World Transactions on Engineering and Technology Education
Vol.17, No.3, 2019

Backtracking assessment of IT and engineering courses

Abdullah M. Hussein† & Mohammed Hussain‡

University of Sharjah, Sharjah, United Arab Emirates†
Zayed University, Dubai, United Arab Emirates‡

ABSTRACT: Measuring student learning outcomes in IT and engineering courses is required to improve course quality. Course outcomes are measured based on the results of student engagement in various course activities, assessments and questionnaires. In this article, the authors present a case study on measuring the outcomes of a programming course, based on student achievements in 13 IT courses, for which the programming course is a prerequisite, i.e. it provides fundamental required skills. First, the learning outcomes of each course were mapped to the learning outcomes of the programming course. Second, the student achievements of the learning outcomes of each course based on student grades in course assessments were calculated. Finally, the calculations obtained from each course were used to calculate the student achievements of the learning outcomes of the programming course, based on the mapping from the first step. The results provided insights regarding student achievement of the programming course outcomes. This research was conducted on data collected from IT courses, but the results are applicable to other disciplines.

INTRODUCTION

Assessment of IT and engineering courses plays an important role in the quality control and improvement of the methods of teaching courses. A course can be assessed by defining a set of learning outcomes for that course and assessing the students' achievement of these outcomes each semester [1][2]. Course outcomes can be evaluated with both quantitative tools, such as the students' marks, students' expectation of course achievement, students' rating of the course and faculty at the end of the semester, and qualitative tools, such as faculty observations of students' performance, surveys and questionnaires [3-5].

A growing number of institutions require faculty members to submit course reports at the end of each semester, which contain an analysis of student achievement for each course outcome for every course. The analysis indicates the level of the students' achievement against the course outcomes, which should lead to a plan of action for improving each course, if some of the outcomes were not achieved. Each semester course report is updated to indicate the changes and the improvements that have taken place over a number of semesters.

However, in some scenarios, relying on student engagement and achievement in course assessment as the main method of measuring course outcomes may not be accurate. Consider the following:

Scenario 1. Course instructor may not correctly align assessment to outcomes. For example, an instructor may use the midterm examination grade as a tool to measure a course learning outcome. However, the midterm questions may be designed to target many outcomes, with only a few targeting the outcome at hand. If student grades in the midterm satisfy the target for that outcome, it is not accurate to conclude that the outcome was achieved.

Scenario 2. Course instructor may use an easy assessment to measure a course outcome. Most students will achieve high grades in that assessment and the grades will surpass the target for that outcome.

Scenario 3. Course instructor may use an unsuitable assessment to measure a course learning outcome. For example, an instructor may use a set of multiple-choice questions as a tool to measure an outcome that may be better measured via an essay question.

In all three scenarios, the instructor may reach a conclusion that the course outcomes were achieved. Illustrated in this article is that the calculation of student achievement of a given course outcome can be enhanced by measuring the student achievement in courses that depend on that given course. In other words, if course A is a prerequisite for courses B, C and D, then the data from the latter courses can be applied to measuring student achievement of the outcomes of course A. Demonstrated in the article is the proposed approach by calculating the outcomes of a programming course, based on the data collected from 13 dependent IT and computer science courses over eight semesters. The proposed method is meant to complement, and not to replace, the evaluation of course outcomes based on course assessment.

RELATED WORK ON COURSE EVALUATION

The process of evaluating course outcomes is meant to ensure that the course learning outcomes are monitored and measured. Results obtained should guide continuous improvement of the courses, the programme and the educational experience of the students [6]. Measures of student engagement and achievement of course outcomes can be categorised into qualitative, i.e. open-ended questions, and quantitative measures, i.e. student grades in assessment. Measures also can be categorised into self-reporting, such as questionnaires or observational, such as rubrics [7].

Faculty observations of students' performance can be recorded through rubrics [8-10]. Hence, a comprehensive course assessment and collection of results are applied to evaluate the achievement of course and programme outcomes. Grover et al examined student achievement of outcomes of a basic programming course [8]. Diller et al reported on the application of rubrics in measuring the information literacy component of the general education programme at the authors' institution [9]. Shuman et al discussed the use of rubrics to measure professional skills in the Accreditation Board for Engineering and Technology (ABET) proposed outcomes for engineering programmes [10].

