RESEARCH TOPICS

2026-01-09

AUTOMATIC ANALYSIS AND GRADING OF UTML UML DIAGRAMS

Douwe Osinga
d.r.osinga@student.utwente.nl

Supervisor
dr. ir. Vadim Zaytsev
v.zaytsev@utwente.nl

Supervisor
dr. Nacir Bouali
n.bouali@utwente.nl

Faculty of Electrical Engineering,
Mathematics, and Computer Science

UNIVERSITY OF TWENTE.

ABSTRACT

During computer science studies, students are often required to submit UML diagrams. The grading of these diagrams is often done by humans, resulting in a costly, lengthy, and error-prone process. In this paper, we investigate the theoretical feasability of automatically grading UML diagrams, focusing on the UTML variant developed at the University of Twente. In the final thesis, we compare the most suitable autograder from our related works to human grading.

1. INTRODUCTION

UML diagrams play a significant role in computer science, as they allow for communicating system designs in a standardised format. During technical studies, students are often required to make a UML diagram for a graded assignment or exam.

However, the grading of these diagrams can often be a costly and lengthy process, involving multiple paid members of staff [1][1].

Additionally, this process is prone to grading inconsistencies [1] , as humans are inherently unreliable when grading [2] . Letting the process of determining a grade based on a rubric be performed by a (deterministic) program instead of a human reduces these inconsistencies [3] .

In this Research Topics paper, I examine the current state of autograding diagrams and propose a plan for the implementation of *Seshat*, an automatic diagram grader that combines concepts from related works (Section 3), which is to be implemented and verified in the final thesis.

2. PROBLEM STATEMENT

The grading of (UML) diagram submissions by students can often be a costly and lengthy process, involving multiple paid members of staff[1] , which can take multiple hours of active work. Additionally, human grading is inherently subject to inconsistencies in grading, according to M. Meadows *et al.* [2] , who pose two possible solutions: either "report the level of reliability associated with marks/grades, or find alternatives to marking." We propose a third alternative: what if, instead of finding alternatives to marking/grading, we find alternatives to the grading *process*?

The automatisation of grading diagrams provides an grading marking method that could both reduce the cost and time required for institutions and reduce the inherently present inconsistencies in human grading[2] . This could result in similar performance compared to human grading in terms of **accuracy** and **process transparency**, while improving **consistency**.

With accuracy, we mean the percentage of points assigned to a submission that are prescribed by the rubric for a particular excercise. With consistency, we mean both the extent to which similar grades are given to similar submissions, and the difference between consecutive runs (i.e. determinism). With transparency, we mean the extent to which the reasoning for a particular grade is explained. These properties are desirable in the grading process, as it means that students are graded in a way that reflects their performance.

For this research, we focus on the automatic grading of *UTML* UML diagrams, a recent, in-house developed diagram format of the University of Twente [4], [5] . However, as UTML is just a representation format and tool for creating UML diagrams, we aim to generalise these results to provide advice on the automatic grading of UML diagrams as a whole.

2.1. Research Questions

In order to examin the feasibility of automatically grading UTML UML diagrams, we provide a main research question (**MRQ**):

To what extent can UML diagrams be graded automatically while keeping or improving the accuracy, consistency, and transparency of human grading?

[1] Also from personal experience.

[2] Given that the process is deterministic

We aim to answer the main research question with the following sub-research questions:

RQ1: What existing work can be found for automatically analysing and/or grading UML diagrams?
- **RQ1a**: What correction models are employed by existing works?
- **RQ1b**: To what extent can Intended Learning Objectives be translated into different types of autograder correction models?

RQ2: To what extent are existing solutions suitable for use in autograding UTML diagrams with regards to (1) accuracy, (2) consistency, (3) transparency, (4) availability of source code, (5) extent of linking ILOs to grading instructions, (6) ease of integration into the grading process, and (7) UTML support?

RQ3: To what extent can a suitable autograder be constructed from previous work to be able to grade UTML UML diagrams?

RQ4: To what extent does the autograder compare to human grading in the context of grading first-year UML exam questions?

RQ1 is answered in Section 3 , giving us an overview of existing solutions and their grading methodologies. **RQ2** is answered in Section 3 by analysing these works for suitability of grading. Finally, **RQ3** and **RQ4** are to be answered in the final thesis, where we grade UTML diagrams using an implementation based on related work and compare it to human grading.

