https://doi.org/10.48009/1_iis_2008_60-67

The Role of Learning Technique on Student Performance in CS1 courses

Stanley T. Schuyler, Edinboro University, sschuyler@edinboro.edu
Robert Joseph Skovira, Ph. D., Robert Morris University, skovira@rmu.edu
A. J. Grant, Ph. D., Robert Morris University, granta@rmu.edu

ABSTRACT

Umberto Eco theorized a semiotic process through which people perceive, interpret, internalize (learn), and produce (articulate) information. The process involves a set of symbolic representations and transformations that model how humans acquire and process information. Hughes and Peirsi investigated *Learning Approaches* that students use to learn object oriented (OO) programming. Three learning approaches were described and studied to determine how effective each were for acquiring OO programming competencies. The semiotic model stimulated the idea for a conceptual framework that would incorporate these *Learning Approaches*. In order to integrate them into the framework, nine elemental *Learning Techniques* (denoted LTs) were identified. Using the framework, a strategy was developed for determining which *Learning Techniques* students use and which are effective for learning computer programming. This paper describes the research on these learning techniques and how they can be used to predict the performance of students in their first course in computer programming.

INTRODUCTION AND RELATED STUDIES

This study was done as a part of a broader study [12] researching students' *Problem Formulation Ability* (PFA) and the role it plays in learning computer programming [5] [6][8] [9][11][14]. Participants were students in a first course in computer programming (denoted *CS1*). Although the role of PFA is described in conjunction with LT, the interested reader can review details about PFA in [12]. The objective is to develop an assessment for identifying that a student is, or is not, ready for CS1. The term *CS0* is used to refer to a preparatory programming course.

The investigation of *Learning Technique* (LT) was motivated by three separate studies: the first by Umberto Eco and his description of semiotics [3]; the second, by the application of semiotics to systems analysis and engineering [7]; and third, a study on learning approaches by Hughes and Peirsi [4]. Figure 1 (bottom of report) presents a highly compressed model of the theory of semiotics. The diagram was synthesized from reading Eco's description of the theories of signification and of codes [3]. The transformations of signifiers to (or from) denotative interpretants, to (or from) connotative sememes, and into (or out of) a semantic model, can be used as one dimension of a conceptual framework to relate learning approaches and techniques.

The concept of Learning Approach [4] had three main components: *surface* (i.e. memorizing), *strategic* (i.e. seeking a specified outcome), and *deep* (i.e. seeking underlying meaning). Their investigation was conducted with participants learning OO programming. A summary of their findings is presented in Table 1 below.

Table 1: Distribution of students' Performance based on Learning Approach [4] *

Learning Approach	Deep (seeks meaning)	Surface (Memory)	Strategic (seeks grade)	High Strategic Low Surface	High Surface Low Strategic
N	0	2	20	34	
Rating	NA	Poor	Good	Good	Poor

*Adapted from (Hughes & Peirsi , 2006, pp. 276-277)
Note: there were exceptions reported in each category.

A *Surface* learning approach was associated with a poor performance rating [4]. Further, an approach dominated by a *surface* approach but in weak combination with *strategic* approach (termed *High Surface, Low Strategic*), was also associated with poor performance. *Strategic* approaches (outcome or goal focused), and approaches that were predominantly *strategic* but mixed with a low *surface* approach, were associated with good performance. No participants were identified using *deep* approaches, but they attributed not identifying these cases in part due to a limitation in their methodology.

Hughes and Peirsi did not provide visibility into the tactics that participants used to enable these learning approaches. Learning depends on the *Learning Approach* [4], but learning approaches are dependent on the tactics used to internalize information [2]. For example, does a student use special cases and inductive reasoning to formulate the general case?

Alternatively, does a student take a general rule and use deduction to determine a specific case? Do they even understand these concepts? Another aspect is related to tactics used for representing or conveying information: do students depend on visualization, rewriting, or discussions for retaining information?

In this report we describe nine learning tactics, the method for determining predominant usage, what patterns of use emerged, the conversion of patterns into numeric scores, and the use of these scores to predict CS1 course grades. The numeric measure is denoted *LTS* (for Learning Technique Score). LTS was demonstrated to be effective for predicting the course grades of first time programming students (also PFA) [11] [12]. Neither factor predicted course grades for experienced students.

The Learning Technique Model

Three *Learning Approaches*, described by Hughes and Peirsi [4], and introduced above were: *Surface*, *Strategic*, and *Deep*. Each of these can be related to the semiotic model in Figure 1. To show their relationships, a semiotic-learning framework was developed. The framework is depicted in Table 2 (at the bottom of document). The development of Table 2 is explained by stepping through each component. First, the semiotic transformations [3] are placed in the columns of Table 2 and form one of the dimensions.

