-Trav-DM (KG-guided concept traversal for domain modeling) and compares alternative traversals over the KG as a family of decomposition patterns. Establishing why some patterns work better than others belongs to the same phase.

The second phase is design-oriented and contingent on the first. Because the integration unit is a single concept entity, every intermediate state is a partial domain model open to inspection, and the synthesized artifact carries a trace from each element back to the sentences and KG fragment that produced it. A human-in-the-loop design could build on these properties to address the second difficulty raised above. Building it, however, is warranted only once the axis itself has been shown to work. This paper therefore states the research questions for both phases and reports the current state of the first: the framework, a preliminary study on UML class diagram tasks, and a research plan.

2. Related Work

LLM-Enabled Domain Modeling. Early experience reports found that LLMs produce syntactically valid UML diagrams from natural-language input [9], with semantic correctness apparently degrading once models exceed roughly ten classes [16]. Targeted studies have since quantified the gap between identifying entities and relating them. Wang et al. [10] report relationship-extraction accuracies substantially below entity-identification accuracies for class and use-case diagrams, and Calamo et al. [11] confirm the gap persists across LLM backbones and prompting strategies.

Recent work has begun to address this reliability gap by decomposing the modeling task into multiple steps rather than producing models in a single prompt. The most widely explored decomposition

principle is the element-type axis, which advances by producing one kind of UML construct per step across the entire model. Silva et al. apply a ToT framework [14] that processes classes first, then relationships of various kinds, then association classes. Chen et al. [17] adopt question decomposition with separate sub-tasks for classes, for associations and aggregations, and for inheritance. Yang et al. [18] interleave class-and-attribute generation with a self-reflection step before tackling relationships. A different principle, the input-granularity axis, has been investigated by Reinhartz-Berger et al. [15], who study how whole-text versus paragraph-by-paragraph presentation and different prompting strategies affect output quality. Both axes operate on structures taken from the target notation or the document layout rather than from the domain’s concepts. In contrast to both, this research investigates a third axis, decomposing the modeling task by the conceptual structure of the source content rather than by the target notation or the document layout.

A complementary line introduces human supervision into LLM-driven modeling. Silva et al. propose a ToT-Q framework [19] in which a rule-based agent selects questions about uncertain aspects of the model and engages the modeler in dialogue. Pre-LLM AI-assisted modeling explored similar refinement framings [8]. The second phase of this research belongs to the same line, but approaches it from the decomposition side, asking whether concept-structured steps, each tied to identifiable source sentences, offer better-localized points and time for a modeler to intervene.

LLM Limitations Potentially Relevant to Domain Modeling. Several documented limitations of generation by LLMs may affect LLM-driven domain modeling, though their precise manifestation in modeling tasks remains conjectural. LLMs degrade super-linearly with compositional depth [12], hallucinated content compounds across steps when prior outputs feed subsequent context [13], and LLMs are sensitive to irrelevant context even when that content is itself correct [20]. Domain modeling exhibits all of these conditions. It is compositional in nature, and depending on how the task is decomposed, the LLM may face distractor content at each step, the burden of repeatedly processing the full source, or redundant modeling of the same elements across iterations.

Mitigation techniques from general LLM reasoning have begun to be adapted to structured generation tasks. Sequential decomposition strategies such as least-to-most prompting [21] break a task into ordered subproblems, and self-consistency [22], building on chain-of-thought (CoT) reasoning [23], improves reliability through repeated sampling combined with consensus selection. KG-Trav-DM adopts both principles in the conceptual modeling setting. The traversal fixes an explicit order over subproblems, and each hop samples several candidate integrations and selects among them by consensus.

Knowledge Graph Construction from Text. Extracting structured knowledge from text has moved from pipelined entity recognition, relation extraction, and canonicalization [24], through neural multi-stage architectures built on pretrained language models [25], to LLMs used as the extraction engine, producing typed (subject, predicate, object) triples directly [26]. Our pipeline uses an extractor of the last kind and additionally grounds each triple to its supporting source sentence, treating the resulting graph as an intermediate structure that organizes the subsequent modeling task.

