

LLM-Assisted Semantic Assessment of Entity–Relationship Diagrams in Conceptual Modeling Education

Meike Ullrich^{1,*}, Fabian Aisenbrey¹ and Andreas Fritsch¹

¹Karlsruhe Institute of Technology (KIT), Institute AIFB, Kaiserstr. 89, 76133 Karlsruhe, Germany

Abstract

Providing timely and meaningful feedback on student-created conceptual models requires assessing syntactic and semantic correctness, the latter indicating whether a model adequately represents the intended domain. Existing rule- and label-based approaches provide limited support for alternative valid terminology and modeling decisions, while direct large language model (LLM) assessments may produce incomplete or inconsistent feedback. This paper presents an LLM-assisted semantic assessment approach for Entity–Relationship diagrams that combines explicit diagram mapping and deterministic difference detection with LLM-based semantic interpretation. The approach generates element-specific feedback and makes identified differences between the task description, the sample solution, and the submitted model transparent. It supports configurable assessment instructions and generates detailed formative feedback. Evaluation on student submissions shows that the assessment approach with explicit diagram mapping and deterministic difference detection produces substantially more complete and correct feedback than direct LLM assessment.

Keywords

conceptual modeling, modeling education, Entity–Relationship diagrams, semantic assessment, automated assessment, formative feedback, large language models

1. Introduction

Conceptual modeling is a core competence in Information Systems and Computer Science education. Learning to create conceptual models requires feedback on syntactic, semantic, and pragmatic model quality [1]. While syntactic properties can often be checked deterministically, semantic assessment must determine whether a model adequately represents the intended domain. This is difficult because valid models may use alternative terminology, represent concepts at different levels of granularity, or contain different numbers of model elements.

Early systems for the automated assessment of conceptual models required student submissions to closely match a predefined solution or compared model elements using fixed attributes and labels [2, 3, 4, 5, 6]. Such approaches can identify structural deviations but provide only limited support for alternative terminology, different modeling granularities, and semantically equivalent solutions.

More recent work applies LLMs to textual or visual representations of conceptual models [7, 8, 9, 10]. Most approaches use a single-turn interaction in which an LLM receives the complete submission and grading criteria and directly generates a score or feedback. Other work augments the LLM with retrieval-augmented generation [11] or additional prompt-processing pipelines [12]. Although these approaches demonstrate the feasibility of LLM-based assessment of conceptual models, they may still produce misleading, incomplete, or inconsistent feedback due to insufficient diagram understanding, incorrect element correspondences, and misinterpreted grading criteria [7, 8, 9, 10, 13].

The presented assessment approach follows a different direction. Rather than relying on a single end-to-end LLM assessment, it combines explicit diagram mapping and deterministic difference detection with LLM-based semantic interpretation supported by targeted diagram views. By establishing explicit correspondences between diagram elements and systematically evaluating identified differences, the structured interaction is intended to produce more correct, complete, and consistent feedback.

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.

✉ meike.ullrich@kit.edu (M. Ullrich); fabian.aisenbrey@student.kit.edu (F. Aisenbrey); andreas.fritsch@kit.edu (A. Fritsch)

🆔 0000-0003-3747-4229 (M. Ullrich); 0000-0002-1441-295X (F. Aisenbrey); 0000-0002-2124-4720 (A. Fritsch)



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

As a result, the approach combines the systematic coverage and reproducibility of deterministic processing with the semantic flexibility of LLM-generated feedback. It provides detailed, element-specific formative feedback that accounts for alternative terminology and valid modeling decisions while making identified differences between the task description, the sample solution, and the submitted model transparent. Instructors can provide task-specific assessment instructions aligned with their learning objectives.

2. Assessment Approach and Grading Service

Our approach is implemented as an extension to KEA-Mod, a modular e-assessment system for higher education in which independently deployable grading services assess modeling submissions [14]. Grading services receive a modeling task, task-specific assessment instructions, a sample solution, and a submitted model and return assessment results. As a first assessment step, existing deterministic grading services check the syntactic correctness of the submission.

The proposed semantic grading service complements these syntactic checks for Entity–Relationship diagrams and returns element-specific formative feedback. For semantic assessment, the service converts the submitted diagram and sample solution into structured textual representations of their entities, relationships, attributes, edges, and relevant properties. It then determines semantic correspondences between the represented elements. The mapping may associate multiple elements of one diagram with a single element of the other diagram, allowing the service to account for different modeling choices—for example, when an attribute in one diagram is represented as a separate entity in the other.

