AI Assisted Domain Modeling Explainability and Traceability

Jonathan Silva
jonathan.silva@uni.lu
University of Luxembourg
Belval, Esch-sur-Alzette, Luxembourg

Abstract

Domain Models are abstract representations of selected elements in a domain that is created in a collaborative process between domain and modeler experts. The participants share domain knowledge to conceptualize and reason about the elements that will create the domain models. Through this exchange, a comprehensive and accurate representation of the domain is achieved, ensuring that the model captures the relevant aspects and relationships in the domain. Research in Artificial Intelligence (AI) has explored various methods to assist in the creation of domain models from text using Natural Language Processing (NLP) and Machine Learning (ML). Recent advancements with Large Language Models (LLMs) have shown that it is possible to create domain models using prompting techniques; however, the generated domain models contain errors and remain constrained by the performance of the LLM used.

Despite the impressive capabilities of LLMs to create domain models, it is evident that it does not address the needs of domain and modelers experts that participate in the creation of domain models. Every AI technique has its advantages and limitations that must be integrated with human feedback in a collaboration process. Therefore, we propose an approach that incorporates human-AI collaboration supported by AI assistants that follows a dialogue approach to understand the users needs and purpose to suggest relevant models. Our proposal combines symbolic and subsymbolic AI techniques with explainability and traceability of the decisions that assist to create domain models that are relevant for the users.

CCS Concepts

• **Computing methodologies** → **Natural language processing**; **Artificial intelligence**; • **Software and its engineering** → **Model-driven software engineering**; **Traceability**.

Keywords

Domain Modeling, Large Language Models, Uncertainty, Explainability, Traceability.

ACM Reference Format:
Jonathan Silva. 2024. AI Assisted Domain Modeling Explainability and Traceability. In *ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion '24), September 22–27, 2024, Linz, Austria.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3652620.3688197

MODELS Companion '24, September 22–27, 2024, Linz, Austria

ACM ISBN 979-8-4007-0622-6/24/09
https://doi.org/10.1145/3652620.3688197

1 PROBLEM STATEMENT

Domain modeling is a complex task that requires collaborative effort [17] between domain experts and modeling experts. The outcome is to create a domain model that represents the selected details of the domain [15]. To create an abstract representation of these details, four conceptions are needed: a full understanding of the domain, the purpose to create the model, the focus on the domain details based on the purpose, and the artifacts that will represent the domain [15]. Furthermore, relational reasoning is necessary in domain modeling to form inferences from facts or premises to propose the relations that will be part of the model. [30]

The advancements of AI allow us to integrate virtual assistants in the process of creating domain models. It is important that these assistants are created with solid frameworks that address the limitations identified in previous research and focus on addressing user needs [20]. Furthermore, to implement a useful AI assistant, it is important that it explains to the user why the recommendations are relevant [21]. Thus, the relevance of the recommendations should be based on user needs and the purpose of the modeling process.

The use of symbolic AI, which represents knowledge using symbolic representations [16], and subsymbolic AI, where statistical learning methods are used to find correlations based on training data [18], has been explored to assist in the creation of domain models. An approach of symbolic AI to assist in the creation of domain models is to use rule-based Natural Language Processing (NLP) [15]. Arora et al. implemented rule-based NLP to create domain models and identified that some model elements are not relevant to the level of abstraction and scope of the system [3]. Another approach by Saini et al. added subsymbolic AI to implement an interactive tool that allows the user to update the proposed domain models, this tool was implemented combining rule-based NLP and Machine Learning (ML) to create domain models [24].

With recent advances in Large Language Models (LLMs), there has been an increasing interest in researching the capabilities of these subsymbolic AI models for the creation of domain models. LLMs have demonstrated capabilities to create domain models from domain descriptions; however, the outputs have errors in representing the abstraction of some model elements, which limits the quality of the models. Furthermore, LLMs create different domain models for the same input, making the reproducibility of experiments difficult [11]. In the literature, there are proposals to use prompting techniques to address complex tasks that include reasoning [29], planning, and self-assessment [33]. The use of prompting techniques with examples of reference solutions helps increase the number of correct model elements; however, it still has some limitations to recommend correct relationships, and LLMs are not ready to automate domain model creation due to this low performance [9].