3. Research Questions and Objectives

This doctoral research is structured into two main phases. The first asks whether decomposing along the conceptual structure of the source, and along particular orderings within that structure, helps an LLM produce better domain models. The second builds on the first and asks how a modeler can be brought into such a process at the right moments, with a traceable record of what each step rested on.

3.1. Phase 1: Assessing Concept-Structured Decomposition

Neither axis reviewed in Section 2 reflects how the domain’s concepts relate to one another, and a source text is under no obligation to introduce those concepts in the order or the grouping their relationships

would suggest. A single paragraph may touch several unrelated concepts, and one concept may be described across passages far apart. Organizing the task by concepts is expected to give the LLM a more focused signal at each step, since the evidence presented is the evidence bearing on one concept, and a more continuous signal across steps, since consecutive steps concern concepts that are actually related.

Whether this expectation holds has to be settled against the alternatives rather than in isolation. The objective of this phase is to place concept-structured decomposition alongside holistic synthesis, which decomposes nothing and varies only in prompting technique [27], element-type decomposition [14], and paragraph-level decomposition [15], and to determine whether it reaches comparable or better quality and where it gains and loses.

The knowledge graph extracted from the source is what makes the decomposition concrete, and it opens a second line of inquiry. Once concepts and their relationships are explicit, the order in which they are integrated becomes a variable that can be set deliberately, rather than being inherited from whatever order the text happens to use. If quality varies systematically with the ordering, that variation is itself evidence about which aspects of conceptual structure the LLM is exploiting.

Two research questions follow for this phase.

- **RQ1:** Does decomposing the modeling task by the conceptual structure of the source yield higher-quality UML class diagrams than holistic single-prompt synthesis or other decompositions?
- **RQ2:** How does the traversal ordering over the KG affect the quality of the generated model, and do the differences between orderings admit a structural explanation?

3.2. Phase 2: Towards Traceable Human Involvement

The second phase depends on the outcome of the first, though not on its being decisively favorable. If concept-structured decomposition proves advantageous, or merely matches the existing axes on quality, each step still yields a partial model attached to the concept and the source sentences that produced it. This phase would build on that record as one basis for involving the modeler during assembly. A common arrangement places the modeler after the fact, revising or answering questions about a model already generated in full. A decomposition that pauses at each concept admits a different arrangement, in which the modeler can act while the model is still being assembled and can see what each element was derived from. Whether such an arrangement is genuinely better for the modeler, and whether it yields better models than intervening at the end, are separate questions and are treated as such.

- **RQ3:** Do the concept-level intermediate states, and the evidence links they carry, give a modeler better-localized opportunities to intervene than review of a completed initial model?
- **RQ4:** How does model quality change under intervention at these intermediate states, compared with post-hoc revision of a completed model?

4. Current Framework Design

KG-Trav-DM realizes concept-structured decomposition as the three-stage pipeline shown in Figure 1. Stage 1 (Section 4.1) extracts a domain-relevant conceptual structure from the source text, canonicalizing entity names and grounding each triple in the sentence that licensed it. Stage 2 (Section 4.2) uses that structure to control the order of integration and the evidence exposed at each step. Stage 3 (Section 4.2) recovers global consistency and any content the earlier stages missed. The design currently serves the first phase alone, making concept-structured decomposition operational and its orderings comparable as **RQ1** and **RQ2** require. The intermediate states it produces are nevertheless the material the second phase would build on.

