Assessing Knowledge Through Written Reviews of First-Year Programming Students

Emil Folino Blekinge Tekniska Högskola
emil.folino@bth.se

Abstract

*In this paper a method of qualitative assessment of programming students is investigated. The qualitative assessment is done by reading students' review texts from three subsequent courses' individual programming project. The review texts are analyzed according to the SOLO Taxonomy and the students are awarded a SOLO level of **Unistructural**, **Multistructural** or **Relational**. The SOLO level is compared to the final grade of the three courses and a correlation between a student's final grade and the SOLO level is shown. Furthermore a positive progression in the students' comprehension and understanding of the course material is observed through the three subsequent courses. A recommendation is given to complement programming exercises with written assignments where the students get an opportunity to reflect and expand on the completed exercises.*

I. Introduction

Quantitatively assessing students in programming and computer science courses is often easy to do. Have the students completed the assigned tasks? Does the application work as it is supposed to do? Does any software tests fail? These questions are easily answered by automated procedures or a simple yes or no. However assessing the students' comprehension and understanding of the completed assignments and carry out a qualitative evaluation is harder [1]. The SOLO Taxonomy was proposed by Biggs and Collis in 1982. SOLO is abbreviated from Structure of the Observed Learning Outcome and the SOLO Taxonomy is used to qualitatively assess students' assignments. The SOLO Taxonomy consists of five levels of understanding: *Prestructural*, *Unistructural*, *Multistructural*, *Relational*, and *Extended Abstract*. In section III the five levels of understanding are exemplified by the students' review texts.

In [2] McCracken et al. evaluated first year Computer Science students' programming competency. Several universities took part in the study that showed disappointing results. To reverse the disappointing results a framework outlining the expectations of first year Computer Science students is proposed. However, the assessment is carried out in a quantitative way and there are no recommendations for any qualitative assessment methods.

Paulo Blikstein [3] uses snapshots of students' code during a programming assignment together with data mining to automate a technique to assess and analyze students learning programming. It is concluded that together with other data sources (Blikstein propose interviews, tests and surveys) the automated assessment can give insights into the understanding of the learning of programming. The assessment of the students' code is done entirely with quantitative methods. However, the proposed interviews, tests and surveys can give qualitative insights into the students' learning.

In [4] Lister et al. study written and think-aloud responses to exam questions. The authors analyze the responses according to the SOLO Taxonomy and conclude that experienced programmers more frequently answered with SOLO Relational responses compared to novice programmers' Multistructural responses. Lister et al. recommend that the students are given written assignments together with programming assignments. The written assignments make it easier to evaluate the understanding and comprehension the students obtain in programming courses.

1

The study in this paper builds upon the results outlined by Lister et al. in [4] by analyzing review texts written by first-year programming students. The review texts are handed in as a complement to the source code of programming projects. The review texts together with weekly assignments and the project forms the basis for the final grade given in the course.

II. Method

In the daily teaching at Webbprogramming (db-webb.se) at Blekinge Tekniska Högskola (BTH) the students hand in programming assignments each week. Together with the exercises they hand in a written review, answering 3-5 questions centered around the topics and assignments of the week. At the end of each 10 week study period the students hand in an individual project together with an extended review text. The students are graded both with regards to the completed assignments and the review texts. The following web pages [5] and [6] from the educational program's website explains how the students are graded on their review texts according to the SOLO Taxonomy.

In this report the review texts of three subsequent courses' projects are analyzed and analyzed according to the SOLO Taxonomy. The three courses are given in study period 1, 2, and 3. The three courses are "Databases, HTML, CSS and script based PHP-programming" (htmlphp), "Programming with JavaScript, HTML and CSS" (javascript1) and "Programming webservices in Linux" (linux)[1]. The htmlphp course is a beginners course and the course starts at a very basic level. The students' programming skills range from complete novices to more experienced programmers looking to broaden their skills. The javascript1 and linux courses builds upon the knowledge and practical skills obtained in prior courses.