Based on the resulting mapping, deterministic difference detection identifies unmatched elements and structural differences in the connections and properties of mapped elements. These differences are then evaluated by an LLM in the context of the task description, the sample solution, and the assessment instructions. The LLM determines whether a difference represents an error, an acceptable interpretation of an ambiguous task, or a different but valid modeling decision and generates corresponding element-specific feedback. For semantic interpretation, it can request targeted diagram views of the information required for the current step, such as selected element types, cardinalities, or identifying and weak-entity properties. This supports focused semantic reasoning and reduces the amount of irrelevant diagram information presented at once.

While the approach is in principle applicable to different modeling languages, the current implementation supports Entity–Relationship diagrams in Chen notation. Its feedback addresses missing or unnecessary elements, alternative representations and labels, key and weak-entity properties, identifying relationships, and relationship cardinalities. For each identified discrepancy, the relevant passages and model elements are highlighted across the task description, the submitted model, and the sample solution and linked to the corresponding feedback; this interaction is illustrated in the screencast referenced in Section 4.

3. Evaluation

The evaluation examines two complementary questions. First, is the generated feedback sufficiently actionable and internally consistent to support iterative model improvement? Second, does the fully supported LLM assessment improve the correctness and completeness of the generated feedback compared with direct LLM assessment?

While several LLMs were considered during development to compare diagram representations and refine the workflow, the final system configuration combines a locally deployable LLM with a state-of-the-art proprietary LLM. Google’s Gemma-4-26B performs the token-intensive but conceptually simpler diagram-mapping steps, whereas OpenAI’s GPT-5.4 performs semantic evaluation and feedback generation. In both evaluations, *fully supported LLM assessment* denotes the complete assessment approach with explicit diagram mapping, deterministic difference detection, and access to targeted

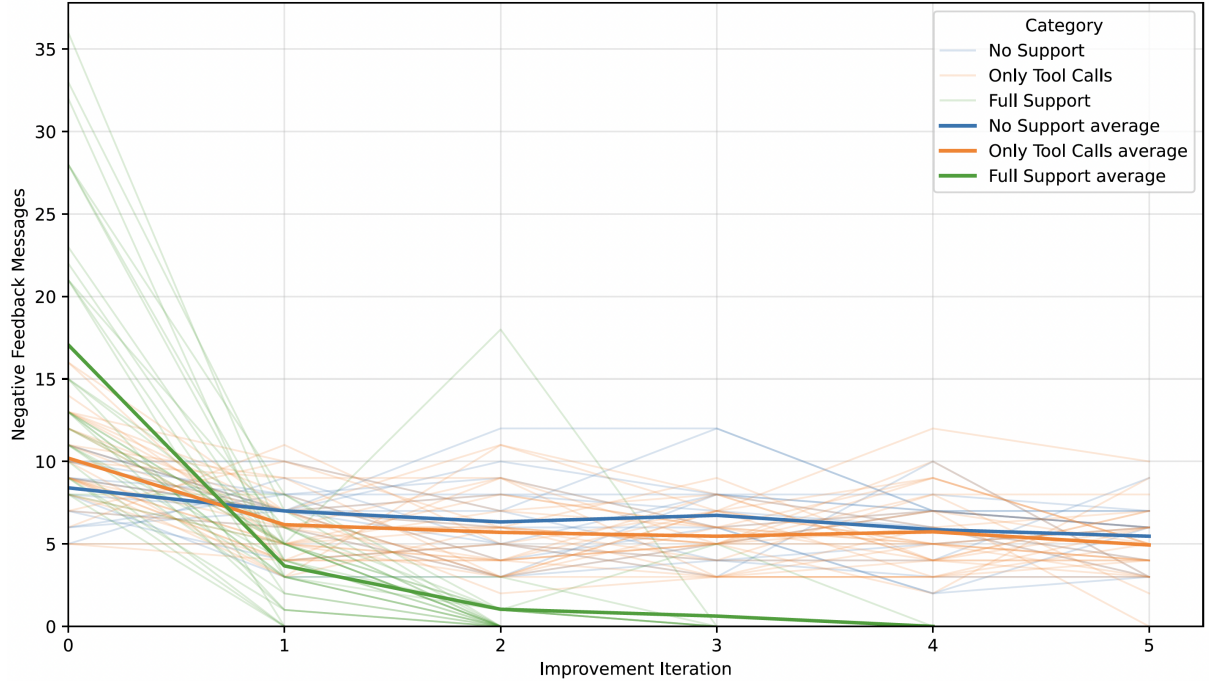


Figure 1: Remaining negative feedback after each improvement iteration, in which an LLM revised the diagram; thin lines represent individual submissions and thick lines the mean for each assessment configuration.

diagram views, while *direct LLM assessment* denotes the baseline in which the LLM assesses the complete submission without these supporting steps.