4.1. Knowledge Graph Extraction

We adopt a KG as the conceptual structure on which downstream synthesis operates, extracting it with KGen [26], an LLM-based extractor. Two of its properties matter here. It operates without per-domain

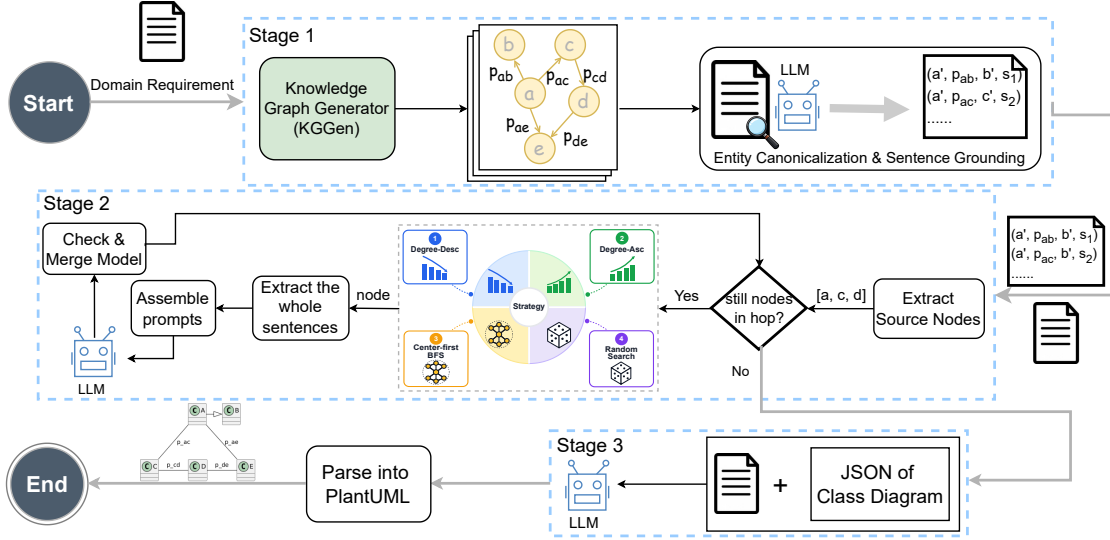


Figure 1: Overview of the KG-Trav-DM (KG-guided concept traversal for domain modeling) framework.

tuning, so the same configuration applies across all source descriptions, matching the domain-agnostic stance the rest of the framework takes. It also produces typed (subject, predicate, object) triples directly from text, without an intermediate logical form or a predefined schema.

Two things its output lacks are a canonicalized entity vocabulary and source attribution at the sentence level. The same domain entity may appear under several surface forms across raw triples, which would inflate the hop count and fragment an entity’s neighborhood across nominally distinct nodes. The absence of sentence attribution is more consequential, since per-hop evidence retrieval depends on it. A single LLM-based post-processing pass addresses both. Presented with the source text and the raw triples, the LLM performs entity canonicalization, mapping the surface forms of one concept to a single name, and sentence grounding, attaching to each triple the sentence that licenses it and dropping as unsupported any triple it cannot align to a sentence. Each retained triple is thereby extended to a quadruple (s, p, o, σ) , where $s, o \in V$ are canonicalized entities, p is a predicate label, and σ identifies the licensing sentence, and we write the augmented KG as $G = (V, E)$. The augmented KG then serves a double role downstream. Through its entity set and edges it is a structural index over the domain, and through its per-triple sentence pointers it is an evidence index over the source text. The second role is what later makes each generated element traceable to the text behind it.

4.2. Concept-Structured Incremental Synthesis

This stage processes KG entities one at a time, integrating each concept into a partial domain model that grows across hops. The hop queue contains those entities that are the source of at least one extracted triple. Entities that appear only as targets surface implicitly through the relations owned by their source’s hop. Figure 2 walks through the process on a concrete example.

Traversal Strategies. The framework adopts *degree-descending* order as its default. Let $V_H = \{s \in V : \exists (s, p, o, \sigma) \in E\}$ denote the hop-entity set. For each $e \in V_H$, the *hop-induced degree* $\deg_H(e)$ is the number of distinct other hop entities adjacent to e via at least one triple:

$$\deg_H(e) = |\{e' \in V_H \setminus \{e\} : \exists (s, p, o, \sigma) \in E, \{s, o\} = \{e, e'\}\}|.$$

Degree-descending visits hop entities in decreasing order of $\deg_H$, with alphabetical tie-breaking for determinism. Three alternatives are also supported. *Degree-ascending* reverses this order. *Center-first BFS* runs BFS over the hop subgraph from the highest- $\deg_H$ entity, seeding any disconnected component

