Leveraging LLMs in text-based ontology-driven conceptual modeling

Matheus L. Coutinho[1]

[1] *Ontology & Conceptual Modeling Research Group (NEMO), Federal University of Espírito Santo (UFES), Brazil*

Abstract

This research introduces the use of a Large Language Models (LLM) assistant into the modeling tool proposed for Tonto, a text-based language for ontologies based on the Unified Foundational Ontology (UFO). By integrating Tonto with LLMs, we aim to improve conceptual modeling by employing LLMs in assisting tasks such as summarizing models, inferring meta properties of classes and the ontological nature of their instances, and creating elements based on context. This project investigates how this integration can improve modeling efficiency, accuracy, and the overall user experience with an LLM-powered assistant.

Keywords

Textual Language, Ontology Development, UFO, OntoUML, AI, Large Language Models (LLM), Machine learning (ML)

1. Introduction

Conceptual modeling is fundamental within Computer Science, impacting areas like AI and software design. It offers a way to formally represent real-world ideas, essential for building effective information systems. Strong theoretical groundwork and tools are key to ensuring model quality and supporting analysis. Recently, foundational ontologies have become the basis for conceptual modeling languages, providing a structured framework for representing knowledge. The Unified Foundational Ontology (UFO) [1, 2] is a prime example, drawing from linguistics and philosophy to offer a robust foundation for ontology-driven conceptual modeling.

UFO was employed in the design of the OntoUML profile for UML class diagrams [1, 3] which introduces various stereotypes that correspond to the concepts defined in UFO as well as grammatical formal constraints that reflect UFO's axiomatization. A key goal for the language was to support the definition of high-quality well-founded reference ontologies. Over time, OntoUML has been accompanied by advanced tools focused on: (i) editing and syntactic verification of models to conform with UFO's axioms [2, 4]; (ii) model simulation, respecting the modal aspects of UFO [5]; (iii) automatic generation of database schemes guided by UFO meta properties [6]; (iv) detection of anti-patterns [7], among others.

Despite the various benefits of the language, it relies on a diagrammatic notation, which imposes certain burdens for managing large models and coping with version management (among other tasks). The utility of representing ontologies through a textual syntax has led to the creation of tools such as Turtle [8] and XML [9] serializations of OWL [10] ontologies. Other text-based conceptual modeling languages with similar purposes were created, e.g., OML [11], Alloy [12], among others. Textual formats enable easier manipulation in editors and tools and facilitate the integration of ontologies within software systems.

LLMs have been proving to have good results when dealing with natural language tasks, and even formal language tasks. A number of works are using Large Language Models to enhance modeling, for example GPT4 with its code generating capabilities [13], detection of mistakes in domain models [14], automating domain modeling with LLMs [15], among others [16, 17, 18, 19].

Proceedings of the Joint Ontology Workshops (JOWO) - Episode X: The Tukker Zomer of Ontology, and satellite events co-located with the 14th International Conference on Formal Ontology in Information Systems (FOIS 2024), July 15-19, 2024, Enschede, The Netherlands.

matheus.l.coutinho@edu.ufes.br (M. L. Coutinho)

2. Proposed work

This work is based on the language for well-founded ontologies named Tonto (**T**extual **Onto**logies)[1] and its set of tools in order to enable conceptual modeling based on Ontological foundations with UFO and OntoUML. The set of tools include a Command Line Interface (CLI)[2], a Visual Studio Code Extension[3], and a Package Manager to allow the modularization of models. Tonto syntax is created to be more concise and easier to read than OWL serializations, allowing fast understanding of concepts in a model and to better verify changes in version control tools. It also provides semantically-motivated syntactic verification of models to ensure ontological consistency.

Unlike OntoUML and Tonto, other existing textual modeling languages and the mentioned modeling approaches using LLMs lack the systematic grounding in a foundational ontology, limiting the benefits achievable from the rigorous distinctions and principles such a framework provides.

Objectives. Tonto, as a textual language, has the potential for integration with LLM-based tools. Recent research highlights the remarkable zero-shot and few-shot capabilities of Large Language Models (LLMs), such as GPT-4 [20], in diverse Natural Language Processing (NLP) tasks. This demonstrates their potential as versatile tools for addressing NLP challenges [21].

Three key objectives are proposed: (i) Develop an integration Framework, creating a robust interface between Tonto and advanced LLMs (e.g., GPT [20]), which will enable bidirectional communication and processing of modeling language and natural language data, making it possible to insert and retrieve information from a model specified in Tonto with natural language and vice-versa; (ii) Explore the potential of LLMs to provide assistance in the following areas: Summarizing models, inference of stereotypes (using UFO theory) [22], and the creation of new elements for the model [15] for context-aware guidance; (iii) Assess the impact of LLM-powered features on modeling efficiency, accuracy, and user experience. Iterate on the integration framework for optimization.