Missett et al compiled a dataset of student participation in course discussion boards, surveys and email correspondence [11]. The dataset was then analysed to measure student engagement and to demonstrate evidence of learning. Parker et al incorporated the concept of the learner's readiness to change while measuring the outcomes of medical courses [12]. The work addresses the situation where students may achieve the outcomes, but that achievement may not result in a change of behaviour.

Educational data mining and learning analytics apply the data generated from the learning management systems and other digital tools used by the students to build models of student behavior [13][14]. Such models help predict student performance in courses, anticipate student preferences and increase attrition rates, identify students at risk and improve learning outcomes [15]. The application of big data in higher education extends and enhances the benefits gained from the above models and support institutions to better understand their data [16][17].

A common issue with the present measures for evaluating student achievement of a given course outcomes is their focus on assessing the student performance and engagement in that particular course. This work takes into consideration that student performance and engagement in later courses serve as an additional source of data for course outcome evaluation.

CASE STUDY: COURSE RESULTS AND ANALYSIS

The first programming course in the Computer Science undergraduate programme was selected for the case study. The course is similar to typical first programming courses found in any computer science or IT programme. This course, referred to as Programming 1, is a prerequisite to many science and engineering courses, also contributing to many advanced computer science and computer engineering courses.

The assumption here is that if the learning outcomes of Programming 1 are achieved successfully, then this will positively contribute to the achievement of the learning outcomes of all future courses for which the course Programming 1 is a prerequisite. This assumption was made because programming is a course of accumulating knowledge and skills, and if the basic concepts are not understood correctly then the material of more advance courses will not be understood correctly. Below is a list of university courses for which Programming 1 is a prerequisite or the skill of programming is needed:

1. Object Oriented Programming
2. Data Structures
3. Java Programming
4. Programming Languages and Paradigms
5. Operating Systems
6. Design and Analysis of Algorithms
7. Web Programming
8. Vision and Image Processing
9. Computer Graphics
10. Development of Web Applications
11. Mobile Applications and Design
12. Game Design and Development
13. Senior Project

The achievement of learning outcomes of the above courses is an indication that the learning outcomes of Programming 1 have been achieved. The following sections will refer to the course Programming 1 as the independent course, while the 13 courses (above) are the dependent courses. The learning outcomes of Programming 1 are listed in Table 1.

Table 1: Programming 1 learning outcomes.

Code	Outcome
LO1	Identify different phases of problem solving and algorithm design.
LO2	Use the concepts of variables, data types, input, output, expressions and assignment.

LO3	Develop, test and debug computer programs.
LO4	Apply selection and repetition statements.
LO5	Apply modular programming.
LO6	Use the concept of pointers and arrays.

Two traditional tools usually are applied to evaluate the leaning outcomes achievement of a course; namely, student marks and the faculty observation of students' performance. The first involves applying the marks of the midterm examination, the course work (quizzes and assignments) and the final examination. The mark distribution adopted to evaluate learning outcomes for the case study in one semester is shown in Table 2.

Table 2: Marks distribution for the learning outcomes.

Tool	Mark	LO1	LO2	LO3	LO4	LO5	LO6
Midterm examination	20	6	3	5	6		
Course work	35	6	6	6	5	6	6
Final examination	45	2	2	8	10	11	12
Percentage	100	14	11	19	21	17	18

The last row in Table 2 is a weighted measure for evaluating course outcomes based on their importance. If outcome A, say, is more important than outcome B, then outcome A should carry a higher weight. For example, LO2 in Table 2 is less important than LO3; hence, LO3 was weighted at 19%, while LO2 was weighted at 11%. Table 3 shows the course average achievement for each course outcome, by calculating students' marks.

Table 3: Course learning outcomes: averages.

Outcomes	LO1	LO2	LO3	LO4	LO5	LO6
Average	77.1	77.9	62.1	63.2	63.6	62.0

The average row in Table 3 is a measure whether a course outcome is achieved or not. Having different values of averages for the course outcomes indicates that the outcomes were not equally achieved. Low averages necessitate action to improve for the next course offering.