The collection of review texts is done with a

---

[1]The course names have been translated by the author from "Databaser, HTML, CSS och skriptbaserad PHP-programmering", "Programmering med JavaScript, HTML och CSS" and "Programmera webbtjänster i Linux".

web scraper implemented in python. The web scraper fetch the review texts from the students published projects. The review texts are stored in a database together with the website URL of the published project and a traceable reference to the student. The review texts are fetched in a manner that removes the names and student acronyms from the review texts to ensure anonymity in most cases.

The SOLO analysis is done manually by the author by reading the review texts. The SOLO analysis is done anonymously, but traceable. The students are given a grade of 1-5 according to the five levels of the SOLO Taxonomy, *Prestructural* (1), *Unistructural* (2), *Multistructural* (3), *Relational* (4), and *Extended Abstract* (5).

In [1] Biggs and Collis give recommendations for applying the SOLO Taxonomy to different teaching subjects. Programming is not mentioned among the subjects but the authors give examples of application in technical subjects for example elementary mathematics. For the study reported in this paper the SOLO analysis was adapted to the technical language used in the students' review texts. In section III an explanation of each SOLO level is shown together with examples from the students' review texts.

The analysis of the review texts are done in a web form and the SOLO level is stored in the same database table as the review texts. The web form removes all styling done by the students and due to the way the collection of review texts are done this further ensures the anonymity of the students. The anonymity of the students is important to ensure that the SOLO analysis is done without any reference to the final grade given to the students in the courses.

After the review texts have been analyzed the SOLO level will be compared to the final grade of the course. The students' final grade will be fetched and stored in another database table together with the same traceable reference to the students. The final grade for the course and the SOLO level can now be compared and analyzed to evaluate if there is a correlation between the SOLO level and the final grade in the course.

The courses are graded based on the following criteria with a maximum of 100 points in total. 30 points for the completion of the six weekly assignments, 10 points for weekly extra assignments and extraordinary weekly review texts. 10 points for each mandatory requirement of the project and 10 points for each optional requirement of the project. The students are awarded credits on the ECTS-credit scale where more than 90 points equals an A, more than 80 equals B etc.

As the review texts are taken from three subsequent courses the progression of the students' understanding of the course material and programming in general can be investigated.

The web scraper and analysis web form can be found at the author's Github page [2].

III. Examples of SOLO levels

In this section examples of how the review texts are mapped to each level of the SOLO Taxonomy are shown. The review texts have been translated from Swedish to English by the author. The original Swedish texts are found in appendix A.

i. Prestructural

No answer more than repeating the question. The student is failed based on the text. The Prestructural SOLO level is used for students that have not handed in the projects.

ii. Unistructural

The answer contains no technical description of how the solutions have been implemented.

Example: "The search feature has it's own page (accessible from the navbar) where you can search for articles and object descriptions with a word consisting of letters(a-ö) and numbers(0-9). The search results are presented in a list and the first result in an article/object description is marked with and part of the text."

iii. Multistructural

The student gives a technical explanation of the implementation, more or less line by line and does not relate the implementation to other parts of the code or prior exercises in the courses.

Examples: "For every iteration in the loop an object is appended to slar.json. So when the loop is done it was just adding the last parenthesis and clean the file so it is valid JSON. I thought this requirement was kind of complicated and my solution is absolutely not the fastest but it does what it is supposed to do and that should suffice."

"I have tried to use a lot of built-in functions like 'map', 'reduce', 'filter', etc. to get the correct information from the arrays I use."

iv. Relational

The student gives a technical explanation of the implementation and justifies their choices through related course material or real-world examples.

Examples: "I chose to not split my client as it is done in the Gomoku assignment [3]. I know that the reason is to separate general and domain-specific code, but I am not going to extend on this client so I have decided to put all the code in the same file."

"With earlier assignments' clients as a base I did a client that tests the server."

v. Extended Abstract

No examples of Extended Abstract texts were found in the review texts. The students are first-year students and are not asked to produce novel material.

---

[2]https://github.com/emilfolino/pedagogy

[3]A programming assignment earlier in the course

IV. Results

In table 1 the SOLO level distribution is shown for the three courses: htmlphp, javascript1 and linux. In table 2 a relative distribution of the SOLO levels is shown.