3. RELATED WORK

In order to answer research questions **RQ1** until **RQ4**, we have conducted a small-scale study covering roughly 40 works. These works were collected from sources such as Google Scholar[3] and ResearchGate[4], using terms such as "automatically grading UML diagrams", "autograder diagram", and "UML diagram assessment" for autograder-based related works, and terms such as "ILO translation", "intended learning objective grading", and more terms and stuff about ILOs

3.1. Autograders

3.1.1. Frameworks / Theoretical

[6], [7], [8], [9], [10]

N. Smith *et al.* [6] provides a five-step framework for assessing "possibly ill-formed or inaccurate diagrams" that include (1) segmentation, (2) assimilation, (3) identification, (4) aggregation, and (5) interpretation. While the first two steps are aimed at translating images or other "raster-based input" into diagrammatic primitives, the latter stages provide a foundation to grade diagrams used by other papers [11] .

N. H. Ali *et al.* [7] proposes a UML class diagram assessment system using Rose Petal files, but does not mention enough specifics about algorithms to warrant further investigation.

F. Batmaz [8] takes a broader look at the process of grading, identifying and developing techniques to reduce repetitive actions, focusing on database Entity Relation diagrams. It proposes a semi-automatic grading system which identifies identical segments between a submission and the solution. Assuming multiple submission revisions are available, it suggests to "not only [use] the reference text but also the intermediate diagrams" for identifying semantic matches [8, p.40] .

V. Vachharajani *et al.* [10] proposes a UML use case assessment architecture. It provides a useful catalogue about edge cases related to (use case) diagram assessment, such as the chance of misspellings, synonyms, abbreviations, directionality of relationships, etc.

W. Bian *et al.* [9] establishes a metamodel to map submissions to example solutions and proposes a metamodel to grade submissions. It suggests using syntactic matching, semantic matching, and structural matching, with the goal to optimally match parts of a student submission with those of a teacher, considering spelling mistakes, synonyms and related words, and neighbours / inheritance, respectively.

[3] https://scholar.google.com

[4] https://www.researchgate.net

3.1.2. Non-ML/LLM

The automatic analysis of diagrams seems to be a relatively new field, having started somewhere in the early 2000s [12] . Multiple types of diagrams are researched, including UML class and use case diagrams [3], [13], [14], [15], [16], [17], [18], [19]

and Entity-Relation Diagrams [11], [12], [20], [21], [22], [23], [24], [25], [26] .

2020 expands their previous work [9] with a case study. Their main findings are that multiple teacher solutions result in more accurate grades, that grading configurations change per exam if you want similar grades to the teacher, and that their autograding "has shown to be more consistent and able to ensure fairness in the grading process" [3, p.11] .

M. Hosseinibaghdadabadi *et al.* [13] implements the work of W. Bian *et al.* [9] by comparing UML use case diagrams to one or multiple example solutions, preferring the maximum grade. It uses a graph similarity strategy which matches nodes based on structural matching, along with syntactic and semantic word matching. Syntactic matching with Levenshtein distance, semantic matching with WordNet similarity score (uses HSO, WUP, LIN metrics). It achieves high correlation with human grades. more?

O. Anas *et al.* [14] compares UML class diagram submissions to an example solution. It uses graph similarity scores based on structural matching along with syntactic and semantic matching. Syntactic matching is done with substring matching, semantic matching is done with neighbour similarity ("the comparison of the neighboring classes" [14, p.1585]), relationship name, type, multiplicity, and inheritance. It achieves high correlation with human grading (more than 80% is perfectly similar, over 90% had a correlation >0.85, no correlation was lower than 0.7).

Multiple papers mention the use of XMI [15], [16] , the object notation standard by OMG [27] , or Rose Petal files [17] , the standard of IBM Rational Rose [28] , but fail to mention specifics about matching algorithms or results.

H. AlRawashdeh *et al.* [18] provides an interesting alternative way of grading submissions: by means of combining many UML diagram validators, model checkers, and even LTL properties given by instructors, but a clear purpose, scope, and results are lacking from the paper.