The other assessment tool of course outcome achievement is the faculty observations of students' performance. This can be achieved using rubrics for the different course components, such as laboratories, presentations and assignments. A rubric is a table in which the first column lists the course outcomes and the remaining columns detail the level of understanding of the course materials (outstanding, adequate, developing and ineffective), which reflect on the level of course outcome achievement.

The course instructor can design a rubric to assess the learning outcomes and record the level of students' understanding of the course materials during a laboratory session or in a discussion or when marking an assignment. These observations are recorded in one rubric for the whole session and not for each student. As a student progresses through the semester the course instructor can form an opinion on the level of course outcome achievement.

If there are 12 laboratories to be covered during the semester in a programming course then suppose that laboratories 1, 2, 3 and 4 cover outcome LO1 and LO2, while laboratories 5, 6, 7 and 8 cover outcomes LO1, LO2, LO3 and LO4. Also, laboratories 9, 10, 11 and 12 cover outcomes LO1, LO2, LO3, LO4, LO5 and LO6, then the instructor can monitor how the students' understanding of the course materials is improving as the semester progresses.

PROPOSED METHOD OF ASSESSMENT

The proposed method of assessment is based on the assumption that to understand a dependent course it is necessary to understand the independent course. If a student can progress at ease through all dependent courses, then that is an indication the student understood the independent course.

Assuming that the learning outcomes of a dependent course are labelled A, B, C, D, E, F, then the learning outcomes of the independent course can be related to the learning outcomes of a dependent course, e.g. to relate a particular learning outcome of the independent course (say LO1) with the learning outcomes A, B, C, D, E and F of a dependent course, the value 1 is assigned, if LO1 is related and the value 0, if it is not related.

The sequence of values 100110 indicates that the learning outcome LO1 of the independent course is related to the learning outcome A, D and E of the dependent course. The binary bits in Table 4 illustrate the relation of learning outcome LO1 of the independent course to the learning outcomes of the dependent course. Table 5 shows the relationship of all the learning outcomes of the independent course Programming 1 with the learning outcomes of the 13 dependent courses.

Table 4: Relating dependent course outcomes to an outcome of the independent course.

LO	A	B	C	D	E	F
LO1	1	0	0	1	1	0

Table 5: Relating Programming 1 learning outcomes to the learning outcomes of the dependent courses.

	LO1	LO2	LO3	LO4	LO5	LO6
1	100000	010000	111110	001000	000010	000100
2	010001	111110	111010	110010	010000	101110
3	100000	111100	111100	010000	000000	011100
4	100000	010000	010011	001010	001100	000001
5	100010	010000	101001	110000	100100	000100
6	100000	001010	000000	101011	001001	000011
7	100000	010000	111101	001000	011100	000010
8	110100	000100	011100	011000	111000	001000
9	111000	001011	011110	000100	110111	001100
10	100000	010000	011000	001110	011111	000000
11	100000	010000	011000	001000	011110	001000
12	100000	010000	101111	001010	011111	000010
13	110000	000000	011000	000000	010000	010000

The learning outcomes averages of all the dependent courses during a semester are recorded in a table similar to Table 6. The contribution of all dependent courses to the achievement of the learning outcomes of the independent course is calculated with an average of the dot product of a row of Table 6 by a set of 0s and 1s value from Table 5. For example, the contribution of the dependent course number 1 to the learning outcome LO, is the dot product of the first row of Table 6 with the row vector (1, 1, 1, 1, 1, 0), which is the set of 1 and 0 values in the cell of row number 1 and column LO3. The result of the dot product is then divided by how many 1s are in the row vector (1, 1, 1, 1, 0). Table 7 shows the result of the calculations for the contribution of all courses to all the learning outcomes LO1, LO2, LO3, LO4, LO5 and LO6 of the independent course.

Table 6: Average achievement of the learning outcomes of the dependent courses.

Course	A	B	C	D	E	F
1	73.6	70.8	79.3	82.5	69.8	
2	67.9	66.1	81.8	67.4	63.5	80.5
3	78.5	65.9	85.5	88.3		
4	79.5	79.9	80.7	79.9	79.9	95.6
5						
6						
7	73.9	68.9	78.4	65.6	80.8	
8						
9						
10	88.8	89.1	81.7	79.8	69.9	86.6
11	76.4	81.1	88.6	76.9	85.3	
12	67.9	64.1	88.4	80.9	88.8	86.2
13	88.7	95.0	92.6	93.2	88.0	

Table 7: Average achievement of the learning outcomes LO1 to LO6 of the independent course.