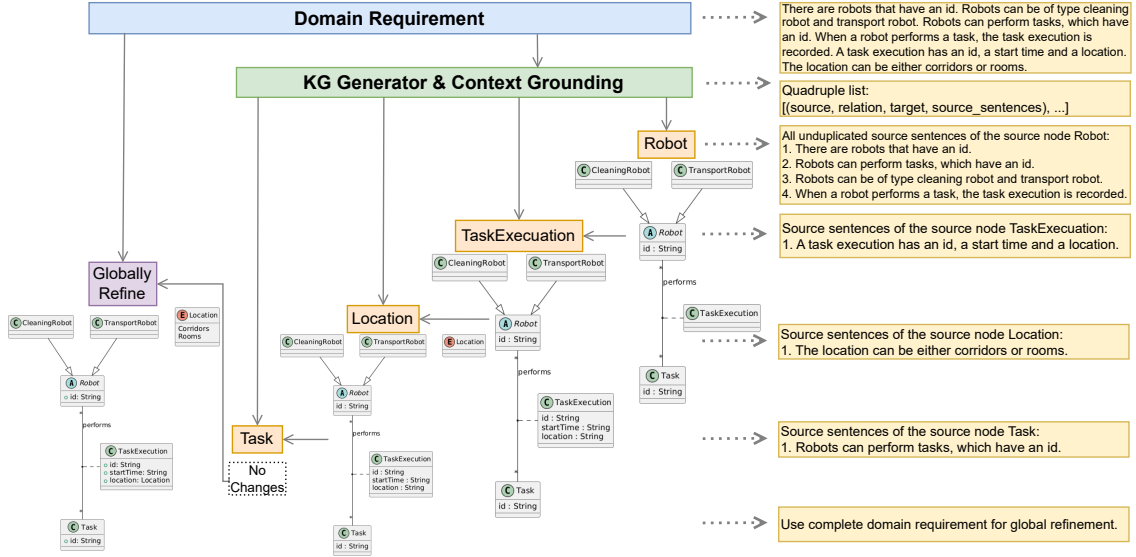


Figure 2: Concept-structured incremental synthesis on a Robots-and-Tasks example under degree-descending traversal. At each hop, the KG triples and source sentences grounded to the current entity drive the extension of the partial UML model, with global refinement applied to the completed model.

from its own highest- $\deg_H$ unvisited entity. *Random* produces a permutation under a fixed seed. These four are the concept-structured decomposition patterns compared in this work, where a pattern is a principle for ordering the domain concepts a source describes, and the KG is the approximation of those concepts that the ordering actually operates on.

The default is motivated by two considerations. First, $\deg_H$ serves as a proxy for structural centrality, since entities connected to many other modeled concepts are typically the structural concepts of the domain. Second, integrating high-degree entities first lets subsequent hops reuse already-present nodes as relation targets, rather than introducing both endpoints together with the relation itself, which is intended to lower the compositional burden of each hop in line with the LLM weakness discussed in Section 2. Whether either consideration translates into measurable quality differences is what RQ2 asks, which is why the alternatives are supported alongside the default.

Per-Hop Generation. At each hop the LLM receives a focused subset of source content rather than the full text. The choice of evidence is mediated through the KG, so that the source sentences linked to the current entity’s outgoing triples constitute the entire textual context, alongside the partial model. Restricting exposure in this way is intended to mitigate the distractor effects discussed in Section 2. Ownership by the source endpoint also ensures that each KG edge surfaces at exactly one hop, which avoids modeling the same edge twice from opposite ends.

At each hop we sample K candidate integrations at non-trivial temperature and let M voter agents select a winner by majority vote, applying self-consistency [22] at the hop level rather than at the level of the final output. The placement matters in incremental synthesis, where a single bad hop propagates into all subsequent ones. Sampling noise is averaged within the hop instead of accumulating across hops, and content the LLM produces inconsistently across candidates, often a signature of hallucination, is less likely to win the vote. The winner is then merged into the partial model, which the next hop takes as its starting point.