M. Striewe *et al.* [19] continues this trend by focusing on graph queries for evaluation, providing a Domain-Specific Language that looks relatively similar to SQL. While it looks promising, the fact that teachers would have to learn a query language and transform their existing rubrics/example solutions into this format could be a real hurdle, especially given the high similarity to existing grading of graph-isomorphism-based solutions. Additionally, the paper does not provide approximate matching that would account for misspelling or synonyms.

S. Foss *et al.* provide multiple papers on AutoER, a database diagram generator and evaluator that provides direct interaction with a description text [20], [21], [22] . Unfortunately, concrete comparisons to manual grading or source code could not be found.

P. Thomas also provides a selection of paperse on the automatic grading of database diagrams [11], [12], [23], [24], [25] . These papers provide a grading strategy that accounts in its basis for *imprecise* diagrams (diagrams containing misspellings, duplicate entities, etc.), basing their analysing on comparing ever increasing subsets of the graph ((Minimal) Meaningful Units) based on the work of N. Smith *et al.* [6] . By 2009 , P. Thomas *et al.* manage to achieve a correlation to human grading of 92%, along with statistically proving that the autograder grades more consistently than human grading.

In 2011, P. Thomas *et al.* provide an online platform for both students and teachers to ease the process of automatic grading further, also used by N. Smith *et al.* [26] , which fruther mathematically specify P. Thomas *et al.* 's work. Unfortunately, we were not able to retrace the source code of this grader.

3.1.3. ML/LLM-driven

Work on AI [29], [30] .

N. Bouali *et al.* [29] uses various Large Language Models (Llama, GPT o1-mini, Claude) to grade

Further proof of unreliability of using Large Language Models (LLMs) for automatic grading: "In the evaluation based on UC4, GPT deducts points for missing relationships between specified actors and use cases, but theses relationships existed in the UML use case" [31, p.13] , and "While the models would provide a final score as requested in the prompt's response format, this core often did not match the actual sum of points awarded in their criterion-by-criterion assessment, where N. Bouali *et al.* [29, p.164] identify the problem perfectly, stating that "This discrepancy can be attributed to the autoregressive nature of LLMs, where they generate responses token by token".

I believe that the observation from N. Bouali *et al.* [29] highlights the underlying problem of using LLMs for automatic grading. Because these models are in their very essence based on predicting tokens [32] , there is no formal guarantee that grades are produced with accuracy. The fact that LLMs produce grades that correlate with human grading does not mean that this grading is done in a fair, consistent, or reliable manner. In particular, reliability is affected by the nondeterminism introduced into LLMs, either deliberately, with 'temperature' controls per model, or accidentally, because batch processing ordering for large-scale LLM deployments can introduce nondeterminism [33], [34] .

While [29] attempts to lower the amount of nondeterminism by setting the model's temperature to zero, nondeterminism can still occur due to

Nondeterminism of AI [33], [34], [35] + counterarg: inherent lack of transparency, risks of nondeterminism in grading (see sources) == bad because same solution might not give same grade), lack of consistency (contexxt window, importance of reducing prompt length, …).

3.2. ILO translation

4. TOOLS AND TECHNIQUES

Adopt existing tool(s), make own tool, what frameworks/languages, …

5. PLANNING

TODO: Graduation planning. Phases, goals per phase.

BIBLIOGRAPHY

[1] F. Ahmed, N. Bouali, and M. Gerhold, "Teaching Assistants as Assessors: An Experience Based Narrative," 2024. [Online]. Available: https://research.utwente.nl/files/457355611/126242.pdf

[2] M. Meadows and L. Billington, "A Review Of THe Literature On Marking Reliability." [Online]. Available: https://assets.publishing.service.gov.uk/media/5a820a57e5274a2e87dc0d5a/0505_Meadows_and_Billington_CERP_RP.pdf

[3] W. Bian, O. Alam, and J. Kienzle, "Is automated grading of models effective?: assessing automated grading of class diagrams," in *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*, in MODELS '20. ACM, Oct. 2020, pp. 365–376. doi: 10.1145/3365438.3410944 .