Recently, new approaches have been proposed to solve complex problems with autonomous agents based on LLM. These agents have profiles that specialize them to complete certain tasks, using the information in memory to create plans and actions to solve specific problems [28]. By implementing virtual AI assistants with the agents framework, we can enhance the alignment of the purpose between human users and virtual assistants in the modeling process.

The integration of an AI assistant into a modeling process requires that AI address human expectations, and trust of users in human-AI interaction is crucial [26]. To allow users to understand the recommendations, the use of explainability and traceability is important because it allows transparency by allowing the user to know how an answer was retrieved [5] and thus generates trust in the recommendations. During this process, the actions performed by the assistant require explainability and traceability to enhance human-AI collaboration, and create relevant models for the user.

During the doctoral research, our aim is to answer which methods and tools should be developed to enable an effective collaboration between human user and AI assistant, specifically, we have the following research questions:

- **RQ1** Which AI assistants are needed to effectively collaborate with users in a domain modeling process?
- **RQ2** What is the design of AI assistants that create domain models based on user needs and modeling purpose?
- **RQ3** What methods are needed to enhance the explainability and traceability of AI assistants recommendations to ensure transparency in the modeling process?

The rest of the paper is organized as follows. In Section 2, we outline the expected contributions. Then, in Section 3, we review related work. In Section 4, we present our proposal. Next, in Section 5, we describe the evaluation plan. Finally, Section 6 informs about the current status and the plan to complete the research.

2 EXPECTED CONTRIBUTIONS

In this section, we detail the contributions of our approach:

C1: A metamodel to represent explicitly the AI assistants participating in the modeling process and capture the interaction between the AI assistants and human user for explainability and traceability of modeling decisions. This contribution address **RQ1** to formalize the AI assistants required in the domain modeling process.

C2: Propose the symbolic and subsymbolic AI techniques that are required by the AI assistants to suggest model elements that are relevant in the domain and for the user purpose. This contribution address **RQ2** by selecting the AI techniques required to create the domain models according to the user needs and modeling purpose.

C3: Identify the sources of uncertainty associated with the AI techniques and implement strategies to reduce it with human-AI interaction. This contribution collaborates with the explainability of modeling elements in **C4**.

C4: Tracking the information exchanged during the Human-AI dialogue that enhance the explainability and traceability of the created domain models. This contribution address **RQ3** to enable transparency in the creation of the domain models.

C5: Implement an open-source tool with the AI assistants that interacts in a human-AI collaborative dialogue following a domain modeling process. The tool will be evaluated in **C6**.

C6: Evaluate our proposal quantitatively with reference solutions and qualitatively with users who have different levels of modeling experience and domain knowledge. With this evaluation, we will validate the proposal in **C1**, **C2** and **C4**.

3 RELATED WORK

In this section, we review the related work about AI assistants, explainability, and traceability in domain modeling.

3.1 Domain Modeling AI Assistants

Arora et al. propose using NLP and extraction rules to generate domain models from requirements documents. They observed that while model elements can be identified, suggesting correct relationships is a challenge because many were not relevant in the domain model [3]. The relevance of these recommendations depends on factors such as context, abstraction level, assumptions, and the scope of the domain model that may not be included in the requirements text [3]. Saini et al. identified the lack of interactive interfaces for AI assistants in domain modeling, for that reason they proposed a bot that combines NLP and ML neural networks to create domain models from university exercises problems [24]. This proposal creates domain models for the domains that exist in its training data; however, it is not validated with domain descriptions from real systems [24]. Further research is required to evaluate the accuracy of the domain model recommendations with more problem descriptions and industrial cases [24].

More recent studies have explored using LLMs to create domain models with Input-Output (IO) and Chain-of-thoughts (CoT) prompting techniques. In IO prompting, the Zero-shot technique relies on the LLM capability gained in the training process to perform the task [7]. Another IO prompt is the Few-shot technique that adds examples of problems and solutions to improve the LLM output on the requested task [7]. Cámara et al. qualitatively evaluated the potential to create domain models from textual descriptions using IO prompting [11]. It was possible to create UML class diagrams; however, the output model contains semantic errors and some abstractions required in the domain model were not used, such as using inheritance instead of attributes or creating association classes [11]. Another approach is CoT prompting that uses examples with intermediate reasoning steps that improve LLM results in complex arithmetic problems [29]. Chen et al. compared various prompting techniques and concluded that the few-shot prompt technique performs better than CoT and improves the performance of LLMs to create the reference solution [9]. However, LLMs have problems in identifying the correct relationships compared to other model elements [9].