Global Refinement. After incremental synthesis, a single global refinement pass produces the final model. The LLM receives the synthesized model alongside the full source text and is asked to refine rather than re-synthesize it, by identifying inconsistencies, completing missing elements, and consolidating redundancies. As in incremental synthesis, K candidates are sampled and M voters select

Table 1

Performance on UML class diagram generation, pooled over the five tasks. Each cell reports P/R/F1. The best F1 per backbone and category in bold. Traversals: Asc = degree-ascending, Desc = degree-descending, BFS = center-first BFS.

Model	Method	Traversal	Class	Attribute	Relationship	Assoc. Class
GPT-4o	0-shot	—	0.914/0.914/0.914	0.861/0.775/0.816	0.400/0.500/0.444	0.500/0.333/0.400
	ToT	—	0.800/0.914/0.853	0.586/0.850/0.694	0.400/0.583/0.475	0.600/0.667/0.632
	Ours	Asc	0.889/0.914/0.901	0.816/0.775/0.795	0.533/0.667/0.593	0.833/0.556/0.667
		Desc	1.000/0.971/ 0.986	0.872/0.850/ 0.861	0.630/0.708/ 0.667	1.000/0.667/ 0.800
		BFS	0.833/0.857/0.845	0.756/0.775/0.765	0.448/0.542/0.491	0.714/0.556/0.625
		Random	0.892/0.943/0.917	0.861/0.775/0.816	0.472/0.708/0.567	1.000/0.444/0.615
GPT-4o-mini	0-shot	—	0.833/0.714/0.769	0.742/0.575/0.648	0.360/0.375/0.367	1.000/0.111/0.200
	ToT	—	0.775/0.886/0.827	0.452/0.825/0.584	0.353/0.500/ 0.414	0.500/0.333/0.400
	Ours	Asc	0.784/0.725/0.753	0.617/0.725/0.667	0.214/0.375/0.273	1.000/0.333/ 0.500
		Desc	0.861/0.886/ 0.873	0.651/0.700/ 0.675	0.270/0.417/0.328	0.600/0.333/0.429
		BFS	0.811/0.857/0.833	0.500/0.700/0.583	0.158/0.250/0.194	0.000/0.000/0.000
		Random	0.838/0.886/0.861	0.548/0.767/0.639	0.222/0.348/0.271	0.750/0.333/0.462

a winner. The pass plays two roles. KG extraction is imperfect, so source information not captured as triples can still be recovered once the LLM sees the full text again. In the limiting case of an empty KG, which has not arisen in the cases examined so far, the pass degrades to single-prompt synthesis from the source. Even when the KG is complete, a global view lets the LLM address features that are hard to reason about within any single hop, such as cardinality consistency across clusters or generalization hierarchies spanning several hops.

5. Preliminary Evaluation

The preliminary study reported here provides first observations on **RQ1** and **RQ2**. The KG is extracted with KGGen [26] using GPT-4o at sampling temperature 0.3. Extracted KGs are cached once per task and reused across all subsequent runs, so that KG-extraction variance does not enter the traversal comparison. Incremental synthesis and global refinement are run under two backbones of different capacity, GPT-4o and GPT-4o-mini. Per-hop generation uses $K = 3$ candidates and $M = 5$ voters, with generation temperature 1.0 and voting temperature 0.5. Each configuration was executed once.

We use the five domain descriptions and reference models of the ToT framework [14], comprising 35 classes, 40 attributes, 24 relationships, and 9 association classes across the reference set. The author scored each generated model manually against its reference, matching elements by semantic consistency rather than strict name equality and following the ToT evaluation protocol so that the numbers remain directly comparable with those reported in [14]. However, paragraph-level decomposition [15] is not included, because the published figures cannot be compared directly on this dataset, no open-source implementation is available to reproduce, and the manual evaluation is demanding. Table 1 reports pooled figures. True positives, false positives, and false negatives are summed over the five tasks, and precision, recall, and F1 are computed once per element category from those totals.