Table 1: *No of SOLO Grades for the Courses*

Course	Relational	Multi	Uni	Total
htmlphp	4	13	5	22
javascript1	6	8	2	16
linux	9	9	0	18

Table 2: *Relative distribution of SOLO Grades for the Courses*

Course	Relational	Multi	Uni
htmlphp	18%	59%	23%
javascript1	37.5%	50%	12.5%
linux	50%	50%	0%

The total number of students in figures 1, 2 and 3 and tables 3 and 4 does not correspond with the number of students for each course in table 1. Not all students have received a grade in the courses because the students have not completed the mandatory oral presentation of their project.

Figure 1 shows the SOLO level compared to the grade that the student obtained in the htmlphp-course.
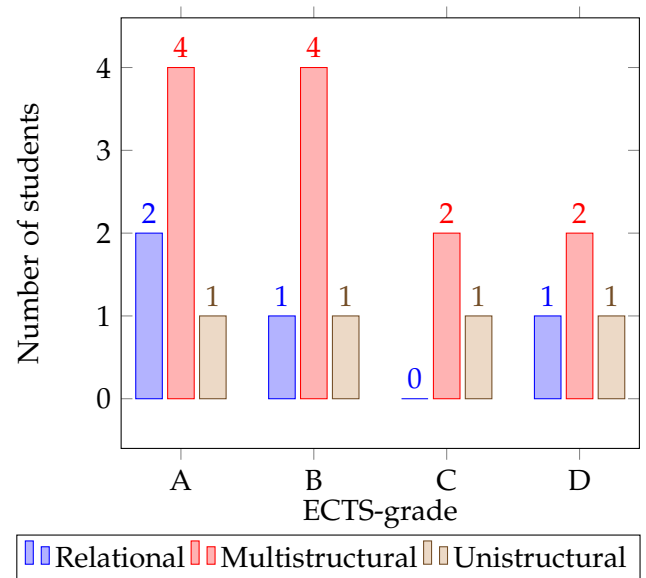

Figure 1: *SOLO level compared to course grade in htmlphp.*

Figure 2 shows the SOLO level compared to the grade that the student obtained in the javascript1-course.
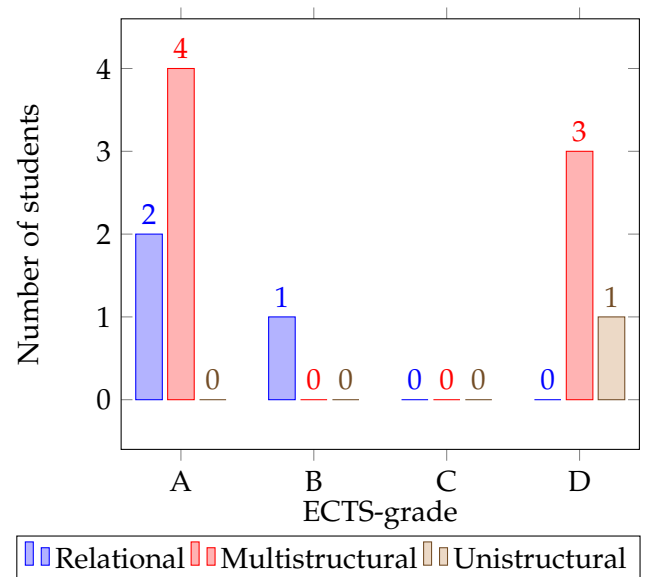