Course	LO1	LO2	LO3	LO4	LO5	LO6
1	73.6	70.8	75.2	79.3	69.8	82.5
2	73.3	69.3	69.8	65.8	66.1	70.2
3	78.5	79.6	65.9	65.9		79.9
4	79.5	79.9	80.3	80.3	80.3	95.6
5 not run						
6 not run						
Average	76.2	74.9	77.4	72.8	72.1	82.0
7	73.9	68.9	57.4	78.4	71.0	80.8
8 not run						
9 not run						
10	88.8	89.1	85.4	77.1	81.4	

11	76.4	81.1	84.9	88.6	83.0	88.6
12	67.9	64.1	80.0	88.6	81.7	88.8
Average	76.8	75.8	76.9	83.2	79.3	86.1
13	91.9	0.0	93.8	0.0	95.0	95.0

Table 7 shows the contribution of the 13 dependent courses to the learning outcomes of the independent course during one semester. Courses 5, 6, 8 and 9 were not offered during that semester. The 13 courses were grouped into three; namely, core courses (1, 2, 3, 4, 5 and 6), elective courses (7, 8, 9, 10, 11 and 12) and course 13, a capstone course. Table 8 shows the weights assigned to the contribution of each group of the dependent courses to the learning outcomes of the independent course.

Table 8: Weights for the learning outcomes.

Group	LO1	LO2	LO3	LO4	LO5	LO6
Core	60%	60%	60%	60%	60%	60%
Elective	20%	20%	20%	20%	20%	20%
Capstone	20%	20%	20%	20%	20%	20%

The core courses usually carry more weight than the elective courses, because the number of credits assigned to the core courses is more than the number of credits assigned to the elective courses. Although the capstone course is only one course, it carries similar weight to the elective courses because of its importance. The above weights are not fixed and can be changed. Different learning outcomes can be assigned different weights, based on their importance. Table 9 shows the average contribution of the 13 courses by applying the weights of Table 8 for one semester.

Table 9: Average contribution to the assessment of the learning outcomes.

Outcomes	LO1	LO2	LO3	LO4	LO5	LO6
Average	79.5	75.1	80.6	75.4	78.1	85.4

RESULTS OF THE ASSESSMENT

Table 9 shows the result of one semester. To measure the contribution of all courses, a time period of eight semesters is needed. The averages in Table 10 were calculated over the eight semesters. Calculations must be done for eight semesters, because it is the time period undergraduate students need to finish the required number of credits. Figure 1 shows a histogram of the above averages over the eight-semester time period.

Table 10: Average achievement of the learning outcomes over eight semesters.

Semester	LO1	LO2	LO3	LO4	LO5	LO6
Fall* 14	79.5	75.1	80.6	75.4	78.1	85.4
Spring 14	77.5	79.2	80.7	74.0	81.6	82.3
Fall* 15	75.0	72.0	75.0	78.6	80.2	75.0
Spring 15	74.0	64.2	75.4	73.9	72.7	77.2
Fall* 16	76.0	78.9	73.0	76.4	76.8	77.8
Spring 16	83.5	76.2	81.0	75.2	85.7	86.2
Fall* 17	77.6	75.9	76.5	75.9	78.1	77.2
Spring 17	81.3	75.3	74.5	73.6	80.3	80.2

*Fall = Autumn

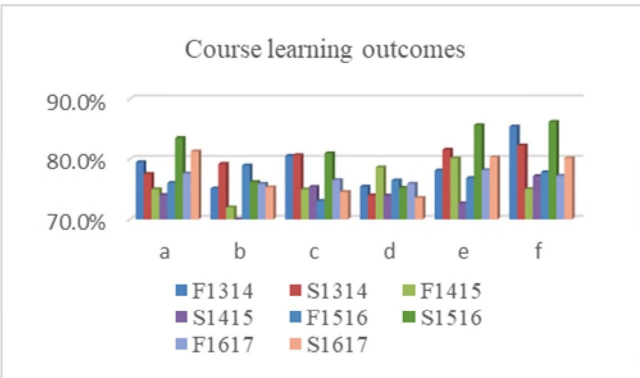

Figure 1: Course learning outcome achievement based on the achievement of LOs of a number of dependent courses.