A first observation on **RQ1** comes from the two baselines under GPT-4o, which show a clear division of strengths in Table 1. Considering F1, the zero-shot baseline leads on classes (0.914 vs 0.853) and attributes (0.816 vs 0.694), while ToT contributes mainly on relationships (0.475 vs 0.444) and association classes (0.632 vs 0.400), the categories that decomposition by element type is designed to address. Concept-structured decomposition with degree-descending traversal combines both strengths, reaching the highest F1 in all four categories. It preserves the class and attribute coverage of the zero-shot baseline while raising relationships and association classes beyond what ToT achieves. The largest gains fall on relationships and association classes, the documented weak spots of Section 2, which is where a concept-oriented axis would be expected to help if the reasoning behind it holds.

Under GPT-4o-mini every configuration degrades and the decompositions separate less clearly. ToT now leads the zero-shot baseline in three of four categories, and concept-structured decomposition still reaches the best F1 on classes, attributes, and association classes, but ToT is best on relationships (0.414 vs 0.328). Decomposition can redistribute the modeling work, but it cannot supply capability the backbone lacks, and where the smaller model does not recognize a relationship at all, no arrangement of the evidence recovers it. The weaker backbone therefore narrows the range within which any decomposition could differ, which leaves **RQ1** less observable at this capability level rather than answered in the negative.

Turning to **RQ2**, degree-descending leads the alternative orderings in all four categories under GPT-4o, most visibly over degree-ascending on association classes and over random on classes. Degree-ascending and center-first BFS are consistently weaker. This pattern of results is consistent with the rationale given in Section 4.2, that high-degree entities integrated first stabilize the partial model and lower the compositional load of later hops, but consistency with a rationale is not confirmation of it, and the mechanism remains a conjecture that these five tasks cannot separate from alternative explanations.

Under GPT-4o-mini the differences between orderings compress in the same way. Degree-descending still leads on classes, attributes, and relationships, but by smaller margins, while degree-ascending leads on association classes and center-first BFS fails entirely there ($F1 = 0$). If ordering works by lowering the compositional load of each hop, its benefit is bounded by how much of that load the backbone could carry unaided, and a backbone that already fails on simpler steps leaves little for the ordering to protect. Whether the narrowed gaps reflect that bound or the noise of five single runs cannot be settled here.

6. Research Plan and Expected Contributions

For the first phase, Section 5 reports a preliminary evaluation on five small cases only, from which no solid conclusion can be drawn, so the dataset has to be extended, for example with the collection provided by the Golden UML Modelset¹ [28]. Baselines are also missing, among them the paragraph-by-paragraph decomposition of Reinhartz-Berger et al. [15], and the backbones should cover as many state-of-the-art open-source LLMs as possible rather than only the outdated GPT-4o series. Baselines alone cannot separate the mechanisms the approach combines, so ablations are needed as well, isolating individual components such as the evidence selection, the traversal order, the voting, and the global refinement. With a dataset of that size and that many baselines, manual evaluation, performed by the author here, is no longer easy and may also carry bias, so a degree of automated evaluation has to be introduced. Even if it cannot be perfect, a uniform criterion allows all baselines to be reproduced and scored on the same standard, which makes the conclusions more solid. Additionally, The extracted KG has to be evaluated in its own right, since its quality may bound what the decomposition can achieve, and the compute budget each configuration consumes should be reported alongside quality, as the decomposed methods issue considerably more LLM calls than single-prompt synthesis.

Once the exploratory work of the first phase is complete, the second phase asks, on the basis of its results, how the modeler can be brought into the process to further improve the reliability of LLM-enabled domain modeling, and then extends the design into the complete framework. Several factors have to be considered here, among them when the modeler intervenes and in what form. Redundant intervention adds to the modeler’s burden, so intervention should be placed where the source text is not clear enough or where the step lies at the boundary of the LLM’s capability, which is where human-AI collaboration can be more effective. A traceable incremental modeling process also makes the construction of the model more transparent. Ultimately, two contributions are expected:

- The first is to explore a way of decomposing a requirements document along the structure of the domain’s concepts, with a systematic evaluation of its effect and a discussion of the advantages it may bring.
- The second, contingent on the first, is a complete framework built on that decomposition, in which the modeler intervenes at the appropriate moments and the process stays traceable.