[4] D. Huistra, "UTML - internal GitLab repository." [Online]. Available: https://gitlab.utwente.nl/ewi/eduapps/UTML/

[5] "UTML - old student repository." [Online]. Available: https://github.com/andrewjh9/UTML

[6] N. Smith, P. Thomas, and K. Waugh, "Interpreting imprecise diagrams," in *Diagrammatic Representation and Inference*, in International Conference on Theory and Application of Diagrams. Mar. 2004, pp. 239–241. doi: 10.1007/978-3-540-25931-2_24 .

[7] N. H. Ali, Z. Shukur, and S. Idris, "Assessment System For UML Class DiagramUsing Notations Extraction," 2007. [Online]. Available: https://www.researchgate.net/profile/Zarina-Shukur/publication/253243639_Assessment_System_For_UML_Class_Diagram_Using_Notations_Extraction/links/55487af30cf2b0cf7acec2e4/Assessment-System-For-UML-Class-Diagram-Using-Notations-Extraction.pdf

[8] F. Batmaz, "Semi-Automatic Assessment ofStudents' Graph-Based Diagrams," 2010. [Online]. Available: https://www.academia.edu/download/66135135/70_22270_EM_26aug_20feb_L.pdf

[9] W. Bian, O. Alam, and J. Kienzle, "Automated Grading of Class Diagrams," in *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, IEEE, Sept. 2019, pp. 700–709. doi: 10.1109/models-c.2019.00106 .

[10] V. Vachharajani and J. Pareek, "A Proposed Architecture for Automated Assessment of Use Case Diagrams," in *International Journal of Computer Applications (0975 – 8887)*, 2014. [Online]. Available: https://www.academia.edu/download/67672696/pxc3900193.pdf

[11] P. Thomas, N. Smith, and K. Waugh, "Automatically Assessing Diagrams," in *Proceedings of the IADIS International Conference on e-Learning*, 2009. [Online]. Available: https://www.researchgate.net/profile/Pete-Thomas/publication/42799920_Automatically_assessing_diagrams/links/0fcfd5060076dd8ba2000000/Automatically-assessing-diagrams.pdf

[12] P. Thomas, "Grading Diagrams Automatically," 2004. [Online]. Available: https://oro.open.ac.uk/90155/1/2004_01.pdf

[13] M. Hosseinibaghdadabadi, O. A. N. Almerge, and J. Kienzle, "Automated Grading of Use Cases," in *2023 ACM/IEEE 26th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, IEEE, 2023. [Online]. Available: https://ieeexplore.ieee.org/iel7/10343461/10343549/10343598.pdf

[14] O. Anas, T. Mariam, and L. Abdelouahid, "New method for summative evaluation of UML class diagrams based on graph similarities," 2021. [Online]. Available: https://www.academia.edu/download/66135135/70_22270_EM_26aug_20feb_L.pdf

[15] R. Jebli, J. E. Bouhdidi, and M. Y. Chkouri, “Assessing Students' UML Class Diagrams: a NewAutomated Solution,” in *2023 7th IEEE Congress on Information Science and Technology (CiSt)* |, IEEE, 2023. [Online]. Available: https://ieeexplore.ieee.org/iel7/10409867/10409868/10409936.pdf

[16] S. Modi, H. A. Taher, and H. Mahmud, “A Tool to Automate Student UML diagram Evaluation,” 2021. [Online]. Available: https://www.academia.edu/download/72488756/575.pdf

[17] N. H. Ali, Z. Shukur, and S. Idris, “A Design of an Assessment System for UML Class Diagram,” in *2007 International Conference on Computational Science and its Applications (ICCSA 2007)*, IEEE, Aug. 2007, pp. 539–546. doi: 10.1109/iccsa.2007.2 .

[18] H. AlRawashdeh, S. Idris, and A. M. Zin, “Using Model Checking Approach for Grading the Semantics of UML Models,” 2014. [Online]. Available: https://iieng.org/images/proceedings_pdf/8684E0114567.pdf

[19] M. Striewe and M. Goedicke, “Automated Checks on UML Diagrams,” in *ITiCSE'11*, in ITiCSE '11. ACM, June 2011, pp. 38–42. doi: 10.1145/1999747.1999761 .

[20] S. Foss, T. Urazova, and R. Lawrence, “Learning UML database design and modeling with AutoER,” in *Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings*, in MODELS '22. ACM, Oct. 2022, pp. 42–45. doi: 10.1145/3550356.3559091 .