3.2 Domain Model Explainability

The use of explainable AI in decision-making processes is important to understand the final outcomes of these decisions [19]. AI that implements rule-based learning is identified as transparent and can be explained easily; on the contrary, models based on neural networks are opaque models that require post hoc explainability [2]. Batot et al. state that it is essential to clarify the reasons behind identified links among artifacts to support explainability, which is also a requirement for traceability [5]. Their proposal uses trace links, which can be collected with textual explanations in natural language or automatically identified using information retrieval and rule-based techniques. It is important to record the source and details of the identification process to ensure explainability [5]. Aslam et al. propose that the output of AI techniques can be augmented by using explanations that clarify how the result was obtained. They propose using an ontology-driven conceptual model to complete the explanations with characteristics, properties, and qualities [4]. Zhao et al. mention that prompting techniques is an emerging paradigm for explaining LLM recommendations, for example, using CoT prompting to explain the behavior of the LLM [34].

Another approach that is increasing in explainability research is to quantify the uncertainty of the predictions to understand the reliability and limitations of LLM models [34]. Uncertainty can be categorized as aleatoric and epistemic [14]. Aleatoric refers to the inherent variability of the phenomenon and it is not possible to reduce it. Epistemic uncertainty occurs due to the lack of knowledge that exists about a system or its elements, and can be reduced with more information [27]. The research of Burgueño et al. addressed belief uncertainty, a specific type of epistemic uncertainty that occurs when the user has doubts about the validity of a statement. Their proposal includes a UML Profile that includes the degree of belief to express the uncertainty that agents, e.g. domain experts or modelers, have about the model elements [8]. Xiong et al. review some approaches to express the uncertainty of the LLM model by communicating the confidence level of the prediction, this approach mimics how human experts express their confidence [31]. They found that LLMs exhibit significant degrees of overconfidence that can be mitigated to some extent by combining Verbalized confidence to ask LLM to measure confidence, and consistency-based confidence by measuring the consistency of multiple answers [31].

3.3 Domain Model Traceability

Traceability is defined in the IEEE Standard Glossary as "the degree to which a relationship can be established between two or more products of the development process" [10]. The traceability of the decisions for domain modeling is needed to increase the trust in the recommendations provided by the AI assistant. Batot et al. stated that traceability is the ability to trace different artifacts of a system and proposed a metamodel to capture evidence of links between related artifacts for model-driven tools with trace elements [5]. In the case where an AI technique is used, it is necessary to capture the rules applied in the case of rule-based techniques or the algorithm and the training set in the case of machine learning [5]. Canovas et al. identified the problem that design decisions were not traced when designing a modeling language. To address it, they proposed a Domain-Specific Modeling Language (DSML) that represents change proposals during a collaboration process that can be used to automate a decision process [12].

Other approaches for traceability collect the information used to design the models. For example, De Kinderen et al. stated that metamodel provenance is needed to record the intentions to include elements in the abstract syntax for the evolution of languages. Their proposal combines concepts from Goal-Oriented Requirement Engineering and DSML design to track the provenance of metamodel concepts, their attributes, relations, and constraints [13]. Another approach from Saini et al. used graphs for traceability of domain model elements, by extracting trace information from a given problem description and recording it in knowledge graphs. Their proposal assist modellers to find words or sentences used to propose a model element [23].

4 PROPOSED APPROACH

Our proposal uses the LLM agent framework proposed in [28] to create a conversational bot to assist in the creation of domain models. The design of the AI assistant supports a collaborative interaction to align AI suggestions with user needs and will be developed using a model-driven approach using the metamodel shown in Figure 2. This approach records the interaction between the assistant and the user using the metamodel, which enables the explainability and traceability of the selected model elements for the domain model. The metamodel is separated into three packages: Modeling Assistant contains the concepts associated with the AI assistants, the Explainability package includes the concepts to explain the model elements proposed, and the Traceability package includes the concepts for the decisions to select the proposed domain model.