Overall, the combination of UFO as a foundational ontology and LLMs have the potential to make conceptual modeling more efficient, accurate and with better UX.

3. Approach

The project will adopt a mixed-methods approach, combining both qualitative and quantitative research techniques. The first step is prototyping and experimenting, where the development of LLM-enhanced Tonto prototypes would be developed.

Prompt Engineering. According to [23], different techniques could be used with LLM in the prompt design to increase the quality of the results. One basic way is to describe the task and hope the LLM already knows how to do it, named zero-shot learning, however this is not always reliable. A better method is to provide a few examples of what you're looking for, named few-shot learning, helping the LLM learn the pattern. For complex tasks, you can even show the LLM step-by-step how to reach the answer, making it easier for the model to follow your thinking. This method is called chain-of-thought.

Evaluation. We propose doing controlled experiments comparing the use of traditional Tonto with the LLM-powered version, measuring task completion time, model quality metrics, and user satisfaction. Each analysis will compare the prompt engineering techniques to discover the best performing one. The second step is executing qualitative studies with domain experts to gather feedback on the utility and usability of LLM-augmented features, and in-depth interviews for identifying potential challenges and areas for improvement.

[1] The Tonto EBNF specification can be found at https://w3id.org/tonto/ebnf.
[2] https://w3id.org/tonto/cli
[3] The extension is available in the marketplace at https://w3id.org/tonto/extension.

The goal of this research is to answer three research questions: (i) Could an LLM-based assistant facilitate the development of higher-quality models with reduced effort? (ii) To what extent can LLMs improve the efficiency and accuracy of conceptual models created in Tonto? (iii) What are the key user perceptions and challenges regarding the integration of LLMs within a conceptual modeling environment?

This research holds the potential to improve the field of conceptual modeling by introducing AI-powered capabilities. Success could lead to more accessible and intuitive modeling environments and enhanced understanding of complex systems through clearer and better-structured models.

4. Related Work

Many ideas for improving conceptual modeling using LLMs are emerging recently. For example, [17] proposes an approach to improve completion in domain modeling activities. [15] shows how given a textual description in its entirety, a complete domain model is derived without any human interaction. [16] enables domain experts to construct purposive models by operating at natural language level. [18] shows an automated approach for the extraction of UML class diagrams from natural language software specifications. Prior to LLMs, there were already some attempts for automated generation of Domain Models using NLP techniques [24, 25, 26] . None of these works take into account using a foundational ontology, like UFO, when using LLMs to build or complete ontologies, meaning that using LLMs with Tonto could make a contribution to the state of the art of the field.

References

[1] G. Guizzardi, Ontological foundations for structural conceptual models, Centre for Telematics and Information Technology, Enschede, The Netherlands, 2005.

[2] G. Guizzardi, et al., Ufo: Unified foundational ontology, Applied Ontology 17 (2022) 167–210. doi:10.3233/AO-210256.

[3] G. Guizzardi, et al., Types and taxonomic structures in conceptual modeling: a novel ontological theory and engineering support, Data & Knowledge Engineering 134 (2021) 101891.

[4] C. M. Fonseca, T. P. Sales, V. Viola, L. B. R. Fonseca, G. Guizzardi, J. P. A. Almeida, Ontology-driven conceptual modeling as a service, in: The Joint Ontology WS, CEUR-WS, 2021.

[5] A. B. Benevides, G. Guizzardi, B. F. B. Braga, J. P. A. Almeida, Validating modal aspects of ontouml conceptual models using automatically generated visual world structures, J. UCS 16 (2010) 2904–2933.

[6] G. Guidoni, J. Almeida, G. Guizzardi, Transformation of ontology-based conceptual models into relational schemas, in: Conceptual Modeling. ER 2020, Springer, 2020, pp. 315–330. doi:10.1007/978-3-030-62522-1_23.

[7] T. P. Sales, G. Guizzardi, Ontological anti-patterns: Empirically uncovered error-prone structures in ontology-driven conceptual models, Data & Knowledge Engineering 99 (2015). doi:10.1016/j.datak.2015.06.004.

[8] W3C, Owl turtle, 2014. URL: https://www.w3.org/TR/turtle. Accessed: March 4, 2024.

[9] W3C, Owl xml, 2009. URL: https://www.w3.org/2007/OWL/wiki/XML_Serialization.html. Accessed: March 4, 2024.