To evaluate actionability and consistency, GPT-5.4 received a submitted diagram and the feedback generated for it, but no task description, assessment instructions, or sample solution. The LLM was asked to revise the diagram based solely on the feedback. The revised diagram was then assessed again, and this process was repeated for up to six iterations. Three assessment configurations were compared: direct LLM assessment (*No support* in Figure 1), LLM assessment with diagram views only (*Only tool calls*), and fully supported LLM assessment (*Full support*).

As shown in Figure 1, only the fully supported LLM assessment consistently generated feedback that enabled successful diagram correction. Most submissions required no more than two revision iterations. In contrast, the configurations with diagram views only or without support showed only moderate improvement after the first iteration and little further progress thereafter. The LLM did not request any diagram views in the configuration in which they were available without diagram mapping. These results indicate that the feedback generated by the fully supported assessment is actionable and internally consistent. Since the same grading service generated the feedback and evaluated the revised diagrams, however, this experiment does not independently establish objective correctness or completeness.

A separate manual evaluation therefore assessed feedback quality directly. To this end, GPT-5.4 generated feedback for thirty randomly selected student submissions to one Entity-Relationship modeling task under the fully supported and direct LLM assessment configurations. The selected task was deliberately designed to cover a broad range of semantic difficulties and edge cases. Each generated feedback entry was inspected manually for correctness, while missing feedback entries were recorded to assess completeness. The results are summarized in Table 1.

The fully supported LLM assessment substantially reduced incorrect and missing feedback. At $\alpha = 0.05$, the differences in correctness and completeness were statistically significant.

Most remaining correctness issues resulted from deterministic edge cases. In particular, 12 of the 16 incorrect feedback entries occurred when a relationship was modeled with a different arity than in the sample solution, because cardinality feedback had been precomputed under the assumption of equal relationship arity.

Table 1

Mean feedback-quality issues per submission ($n = 30$); p is the one-sided Welch’s t -test p -value for the hypothesis that fully supported LLM assessment yields a lower mean.

	Fully supported LLM assessment	Direct LLM assessment	p
Incorrect feedback entries (e.g., incorrect cardinality advice or requesting an implemented change)	0.53	2.47	0.00126
Missing feedback entries (e.g., missed cardinality errors or unflagged extra elements not justified by the scenario)	0.20	7.33	0.00013

4. Demonstrator and Platform Integration

A standalone demonstrator is available at <https://semantic-demo.keamod.de/>. Its purpose is to expose the core functionality of the semantic grading service rather than to provide a polished end-user interface. To limit operating costs, the demonstrator uses Gemma-4-31B as its LLM instead of GPT-5.4, which was used for semantic evaluation and feedback generation in the evaluation. Several non-editable example tasks and solutions derived from an authentic higher-education context allow users to inspect representative feedback directly. Users can also create their own modeling tasks, sample solutions, and submissions and initiate semantic assessment for them. The result view presents the generated feedback together with highlighted scenario passages and the corresponding affected elements in both the sample solution and the submitted model. A short guide to using the demonstrator is provided in the accompanying screencast at <https://www.keamod.de/semantic-demo-screencast>.

The grading service is designed for integration into KEA-Mod, which provides a complete environment for creating modeling tasks, managing submissions, and assessing student solutions [14]. It includes user and role management for students and instructors and can be connected to institutional identity-management systems. Assessment timing and release can be configured for formative and summative settings: feedback may be generated and shown immediately, generated automatically but released only after manual review, or produced through manual assessment where automated grading is not appropriate or permitted. In this setting, the semantic grading service can support both direct formative feedback and instructor-led correction by identifying relevant model differences, proposing feedback, and making the basis of the assessment transparent.

5. Conclusion and Outlook

The results show that carefully structured LLM assistance can make semantic assessment a practical means of supporting conceptual modeling education at universities. By combining explicit diagram mapping and deterministic difference detection with LLM-based semantic interpretation, the grading service generates substantially more correct and complete feedback than direct LLM assessment and produces feedback that can guide iterative model revision. These results demonstrate the feasibility of the approach and provide a foundation for its integration into e-assessment environments. Integrated into a system such as KEA-Mod, the approach can provide timely formative feedback to students and transparent correction support to instructors.

