RESEARCH TOPICS 2025-12-19

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 at some point 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] .

Additionally, this process is prone to errors, as humans are inherently unreliable when grading [1] .

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 and is additionally subject to inherent inconsistencies in grading [1] . M. Meadows *et al.* [1] 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 grading, we find alternatives to the grading *process*?

The automatisation of grading diagrams provides an alternative marking method that could reduce the cost and time required for universities and other institutions, along with reducing the inherently present inconsistencies in human grading (given that the process is deterministic). This could result in similar performance compared to human grading in terms of **accuracy** and **transparency**, while improving **consistency**.

With accuracy, we mean the percentage of points correctly assigned to a submission, according to the rubric for a particular excercise. With consistency, we mean the similarity 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. maybe more explanation about this?

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 [2], [3] . However, a UTML is just a representation format and creation tool for UML diagrams, we aim to generalise these results for 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

[1] From personal experience.

improving the accuracy, consistency, and transparency of human grading?

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

RQ1: What existing work exist 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 to which linking ILOs to grading instructions is possible, (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 by collecting related work (Section 3), which will give 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 aim to grade UTML diagrams with the most suitable autograder 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[2] and ResearchGate[3] , 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. Existing work

The automatic analysis of diagrams seems to be a relatively new field, having started somewhere in the early 2000s [4] . Multiple types of diagrams are researched, including UML diagrams [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] , Entity-Relation Diagrams (including UML ERDs) [4], [17], [18], [19], [20], [21], [22]

More focused on interactivity: [18]

Work on AI [23], [24]

3.2. ILO translation

3.3. Suitability of autograders

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" [25, 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.* [23, 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.* [23] highlights the underlying problem of using LLMs for automatic grading. Because these models are in their very essence based on predicting tokens [26] , 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 [27], [28] .

While [23] attempts to lower the amount of nondeterminism by setting the model’s temperature to zero, nondeterminism can still occur due to

Nondeterminism of AI [27], [28], [29] + 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, …).

Experience on TAs [30]

Reliability of human marking/grading in general [1]

4. TOOLS AND TECHNIQUES

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

5. PLANNING

TODO: Graduation planning. Phases, goals per phase.

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

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

BIBLIOGRAPHY

[1] 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

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

[3] “UTML - old student repository.” [Online]. Available: https://github.com/andrewjh9/UTML

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

[5] 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

[6] 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

[7] 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

[8] 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 .

[9] 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 .

[10] 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

[11] 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

[12] 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

[13] 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 .

[14] 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

[15] 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

[16] 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 .

[17] 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 .

[18] 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

[19] 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 .

[20] 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

[21] 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 .

[22] 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 .

[23] 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

[24] 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

[25] 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

[26] 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 .

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

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

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

[30] 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

6. APPENDICES

6.1. Autograder suitability table

Author	Di	Ac	Co	Tr	OSS	ILO	Int	UTML
M. Hosseinibaghdadabadi *et al.* [5]	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.