4.1 Human-AI collaboration

First, we propose to define a process that includes AI assistants with domain experts and modelers to collaborate on the decision of the model elements that will be included in the domain model. In Figure 1, we specify the four activities required in the process:

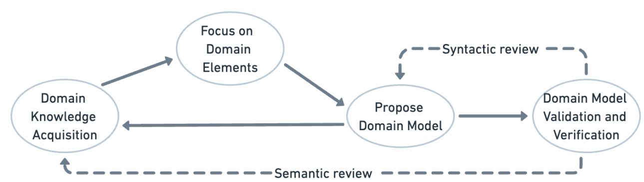

Figure 1: Our proposed process for Human-AI collaboration.

(1) Domain Knowledge Acquisition: The AI assistant interacts with the user to understand the domain, purpose, and user needs. It requests more details in case there are doubts to propose the model elements.
(2) Focus on Domain Elements: To identify the model elements, the AI assistant will use symbolic and subsymbolic AI techniques, and quantify the uncertainty of the recommendations. Then, the interaction with the user executes strategies to decrease the uncertainty and improve suggestions with more domain knowledge.
(3) Propose Domain Model: The AI assistant proposes partial domain models to validate that the proposal is aligned with user needs and purpose. The proposals generated by the assistant improve in an iterative way when new or modified model elements are included.
(4) Domain Model Validation and Verification: This activity aims to increase the quality of the model by implementing syntactic and semantic reviews with the user. The lack of quality of the model will trigger an iterative improvement of the domain model again.

This process integrates human intelligence with AI to create a relevant domain model. The iterative approach of the process ensures that the model is adjusted to the user needs and purpose.