CONCLUSIONS

Presented in this article is an approach for the assessment and evaluation of course outcomes. Rather than measuring the outcomes of a particular course based on student performance in that course, this approach extends the evaluation to include student performance in all courses, for which the evaluated course is a prerequisite.

The approach was applied to measuring the outcomes of Programming 1, a first-year programming course. The measurement was based on student performance in 13 courses, for which Programming 1 is a prerequisite. The presented approach should lead to course improvements, as it provides a more accurate assessment of course outcomes.

REFERENCES

1. Kuh, G.D., Jankowski, N., Ikenberry, S.O. and Kinzie, J.L., *Knowing what Students Know and can Do: The Current State of Student Learning Outcomes Assessment in US Colleges and Universities*. National Institute for Learning Outcomes Assessment, University of Illinois and Indiana University (2014).
2. Jankowski, N.A., Timmer, J.D., Kinzie, J. and Kuh, G.D., *Assessment that Matters: Trending Toward Practices that Document Authentic Student Learning*. National Institute for Learning Outcomes Assessment, University of Illinois and Indiana University (2018).
3. Biggs, J.B. and Collis, K.F., *Evaluating the Quality of Learning: the SOLO Taxonomy (Structure of the Observed Learning Outcome)*. Academic Press (2014).
4. Diamond, R.M., *Designing and Assessing Courses and Curricula: a Practical Guide*. John Wiley & Sons (2011).
5. Suskie, L., *Assessing Student Learning: a Common Sense Guide*. (3rd Edn), John Wiley & Sons (2018).
6. Susilowati, D., Degeng, I., Setyosari, P. and Ulfa, S., Effect of collaborative problem solving assisted by advance organisers and cognitive style on learning outcomes in computer programming. *World Trans. on Engng. and Technol. Educ.*, 17, **1**, 35-41 (2019).
7. Henrie, C.R., Halverson, L.R. and Graham, C.R., Measuring student engagement in technology-mediated learning: a review. *Computers & Educ.*, 90, 36-53 (2015).
8. Grover, S. and Basu, S., Measuring student learning in introductory block-based programming: Examining misconceptions of loops, variables, and boolean logic. *Proc. 2017 ACM SIGCSE Technical Symp. on Computer Science Educ.*, 267-272 (2017).
9. Diller, K.R. and Phelps, S.F., Learning outcomes, portfolios, and rubrics, oh my! Authentic assessment of an information literacy program. *Libraries and the Academy*, 8, **1**, 75-89 (2008).
10. Shuman, L.J., Besterfield-Sacre, M. and McGourty, J., The ABET *professional skills* - can they be taught? Can they be assessed? *J. of Engng. Educ.*, 94, **1**, 41-55 (2005).
11. Missett, T.C., Reed, C.B., Scot, T.P., Callahan, C.M. and Slade, M., Describing learning in an advanced online case-based course in environmental science. *J. of Advanced Academics*, 22, **1**, 10-50 (2010).
12. Parker, K. and Parikh, S.V., Applying Prochaska's model of change to needs assessment, programme planning and outcome measurement. *J. of Evaluation in Clinical Practice*, 7, **4**, 365-371 (2001).
13. Romero, C. and Ventura, S., Educational data mining: a review of the state of the art. *IEEE Trans. on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40, **6**, 601-618 (2010).
14. Baker, R.S. and Inventado, P.S., Educational data mining and learning analytics. *Learning Analytics*. Springer New York, 61-75 (2014).
15. Greller, W. and Drachsler, H., Translating learning into numbers: a generic framework for learning analytics. *J. of Educational Technol. & Society*, 15, **3**, 42 (2012).
16. Daniel, B., Big Data and analytics in higher education: opportunities and challenges. *British J. of Educational Technol.*, 46, **5**, 904-920 (2015).
17. Huda, M., Haron, Z., Ripin, M.N., Hehsan, A. and Yaacob, A.B.C., Exploring innovative learning environment (ILE): big data era. *Inter. J. of Applied Engng. Research*, 12, **17**, 6678-6685 (2017).