¹<https://golden-uml-modelset.vercel.app/>

Acknowledgments

I sincerely thank my supervisor, Prof. Dr. Janis Voigtländer, at the University of Duisburg-Essen, for his guidance and support, and the reviewers of the ER 2026 main conference for their comments.

Declaration on Generative AI

During the preparation of this work, the author used Claude Opus 5 for grammar and spelling check to catch errors that might have been missed, for text translation to reach a broader audience. After using these services, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] N. Iscoe, G. B. Williams, G. Arango, Domain modeling for software engineering, in: [1991 Proceedings] 13th International Conference on Software Engineering, IEEE, 1991, pp. 340–343. doi:10.1109/ICSE.1991.130660.
- [2] M. Snoeck, Enterprise information systems engineering, The MERODE Approach (2014). doi:10.1007/978-3-319-10145-3.
- [3] Object Management Group (OMG), Unified modeling language (UML) version 2.5.1, 2017. URL: <https://www.omg.org/spec/UML/2.5.1>.
- [4] P. J. Frederiks, T. P. Van der Weide, Information modeling: The process and the required competencies of its participants, Data & Knowledge Engineering 58 (2006) 4–20. doi:10.1016/j.datak.2005.05.007.
- [5] C. Batini, M. Lenzerini, S. B. Navathe, A comparative analysis of methodologies for database schema integration, ACM computing surveys 18 (1986) 323–364. doi:10.1145/27633.27634.
- [6] C. Arora, M. Sabetzadeh, L. Briand, F. Zimmer, Extracting domain models from natural-language requirements: approach and industrial evaluation, in: Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems, 2016, pp. 250–260. doi:10.1145/2976767.2976769.
- [7] R. Saini, G. Mussbacher, J. L. Guo, J. Kienzle, A neural network based approach to domain modelling relationships and patterns recognition, in: 2020 IEEE Tenth International Model-Driven Requirements Engineering, IEEE, 2020, pp. 78–82. doi:10.1109/MoDRE51215.2020.00016.
- [8] C. Feltus, Q. Ma, H. A. Proper, P. Kelsen, Towards AI assisted domain modeling, in: International Conference on Conceptual Modeling, Springer, 2021, pp. 75–89. doi:10.1007/978-3-030-88358-4_7.
- [9] H.-G. Fill, P. Fettke, J. Köpke, Conceptual modeling and large language models: impressions from first experiments with ChatGPT, Enterprise Modelling and Information Systems Architectures 18 (2023) 1–15. doi:10.18417/emisa.18.3.
- [10] B. Wang, C. Wang, P. Liang, B. Li, C. Zeng, How LLMs aid in UML modeling: An exploratory study with novice analysts, in: 2024 IEEE International Conference on Software Services Engineering, 2024, pp. 249–257. doi:10.1109/SSE62657.2024.00046.
- [11] M. Calamo, M. Mecella, M. Snoeck, Assessing the suitability of large language models in generating UML class diagrams as conceptual models, in: International Conference on Business Process Modeling, Development and Support, Springer, 2025, pp. 211–226. doi:10.1007/978-3-031-95397-2_13.
- [12] N. Dziri, X. Lu, M. Sclar, X. L. Li, L. Jiang, B. Y. Lin, S. Welleck, P. West, C. Bhagavatula, R. L. Bras, J. D. Hwang, S. Sanyal, X. Ren, A. Ettinger, Z. Harchaoui, Y. Choi, Faith and fate: Limits of transformers on compositionality, in: Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL: <https://openreview.net/forum?id=Fkckkr3ya8>.