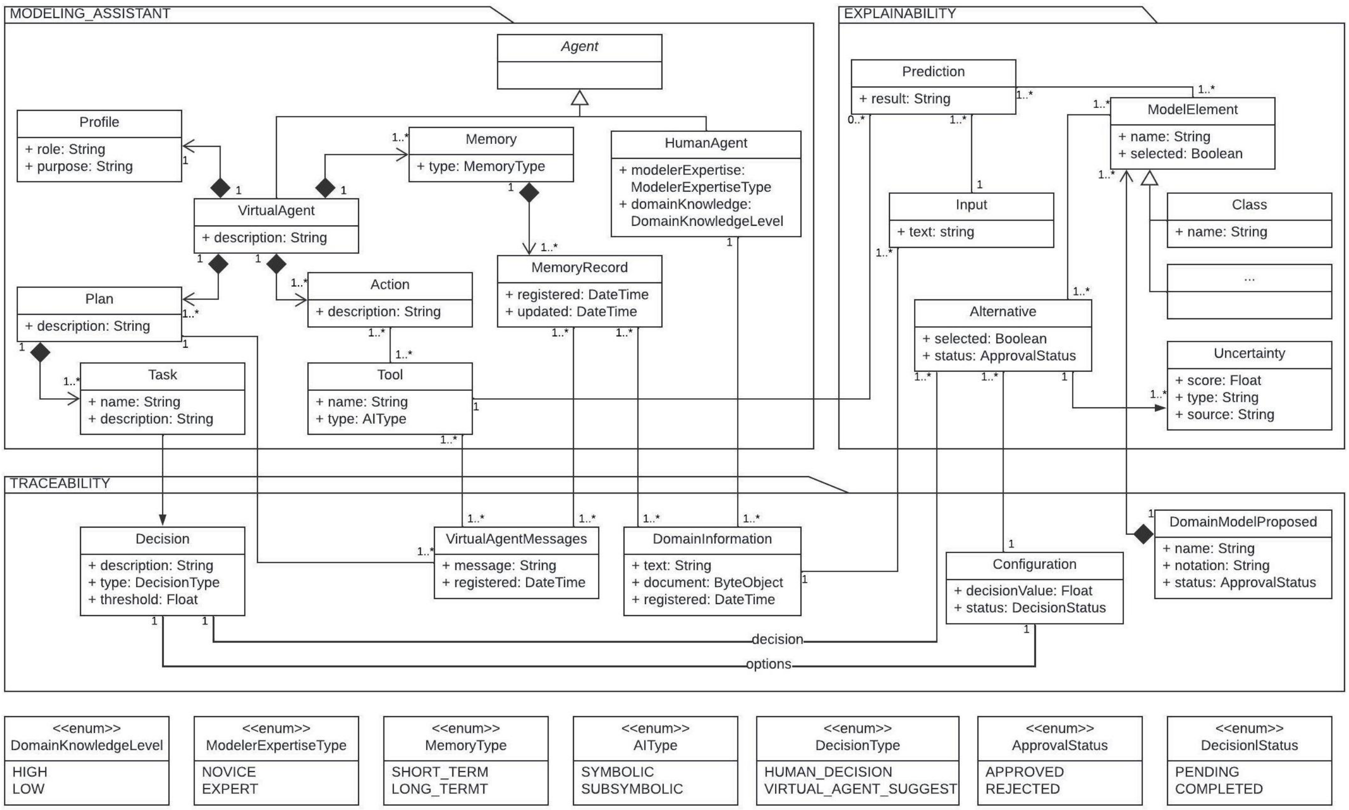

Figure 2: metamodel for AI Assisted Domain Modeling Explainability and Traceability in a domain modeling process.

4.2 AI assistants

Approaches to resolve complex problems with LLMs aim to use autonomous agents. These agents have profiles that specialize them to memorize past events to plan future tasks and execute actions to achieve the desired outcome [28]. In our approach, we envision the AI assistant as a group of autonomous agents that are created with four modules: profile, memory, plan, and action. The agent has a Profile module that specializes in performing specific domain modeling tasks. The Memory module serves to retrieve past interactions with the user and store the model elements proposed. The Plan module is used to execute tasks that aim to reduce the uncertainty of alternative model elements. Finally, the action module uses the tools required for the identification and classification of model elements. These tools are enabled by symbolic AI, subsymbolic AI, or the combination of both techniques to improve the suggestions of the proposed domain model. The use of this agent framework enables our proposal to use various agents with specific purposes such as: identifying and classifying model elements, evaluating model alternatives, and repairing domain models.

In the metamodel shown in Figure 2, the human user is represented as *HumanAgent* and the AI assistants are represented as *VirtualAgent*. The virtual agent is composed of the *Profile*, *Memory*, *Plan*, and *Action*. There are two types of *Memory*: short-term memory for instant interactions and long-term memory to recall past information. The *Plan* is composed of several *Tasks* that the assistant must fulfill. The *Action* is executed by the assistants using the *Tools* that generate predictions that are used to suggest the elements of the model. The results of the tools are associated with the messages to communicate the results to the human user.

4.3 Explainability

In order to enable explainability in the domain modeling process, we propose recording the explanations of the predictions made by the AI assistants. The assistant will generate predictions based on the information provided by the user in the form of chat messages and/or documents; this information is processed to create an input adjusted to the specific AI technique (symbolic or subsymbolic). This information provided by the user is recorded in memory and is used by the assistants as input to identify the model elements. These predictions contain model elements that are grouped into alternatives, which are partial domain models, and each alternative has an uncertainty score. When the uncertainty is above a certain threshold, the agent initiates a conversation with the user to reduce the uncertainty of the alternatives by acquiring more information about the domain.

In Figure 2, we include in the *Explainability* package the *Input* used to generate the *Prediction*, this prediction is associated with the tool used by the assistant and the domain information provided by the user. The result of the predictions contains *ModelElements* that are part of *Alternatives* with their corresponding *Uncertainty*. The uncertainty is reduced when the virtual agent interacts with the user to acquire more domain information.

4.4 Traceability

In the traceability part, we propose to record the decisions taken to select the domain model alternative that is more relevant to the user. During the interaction between the agent and the user, there will be decisions to select alternatives that achieve an uncertainty threshold, these alternatives can be included or excluded based on the modeling expertise of the user, the suggestions of the AI assistant, or a mixed approach using the suggestion of AI and the human expertise. Finally, a domain model is proposed and the interaction recorded in memory, the plan followed, and the actions performed by the assistants can be used to explain the domain modeling decisions.