[10] W3C, Owl, 2012. URL: https://www.w3.org/OWL/, accessed: March 4, 2024.

[11] D. A. Wagner, M. Chodas, M. Elaasar, J. S. Jenkins, N. Rouquette, Ontological Metamodeling and Analysis Using openCAESAR, 2023, pp. 925–954. doi:10.1007/978-3-030-93582-5_78.

[12] D. Jackson, Software abstractions: Logic, Language, and Analysis, MIT Press, 2016.

[13] D. Bork, S. J. Ali, B. Roelens, Conceptual modeling and artificial intelligence: A systematic mapping study, 2023. arXiv:2303.06758.

[14] P. Singh, Y. Boubekeur, G. Mussbacher, Detecting mistakes in a domain model, in: Proceedings of

the 25th Intl Conf on Model Driven Engineering Languages and Systems: Companion Proceedings, 2022, pp. 257–266.

[15] 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 Intl Conf on Model Driven Engineering Languages and Systems (MODELS), IEEE, Västerås, Sweden, 2023, pp. 162–172. doi:10.1109/MODELS58315.2023.00037.

[16] V. Kulkarni, S. Reddy, S. Barat, J. Dutta, Toward a Symbiotic Approach Leveraging Generative AI for Model Driven Engineering, in: 2023 ACM/IEEE 26th Intl Conf on Model Driven Engineering Languages and Systems (MODELS), IEEE, Västerås, Sweden, 2023, pp. 184–193. URL: https://ieeexplore.ieee.org/document/10343767/. doi:10.1109/MODELS58315.2023.00039.

[17] M. B. Chaaben, L. Burgueño, H. Sahraoui, Towards using Few-Shot Prompt Learning for Automating Model Completion, in: 2023 IEEE/ACM 45th Intl Conf on Software Engineering: New Ideas and Emerging Results (ICSE-NIER), IEEE, Melbourne, Australia, 2023, pp. 7–12. URL: https://ieeexplore.ieee.org/document/10173909/. doi:10.1109/ICSE-NIER58687.2023.00008.

[18] S. Yang, H. Sahraoui, Towards automatically extracting UML class diagrams from natural language specifications, in: Proceedings of the 25th Intl Conf on Model Driven Engineering Languages and Systems: Companion Proceedings, MODELS '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 396–403. URL: https://doi.org/10.1145/3550356.3561592. doi:10.1145/3550356.3561592.

[19] R. Saini, G. Mussbacher, J. L. C. Guo, J. Kienzle, Machine learning-based incremental learning in interactive domain modelling, in: Proceedings of the 25th Intl Conf on Model Driven Engineering Languages and Systems, MODELS '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 176–186. URL: https://doi.org/10.1145/3550355.3552421. doi:10.1145/3550355.3552421.

[20] OpenAI, GPT-4 Technical Report, 2024. URL: http://arxiv.org/abs/2303.08774.

[21] C. Qin, A. Zhang, Z. Zhang, J. Chen, M. Yasunaga, D. Yang, Is ChatGPT a General-Purpose Natural Language Processing Task Solver?, 2023. URL: http://arxiv.org/abs/2302.06476.

[22] P. P. F. Barcelos, T. P. Sales, E. Romanenko, J. P. A. Almeida, G. Engelberg, D. Klein, G. Guizzardi, Inferring Ontological Categories of OWL Classes Using Foundational Rules, in: Frontiers in Artificial Intelligence and Applications, IOS Press, 2023. URL: https://ebooks.iospress.nl/doi/10.3233/FAIA231122. doi:10.3233/FAIA231122.

[23] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, 2021. URL: http://arxiv.org/abs/2107.13586.

[24] M. Ibrahim, R. Ahmad, Class Diagram Extraction from Textual Requirements Using Natural Language Processing (NLP) Techniques, in: 2010 Second Intl Conf on Computer Research and Development, IEEE, Kuala Lumpur, Malaysia, 2010, pp. 200–204. URL: http://ieeexplore.ieee.org/document/5489508/. doi:10.1109/ICCRD.2010.71.

[25] M. S. Jyothilakshmi, P. Samuel, Domain ontology based class diagram generation from functional requirements, in: 2012 12th Intl Conf on Intelligent Systems Design and Applications (ISDA), IEEE, Kochi, India, 2012, pp. 380–385. URL: http://ieeexplore.ieee.org/document/6416568/. doi:10.1109/ISDA.2012.6416568.

[26] 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 Intl Conf on Model Driven Engineering Languages and Systems, MODELS '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 250–260. URL: https://doi.org/10.1145/2976767.2976769. doi:10.1145/2976767.2976769.