[21] S. Foss, T. Urazova, and R. Lawrence, “Automatic Generation and Marking of UML Database Design Diagrams,” in *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, in SIGCSE 2022. ACM, Feb. 2022, pp. 626–632. doi: 10.1145/3478431.3499376 .

[22] S. Foss, “AutoER: A System for the Automatic Generation and Evaluation of UML Database Design Diagrams,” 2022. [Online]. Available: https://open.library.ubc.ca/media/download/pdf/24/1.0421624/4

[23] P. Thomas, N. Smith, and K. Waugh, “An approach to the automatic grading of imprecise diagrams,” technical report, 2006. doi: .org/10.21954/ou.ro.00016046 .

[24] P. Thomas, N. Smith, and K. Waugh, “Automatically assessing graph-based diagrams,” *Learning, Media and Technology*, vol. 33, no. 3, pp. 249–267, 2008, doi: 10.1080/17439880802324251 .

[25] P. Thomas, K. Waugh, and N. Smith, “Generalised Diagramming Tools with Automatic Marking,” in *Innovation in Teaching and Learning in Information and Computer Sciences*, 2011. doi: 10.11120/ital.2011.10010022 .

[26] N. Smith, P. Thomas, and K. Waugh, “Automatic Grading of Free-Form Diagrams with Label Hypernymy,” in *2013 Learning and Teaching in Computing and Engineering*, IEEE, Mar. 2013, pp. 136–142. doi: 10.1109/latice.2013.33 .

[27] OMG, “XMI specification.” [Online]. Available: https://www.omg.org/spec/XMI

[28] IBM, “Rational Rose.” [Online]. Available: https://www.ibm.com/support/pages/ibm-rational-rose-enterprise-7004-fix-pack-4-7000

[29] N. Bouali, M. Gerhold, T. U. Rehman, and F. Ahmed, “Toward Automated UML Diagram Assessment: Comparing LLM-Generated Scores with Teaching Assistants,” 5220. [Online]. Available: https://research.utwente.nl/files/496461589/134819.pdf

[30] D. R. Stikkolorum, P. van der Putten, C. Sperandio, and M. R. Chaudron, “Towards Automated Grading of UML Class Diagrams with Machine Learning,” 2019. [Online]. Available: https://ceur-ws.org/Vol-2491/paper80.pdf

[31] C. Wang, B. Wang, P. Liang, and J. Liang;, “Assessing UML Diagrams by GPT: Implications for Education,” technical report, 2025. [Online]. Available: https://www.researchgate.net/publication/397720325_Assessing_UML_Diagrams_by_GPT_Implications_for_Education

[32] A. F. Ferraris, D. Audrito, L. D. Caro, and C. Poncibò, “The architecture of language: Understanding the mechanics behind LLMs,” *Cambridge Forum on AI: Law and Governance*, vol. 1, pp. 1–19, 2025, doi: 10.1017/cfl.2024.16 .

[33] M. Brenndoerfer, “Why Temperature=0 Doesn't Guarantee Determinism in LLMs,” 2025, [Online]. Available: https://mbrenndoerfer.com/writing/why-llms-are-not-deterministic

[34] B. Atil *et al.*, “Non-Determinism of “Deterministic” LLM Settings,” 2025. [Online]. Available: https://arxiv.org/pdf/2408.04667

[35] H. He and T. Machines, “Defeating Nondeterminism in LLM Inference,” 2025, [Online]. Available: https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/

6. APPENDICES

6.1. Autograder suitability table

Author	Di	Ac	Co	Tr	OSS	ILO	Int	UTML
M. Hosseinibaghdadabadi *et al.* [13]	UML Use Case	**H**	**H**	**H**	**N**	**N**	?	**N**

Table 1: Autograders and their suitability scores.

*Di(*agram type*), Ac(*curacy*), Co(*nsistency*), Tr(*ansparency*), OSS = *availability of source code*, ILO = *ease of linking grading to ILOs*, Int(*egration ease*), UTML *support*.

Scoring is divided into “N” (*No Support*), “L” (*Low*), “M” (*Medium*), “H” (*High*), and “?” (*Unknown*), which gives an indication of suitability w.r.t. that particular criterium.