The *Traceability* package in the metamodel is shown in Figure 2 and includes the *DomainInformation* provided by the user, the *VirtualAgentMessages* generated by the AI assistants when the model elements are proposed, and more domain information is requested during the human-AI interaction. Furthermore, some tasks require making a *Decision* to include or exclude model elements, this decision could be achieved by AI suggestion, human decision, or a hybrid approach with human-AI collaboration. The metamodel considers to record *Configurations*, which are a group of alternatives proposed by the AI assistant. The decisions taken during the human-AI interaction aim to create the *DomainModelProposed* that contains the model elements with low uncertainty and that are more relevant to the domain.

5 PLAN FOR EVALUATION

To evaluate our approach, we plan to experiment with reference solutions to evaluate the quantitative performance and perform an exploratory study with users to evaluate the tool qualitatively. This proposal does not involve associated ethical risks.

Quantitative evaluation: To evaluate whether the AI assistants are proposing relevant model elements, we will assess the tool quantitatively by comparing the predictions with reference solutions. This approach requires collecting domain modeling problems in textual descriptions that includes reference solutions created by experts and compare the output with performance metrics such as precision, recall, and F1-score. There are some data sets in the literature that compare the outputs of AI techniques with reference solutions. In [22] the tool was evaluated with problem descriptions divided into ten problems for training and eight problems for testing. Similarly, in [9] the experiments compared different prompting techniques using two problem descriptions for few-shot examples and eight problems for comparing the output performance. In [6], the authors collected a dataset of 120 software requirements, and used 30 of them to evaluate the performance of ChatGPT with a tool that implements NLP techniques.

Qualitative evaluation: The qualitative evaluation of the tool will be performed with an empirical evaluation of users with different modeling experience or domain knowledge to evaluate whether explainability and traceability improve the creation of relevant domain models. For a user with low modeling experience, we expect it to require an assistant that explains the modeling elements to improve their understanding of the domain model. In contrast, the experienced modeler requires a streamlined experience with the tool that assists to model complex scenarios. Regarding users with high domain knowledge, the AI assistant will use their knowledge as the source of truth to generate the relevant model. Instead, if the user has little knowledge about the domain, some assumptions will be required and it needs to be explicitly specified in the output model and traceability of the domain model. Our evaluation approach will use as reference the research in [3] where the authors conducted surveys with users to evaluate the correctness and relevance of the recommendations.

Table 1: Next activities and proposed timeline

Task	Timeline	Progress
Research Topic Definition	Sep '23 - Feb '24	100%
Literature Review	Sep '23 - Sep '24	70%
AI assistants	Mar '24 - Oct '25	50%
Explainability	Aug '24 - Apr '25	0%
Traceability	Feb '25 - Oct '25	0%
Testing and Validation	Jun '25 - Apr '26	0%
Thesis writing	Feb '26 - Sep '26	0%

6 CURRENT STATUS

This section presents the work that has been achieved and the next activities to complete the research.

Open-source tool: We are using the BESSER open source platform [1] for the implementation of our proposal, specifically we have developed a prototype that uses the BESSER-bot framework [1] to create a conversational bot that follows states in a dialogue conversation to assist in discovering model elements with LLM suggestions. The final step in the conversation creates a domain model in PlantUML diagram.

Application of Tree of Thoughts (ToT) in Domain Modeling: The ToT framework allows LLMs to explore different solutions for a problem by incorporating self-assessment and a search algorithm [32]. We submitted a paper [25] with our proposal to decompose the modeling problem into tasks to create domain models. With ToT different proposals of model elements are self-evaluated by the LLM and the best proposal is selected to continue with the discovery of new elements and propose a domain model. The experiments demonstrate that by using this framework, we can enhance the LLM recommendation to incorporate association classes. However, it shows limitations in identifying the correct relationship type. Furthermore, we created an open-source DSL to customize the task decomposition for structural or behavioral diagrams, and to include other model notations.

Next steps: In Table 1, we summarize the tasks performed, the progress for each task and next steps for the research proposal. We started to define the research topic and review the state of the art in AI assistants, Explainability and Traceability for domain modeling. Then, we have a prototype for an AI assistant that suggest model elements, and we will start with the development of the explainability and traceability. After incorporating the changes to enhance the AI assistant, we will proceed to test and validate our open-source tool. Finally, we are considering the thesis writing activity to submit it on September 2026.