While three Entity–Relationship diagrams of different sizes and complexity were considered when evaluating and refining the diagram representations, the final feedback-quality evaluation was limited to submissions for one modeling task. Further studies should therefore examine additional tasks and student populations, as well as mapping robustness, to assess the generalizability of the results.

Beyond this broader validation, the approach can also be extended to additional modeling languages. A first experimental transfer of the grading service to UML class diagrams yielded promising results. Extending it to behavioral modeling languages such as BPMN or Petri Nets will additionally require assessment of execution semantics rather than structural correspondence alone.

Declaration on Generative AI

The authors used OpenAI's GPT-5.6 to draft this paper based on a master's thesis. Generative AI was also used during the thesis project for language revision and coding assistance. All generated content was reviewed and revised by the authors, who take full responsibility for the paper's content.

References

- [1] M. Ullrich, C. Houy, T. Stottrop, M. Striewe, B. Willems, P. Fettke, P. Loos, A. Oberweis, Automated assessment of conceptual models in education: A systematic literature review, *Enterprise Modelling and Information Systems Architectures* (2023) 1–36. doi:10.18417/emisa.18.2.
- [2] L. E. Hall, A. Gordon, A virtual learning environment for entity relationship modelling, in: *Proceedings of the 29th SIGCSE Technical Symposium on Computer Science Education*, 1998, pp. 345–349. doi:10.1145/274790.274327.
- [3] F. Prados, I. Boada, J. Soler, J. Poch, A web-based tool for entity-relationship modeling, in: *Computational Science and Its Applications – ICCSA 2006*, 2006, pp. 364–372. doi:10.1007/11751540_39.
- [4] P. Thomas, N. Smith, K. Waugh, Automatically Assessing Graph-Based Diagrams, *Learning, Media and Technology* (2008) 249–267. doi:10.1080/17439880802324251.
- [5] J. Schramm, S. Strickroth, N.-T. Le, N. Pinkwart, Teaching UML skills to novice programmers using a sample solution based intelligent tutoring system, in: *Proceedings of the 25th International Conference of the Florida Artificial Intelligence Research Society*, 2012, pp. 472–477.
- [6] A. Weerasinghe, B. Evans, UML-IT: An ITS to teach multiple modelling tasks, in: *Artificial Intelligence in Education*, 2015, pp. 816–819. doi:10.1007/978-3-319-19773-9_123.
- [7] D. de Bari, G. Garaccione, R. Coppola, M. Torchiano, L. Ardito, Evaluating large language models in exercises of UML class diagram modeling, in: *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2024, pp. 393–399. doi:10.1145/3674805.3690741.
- [8] N. Bouali, M. Gerhold, T. Ul Rehman, F. Ahmed, Toward automated UML diagram assessment: Comparing LLM-generated scores with teaching assistants, in: *Proceedings of the 17th International Conference on Computer Supported Education*, 2025, pp. 158–169. doi:10.5220/0013481900003932.
- [9] S. Riaz, P. Rooshenas, LLM-driven feedback for enhancing conceptual design learning in database systems courses, in: *Proceedings of the 56th ACM Technical Symposium on Computer Science Education*, 2025, pp. 1001–1007. doi:10.1145/3641554.3701940.
- [10] M. B. Ibáñez, M. L. Barrón-Estrada, R. Zatarain-Cabada, Can multimodal large language models grade like an expert? a study on UML class diagram assessment accuracy, *Computer Applications in Engineering Education* (2025). doi:10.1002/cae.70080.
- [11] P. Ardimento, M. L. Bernardi, M. Cimitile, M. Scalera, A RAG-enhanced AI feedback for UML education, in: *Communications in Computer and Information Science*, 2025, pp. 12–18. doi:10.1007/978-3-031-99264-3_2.
- [12] B. I. Al-Ahmad, A. M. R. Alsobeh, O. Meqdadi, N. Shaikh, A student-centric evaluation survey to explore the impact of LLMs on UML modeling, *Information* (2025). doi:10.3390/info16070565.
- [13] C. Wang, B. Wang, P. Liang, J. Liang, Assessing UML diagrams by GPT: Implications for education, *Journal of Systems and Software* (2026). doi:10.1016/j.jss.2025.112709.
- [14] M. Ullrich, M. Forell, C. Houy, P. Pfeiffer, S. Schüler, T. Stottrop, B. Willems, P. Fettke, A. Oberweis, Platform architecture for the diagram assessment domain, in: *Software Engineering 2021 Satellite Events – Workshops and Tools & Demos*, 2021, pp. 1–8. doi:10.22028/D291-34119.