Figure 2: *SOLO level compared to course grade in javascript1.*

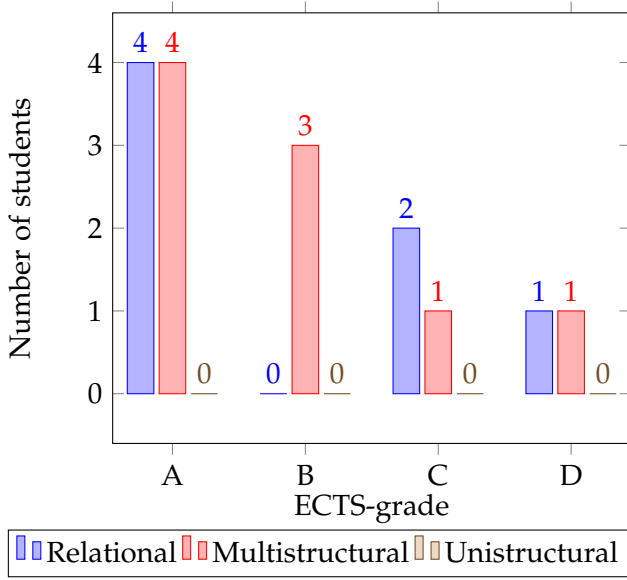Figure 3 shows the SOLO level compared to the grade that the student obtained in the linux-course.

Figure 3: *SOLO levels compared to course grade in linux.*

A comparison of SOLO levels and course grades are summarized in table 3 and a relative comparison of SOLO levels and course grades is shown in 4.

Table 3: *Comparison of SOLO levels and course grades.*

Grade	Relational	Multi	Uni
A	8	12	1
B	2	7	1
C	2	3	1
D	2	6	2

Table 4: *Comparison of SOLO levels and course grades shown relative to the number of students receiving the grade.*

Grade	Relational	Multi	Uni
A	38%	57%	5%
B	20%	70%	10%
C	33%	50%	17%
D	20%	60%	20%

V. Discussion

In table 3 and 4 it can be observed that students receiving a final grade of A in the courses are more likely to hand in a Relational review text than students with a final grade of B or D. Relatively few students received a final grade of C and therefore the grades are skewed towards a higher number of Relational review texts. Despite the skewed distribution of grades students receiving a C there is relatively more A students handing in Relational review texts. Furthermore it can be observed that the students with a final grade of A to a lower degree than any other final grade answers with a Unistructural review text. This confirms the correlation between a higher grade and a deeper understanding and comprehension of the course material.

As a teacher you hope to see a positive trend in your students comprehension of the course material. The SOLO Taxonomy can be used to evaluate the comprehension and understanding that the students have obtained [1]. The htmlphp in study period 1 is given as a beginners course in programming, but the actual level differs from complete novices to more advanced programmers. At the end of the linux course in study period 3 the students have studied programming courses worth 45 ECTS-credits. The students are now familiar with at least 10 programming languages and technologies and have moved from novices to more advanced programmers. According to Lister et al. in [4] it is expected that advanced programmers answer with Relational answers and the results of this study confirms the results in [4]. In tables 1 and 2 it is observed that the number of Relational review texts increase as the students progress through the three subsequent courses. Furthermore in tables 1 and 2 it is shown that the relative number of students answering with Multistructural review texts is constant and it is the Unistructural review texts that decreases as the Relational review texts increase. This further strengthens the observation that the students have a higher comprehension and understanding of the course material and programming in general.

During the study three of the author's colleagues read a subset of 20% of the texts and did their own analysis. The SOLO level for three of the 12 analysis done by the author's colleagues differed by one level. In all three cases the au-

thor had given the students a higher SOLO level than the colleagues. The colleagues highlighted that the analysis were difficult as the written texts are long form and parts of the texts would be a Multistructural review text and other parts is a Relational review texts. The author experienced the same classification problems during the analysis of the entire dataset. Shorter and more concise answers to specific problems would yield more easily analyzed and classifiable material.

VI. Conclusion

In [4] the authors Lister et al. conclude that as a complement to programming exercises students should be assessed on written or think-aloud responses to programming assignments. The study described in this paper confirms that the final grade in programming courses correlate with the level of understanding of the course material. Therefore written assignments or review texts together with programming assignments can be used as a qualitative method in the assessment process. A recommendation is to complement programming exercises with written assignments where the students get an opportunity to reflect and expand on the completed exercises. This introduces a qualitative complement to the conventional quantitative assessment methods.