[1]https://github.com/BESSER-PEARL/BESSER-Bot-Framework

References

[1] Iván Alfonso, Aaron Conrardy, Armen Sulejmani, Atefeh Nirumand, Fitash Ul Haq, Marcos Gomez-Vazquez, Jean-Sébastien Sottet, and Jordi Cabot. 2024. Building BESSER: an open-source low-code platform. In *International Conference on Business Process Modeling, Development and Support*. Springer, 203–212.

[2] Plamen P Angelov, Eduardo A Soares, Richard Jiang, Nicholas I Arnold, and Peter M Atkinson. 2021. Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 11, 5 (2021), e1424.

[3] Chetan Arora, Mehrdad Sabetzadeh, Lionel Briand, and Frank Zimmer. 2016. 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*. ACM, Saint-malo France, 250–260. https://doi.org/10.1145/2976767.2976769

[4] Maria Aslam, Diana Segura-Velandia, and Yee Mey Goh. 2023. A Conceptual Model Framework for XAI Requirement Elicitation of Application Domain System. *IEEE Access* (2023).

[5] Edouard R. Batot, Jordi Cabot, and Sebastien Gerard. 2021. (Not) Yet Another Metamodel For Traceability. In *2021 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE, Fukuoka, Japan, 787–796. https://doi.org/10.1109/MODELS-C53483.2021.00125

[6] Fatma Bozyigit, Tolgahan Bardakci, Alireza Khalilipour, Moharram Challenger, Guus Ramackers, Önder Babur, and Michel RV Chaudron. 2024. Generating domain models from natural language text using NLP: a benchmark dataset and experimental comparison of tools. *Software and Systems Modeling* (2024), 1–19.

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[8] Lola Burgueño, Paula Muñoz, Robert Clarisó, Jordi Cabot, Sébastien Gérard, and Antonio Vallecillo. 2023. Dealing with Belief Uncertainty in Domain Models. *ACM Transactions on Software Engineering and Methodology* 32, 2 (April 2023), 1–34. https://doi.org/10.1145/3542947

[9] Kua Chen, Yujing Yang, Boqi Chen, José Antonio Hernández López, Gunter Mussbacher, and Dániel Varró. 2023. 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)*. IEEE, Västerås, Sweden, 162–172. https://doi.org/10.1109/MODELS58315.2023.00037

[10] IEEE Standards Committee et al. 1990. IEEE standard glossary of software engineering terminology. *IEEE Std* 610 (1990), 12.

[11] Javier Cámara, Javier Troya, Lola Burgueño, and Antonio Vallecillo. 2023. On the assessment of generative AI in modeling tasks: an experience report with ChatGPT and UML. *Software and Systems Modeling* 22, 3 (June 2023), 781–793. https://doi.org/10.1007/s10270-023-01105-5

[12] Javier Luis Cánovas Izquierdo and Jordi Cabot. 2016. Collaboro: a collaborative (meta) modeling tool. *PeerJ Computer Science* 2 (Oct. 2016), e84. https://doi.org/10.7717/peerj-cs.84

[13] Sybren De Kinderen, Monika Kaczmarek-Heß, Qin Ma, and Iván S. Razo-Zapata. 2017. Towards Meta Model Provenance: A Goal-Driven Approach to Document the Provenance of Meta Models. In *The Practice of Enterprise Modeling*, Geert Poels, Frederik Gailly, Estefania Serral Asensio, and Monique Snoeck (Eds.). Vol. 305. Springer International Publishing, Cham, 49–64. https://doi.org/10.1007/978-3-319-70241-4_4 Series Title: Lecture Notes in Business Information Processing.

[14] Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? Does it matter? *Structural safety* 31, 2 (2009), 105–112.

[15] Christophe Feltus, Qin Ma, Henderik A. Proper, and Pierre Kelsen. 2021. Towards AI Assisted Domain Modeling. In *Advances in Conceptual Modeling*, Iris Reinhartz-Berger and Shazia Sadiq (Eds.). Vol. 13012. Springer International Publishing, Cham, 75–89. https://doi.org/10.1007/978-3-030-88358-4_7 Series Title: Lecture Notes in Computer Science.