- [13] M. Zhang, O. Press, W. Merrill, A. Liu, N. A. Smith, How language model hallucinations can snowball, in: Proceedings of the 41st International Conference on Machine Learning, 2024. URL: <https://dl.acm.org/doi/10.5555/3692070.3694535>.
- [14] J. Silva, Q. Ma, J. Cabot, P. Kelsen, H. A. Proper, Application of the tree-of-thoughts framework to LLM-enabled domain modeling, in: International Conference on Conceptual Modeling, Springer, 2024, pp. 94–111. doi:10.1007/978-3-031-75872-0_6.
- [15] I. Reinhartz-Berger, S. J. Ali, D. Bork, Leveraging LLMs for domain modeling: the impact of granularity and strategy on quality, in: International conference on advanced information systems engineering, Springer, 2025, pp. 3–19. doi:10.1007/978-3-031-94569-4_1.
- [16] J. Cámara, J. Troya, L. Burgueño, A. Vallecillo, On the assessment of generative AI in modeling tasks: an experience report with ChatGPT and UML, Software and Systems Modeling 22 (2023) 781–793. doi:10.1007/s10270-023-01105-5.
- [17] R. Chen, J. Shen, X. He, A model is not built by a single prompt: LLM-based domain modeling with question decomposition, arXiv preprint (2024). doi:10.48550/arXiv.2410.09854.
- [18] Y. Yang, B. Chen, K. Chen, G. Mussbacher, D. Varró, Multi-step iterative automated domain modeling with large language models, in: Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, 2024, pp. 587–595. doi:10.1145/3652620.3687807.
- [19] J. Silva, Q. Ma, J. Cabot, P. Kelsen, H. A. Proper, Towards human-in-the-loop LLM-enabled domain modeling, in: International Conference on Conceptual Modeling, Springer, 2025, pp. 127–145. doi:10.1007/978-3-032-08623-5_7.
- [20] Y. Hwang, Y. Kim, J. Koo, T. Kang, H. Bae, K. Jung, LLMs can be easily confused by instructional distractions, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, 2025, pp. 19483–19496. doi:10.18653/v1/2025.acl-long.957.
- [21] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. V. Le, E. H. Chi, Least-to-most prompting enables complex reasoning in large language models, in: The Eleventh International Conference on Learning Representations, 2023. URL: <https://openreview.net/forum?id=WZH7099tgfM>.
- [22] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, in: The Eleventh International Conference on Learning Representations, 2023. URL: <https://openreview.net/forum?id=1PL1NIMMrw>.
- [23] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain of thought prompting elicits reasoning in large language models, in: Advances in Neural Information Processing Systems, volume 35, 2022, pp. 24824–24837. URL: https://openreview.net/forum?id=_VjQlMeSB_J.
- [24] N. Kertkeidkachorn, R. Ichise, T2kg: An end-to-end system for creating knowledge graph from unstructured text., in: AAAI workshops, volume 17, 2017, p. 39.
- [25] I. Melnyk, P. Dognin, P. Das, Knowledge graph generation from text, in: Findings of the Association for Computational Linguistics: EMNLP 2022, 2022, pp. 1610–1622. doi:10.18653/v1/2022.findings-emnlp.116.
- [26] B. Mo, K. Yu, J. Kazdan, P. Mpala, L. Yu, C. I. Kanatsoulis, S. Koyejo, KGGen: Extracting knowledge graphs from plain text with language models, in: The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2026. URL: <https://openreview.net/forum?id=YyhRJXxbpi>.
- [27] K. Chen, Y. Yang, B. Chen, J. A. Hernández López, G. Mussbacher, D. Varró, Automated domain modeling with large language models: A comparative study, in: 2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS), 2023, pp. 162–172. doi:10.1109/MODELS58315.2023.00037.
- [28] C. Verbruggen, L. Netz, P.-L. Glaser, M. Scholz, C. Huemer, M. Calamo, B. Rumpe, M. Snoeck, D. Bork, Toward a community-curated golden dataset of uml models, in: 2025 ACM/IEEE 28th International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 2025, pp. 43–50. doi:10.1109/MODELS-C68889.2025.00012.