Furthermore it is observed that the general level of comprehension and understanding of programming increases during the three subsequent courses that were analyzed in this paper. At the beginning of the educational program and the htmlphp course the level of programming skills varies a lot in the student cohort. At the end of the study period 3 and the linux course we see a more homogenized student cohort which further strengthens the notion that the general level of understanding and comprehension of the course material and programming in general is higher.

VII. Future Work

Based on the observations of the both the author and the author's three colleagues a more suitable dataset would be shorter more concise answers to specific questions. Another study conducted with this type of dataset would be more easily analyzed and probably yield even clearer results.

VIII. Acknowledgements

References

[1] Biggs JB, Collis KF. Evaluation the Quality of Learning: The SOLO Taxonomy (structure of the Observed Learning Outcome). Educational psychology. Academic Press; 1982.

[2] McCracken M, Almstrum V, Diaz D, Guzdial M, Hagan D, Kolikant YBD, et al. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. ACM SIGCSE Bulletin. 2001;33(4):125–180.

[3] Blikstein P. Using Learning Analytics to Assess Students' Behavior in Open-ended Programming Tasks. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge. LAK '11. New York, NY, USA: ACM; 2011. p. 110–116.

[4] Lister R, Simon B, Thompson E, Whalley JL, Prasad C. Not seeing the forest for the trees: novice programmers and the SOLO taxonomy. ACM SIGCSE Bulletin. 2006;38(3):118–122.

[5] Roos M. Att skriva en bra redovisningstext;. Accessed: 2017-02-10. `https://tinyurl.com/ldk7jon`.

[6] Roos M. Varför regnar det på bergssidan - ett exempel på SOLO taxonomin;. Accessed: 2017-02-10. `https://tinyurl.com/mv3uewr`.

A. Original Swedish Texts

The original Swedish review texts and their corresponding English translations is shown below.

Original Swedish Text: Sökfunktionen finns som egen sida(via navbar) där man kan söka i artiklar och i objektsbeskrivningar med ett ord som utgörs av bokstäver(a-ö) och siffror(0-9). Träffarna presenteras i en lista och första träffen i en artikel/objekt markeras med gult och del av texten.

Translation: The search feature has it's own page (accessible from the navbar) where you can search for articles and object descriptions with a word consisting of letters(a-ö) and numbers(0-9). The search results are presented in a list and the first result in an article/object description is marked with and part of the text.

Original Swedish Text: För varje varv i loopen appendas ett object till salar.json. Så när loopen är färdig så var det bara att lägga till de sista paranteserna och städa upp filen så att det skulle validera som JSON. Jag tyckte detta krav var ganska krångligt och min läsning är absolut inte den snabbaste men den gjorde vad den skulle och det fick vara bra nog.

Translation: For every iteration in the loop an object is appended to slar.json. So when the loop is done it was just adding the last parenthesis and clean the file so it is valid JSON. I thought this requirement was kind of complicated and my solution is absolutely not the fastest but it does what it is supposed to do and that should suffice.

Original Swedish Text: Jag har försökt att använda mycket inbyggda metoder som 'map', 'reduce', 'filter', etc. för att få ut rätt information från de arrays jag använder.

Translation: I have tried to use a lot of built-in functions like 'map', 'reduce', 'filter', etc. to get the correct information from the arrays I use.

Original Swedish Text: Valde att inte dela upp min klient som det är gjort i Gomoku. Jag vet att anledningen var för att kunna hålla isär generell och domänspecifik kod, men eftersom jag inte tänkt bygga vidare på den här klienten så lägger jag allt i samma.

Translation: I chose to not split my client as it is done in the Gomoku assignment [4]. I know that the reason is to separate general and domain-specific code, but I am not going to extend on this client so I have decided to put all the code in the same file.

Original Swedish Text: När jag bestämde stil för sidan kollade jag runt lite på andra webbplatser som har en koppling till begravningar.

Translation: When i decided on the style for the page i looked at other websites with a connection to funerals.

Original Swedish Text: Med tidigare uppgifters klienter som grund gjorde jag en klient som kan testa servern.

Translation: With earlier assignments' clients as a base I did a client that tests the server.

---

[4]A programming assignment earlier in the course