[16] Mariusz Flasiński. 2016. *Introduction to artificial intelligence*. Springer.

[17] Paul J. M. Frederiks and Theo P. van der Weide. 2006. Information Modeling: The Process and the Required Competencies of Its Participants. *Data & Knowledge Engineering* 58, 1 (July 2006), 4–20. https://doi.org/10.1016/j.datak.2005.05.007

[18] Eleni Ilkou and Maria Koutraki. 2020. Symbolic vs sub-symbolic ai methods: Friends or enemies?. In *CIKM (Workshops)*, Vol. 2699.

[19] Marçal Mora-Cantallops, Salvador Sánchez-Alonso, Elena García-Barriocanal, and Miguel-Angel Sicilia. 2021. Traceability for trustworthy ai: A review of models and tools. *Big Data and Cognitive Computing* 5, 2 (2021), 20.

[20] David Mosquera, Marcela Ruiz, Oscar Pastor, and Jürgen Spielberger. 2024. Understanding the Landscape of Software Modelling Assistants for MDSE tools: A Systematic Mapping. *Information and Software Technology* (2024), 107492.

[21] Gunter Mussbacher, Benoit Combemale, Jörg Kienzle, Silvia Abrahão, Hyacinth Ali, Nelly Bencomo, Márton Búr, Loli Burgueño, Gregor Engels, Pierre Jeanjean, Jean-Marc Jézéquel, Thomas Kühn, Sébastien Mosser, Houari Sahraoui, Eugene Syriani, Dániel Varró, and Martin Weyssow. 2020. Opportunities in intelligent modeling assistance. *Software and Systems Modeling* 19, 5 (Sept. 2020), 1045–1053. https://doi.org/10.1007/s10270-020-00814-5

[22] Rijul Saini. 2020. Artificial intelligence empowered domain modelling bot. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. ACM, Virtual Event Canada, 1–6. https://doi.org/10.1145/3417990.3419486

[23] Rijul Saini, Gunter Mussbacher, Jin LC Guo, and Jörg Kienzle. 2021. Automated traceability for domain modelling decisions empowered by artificial intelligence. In *2021 IEEE 29th International Requirements Engineering Conference (RE)*. IEEE, 173–184.

[24] Rijul Saini, Gunter Mussbacher, Jin L. C. Guo, and Jörg Kienzle. 2020. DoMoBOT: a bot for automated and interactive domain modelling. In *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*. ACM, Virtual Event Canada, 1–10. https://doi.org/10.1145/3417990.3421385

[25] Jonathan Silva, MA Qin, Jordi Cabot, Henderik Proper, and Pierre Kelsen. 2024. Application of the Tree-of-Thoughts Framework to LLM-Enabled Domain Modeling. (2024).

[26] Sabine Theis, Sophie Jentzsch, Fotini Deligiannaki, Charles Berro, Arne Peter Raulf, and Carmen Bruder. 2023. Requirements for Explainability and Acceptance of Artificial Intelligence in Collaborative Work. In *International Conference on Human-Computer Interaction*. Springer, 355–380.

[27] Javier Troya, Nathalie Moreno, Manuel F Bertoa, and Antonio Vallecillo. 2021. Uncertainty representation in software models: a survey. *Software and Systems Modeling* 20, 4 (2021), 1183–1213.

[28] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2023. A Survey on Large Language Model based Autonomous Agents. http://arxiv.org/abs/2308.11432 arXiv:2308.11432 [cs].

[29] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[30] Ilona Wilmont, Sytse Hengeveld, Erik Barendsen, and Stijn Hoppenbrouwers. 2013. Cognitive mechanisms of conceptual modelling: How do people do it?. In *Conceptual Modeling: 32th International Conference, ER 2013, Hong-Kong, China, November 11-13, 2013. Proceedings 32*. Springer, 74–87.

[31] Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs Express Their Uncertainty? An Empirical Evaluation of Confidence Elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=gjeQKFxFpZ

[32] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[33] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. http://arxiv.org/abs/2305.10601 arXiv:2305.10601 [cs].

[34] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology* 15, 2 (2024), 1–38.