Learning Technique is viewed as a selection of *learning tactics* and *expressive mode tactics* that are employed to implement learning approaches [2]. Six learning tactics were identified: *Memorization*, *Deduction*, *Induction*, *Experimentation* (trial and error), *Relation* (discovery of underlying meaning), and *Abduction* (imagination) [2]. Three expressive mode tactics were identified: *Discussion* (oral discourse), *Reflection* (transformation to written discourse), and *Visualization* (transformations to spatial depictions such as drawings or tables). These two categories are considered as a single dimension and are referred to as *Learning Techniques* (LTs).
The rows of Table 2 contain these nine LTs. The *Learning Approaches* [4] are positioned in the cells of Table2.

A *Surface* approach is associated with the memorization of facts. In this sense, *facts* can be associated with denotative meanings produced by signification: that is, the transformation of //signs// into *interpretants*. A *Strategic* approach involves one's personal motivations for achieving specific goals or competencies. *Motivation* can be thought of as an internalized property of the *Semantic Model* in Figure 1. A *Deep* approach can be associated with seeking underlying meaning produced by the more complex process of semiosis: that is, the transformation of *interpretants* into *sememes* or connotative meanings. The linkage of motivation to a *strategic* approach could be achieved in at least two ways: an association paired with a *surface* approach (theory of signs), or paired with a *deep* approach (theory of codes).

The initial model revealed a gap between *Strategic* and *Deep* approaches caused by the identification of *Experimentation* as a learning technique [10] [11]. This gap was filled by proposing a fourth learning approach labeled *Experience* and positioning it between *Strategic* and *Deep* (note: this approach was not described in [4]).

Purpose and Objectives

The research questions driving this study on factors affecting student performance in CS1 courses were [11]: (1) which technique, or pattern of techniques, is used by students to learn computer programming? (2) Is there evidence that supports one technique, or pattern of techniques, as being more effective for learning computer programming than others?

There were two primary purposes: to gather evidence that answers the research questions described above; and to develop a scaled variable to represent LTS and use it to predict student performance (defined to be a student's final CS1 course grade).

To these ends the following objectives were established: 1) develop an operational definition of LTS (*Learning Technique Score*); 2) qualitatively characterize the composition of LTS in terms of tactical components and pattern; 3) support or refute the results on *Learning Approach* in [4]; and 4), determine the level of *effectiveness* that LTS has in predicting final course grades of students with no prior programming experience. The determination of *effectiveness* is based on the variable LTS being entered into a stepwise linear regression analysis at the p.05 level and accounting for at least 10% of the variance in course grades.

METHODOLOGY

The study of LT was part of a broader study focused on PFA [11]. In that that study five CS1 course sections were solicited to participate at two western Pennsylvania universities. Participants were volunteers in these course sections. A questionnaire was administered in the second meeting of each

course section to gather information regarding gender, GPA, major, prior programming experience, use of learning tactics, and problem formulation competency. Final exam and course grades were returned to the principle researcher at the end of the semester.

Data Collection

There were a total of 107 students enrolled in these sections. Of these, 90 volunteered to participate. Participants were asked to rank order the nine LT tactics as presented in Table 3 (actual wording and order).

Table 3: Learning Technique List for Rank Ordering Task

Skill or Technique used to retain and/or apply information	Rank Order
Deductive Logic (use general rules to regenerate information)	
Discovery of underlying meaning and relationships	
Imagination (create or design unique models)	
Inductive Logic (use collections of specific examples to regenerate general rules)	
Memorization	
Oral communication (discussions, arguments, presentations)	
Trial and Error (guess and validate)	
Visualization (use drawings, graphs, and or tables)	
Written communication (notes, reports or stories)	

Participants were to base their rankings on their perceived personal use of each tactic to learn and retain information. After participants rank ordered the items (ranks 1 through 9), the six learning tactics were reassigned values from 1 to 6, and the three expressive mode tactics values 1 through 3 (preserving the order in each participant's original ranking). It was recognized that the list was terse and subject to wide variation in interpretation. Therefore, only the first two learning tactics and first expressive mode rankings were used.

To determine a population ranking, individual tactic rankings were summed across individuals. Table 4 below presents the rankings for the top two learning tactics and top expressive mode tactic.

Table 4: Ranking of Tactics (N=90)

Learning				
Tactics	Rank	Σ1	Σ2	Σ
Induction	1	22	19	41
Deduction	2	21	18	39
Experiment	3	18	16	34
Memorize	4	18	9	27
Relation	5	8	12	20
Abduction	6	3	16	19
Expressive Mode				
Tactics	Rank	Σ1	Σ2	Σ
Visualization	1	43	24	67
Discussion	2	26	32	58
Reflection	3	21	34	55

Definition of LTS

An LTS (*Learning Technique Score*) had to represent two properties of each participant's rank ordering: first, the property of representing tactic ordering for learning and selection of expression; and second, the difference between the population ranking of the top two tactics and the individual's ranking. *Induction* and *deduction* were ranked 1st and 2nd in the population. If an individual ranked *induction* first and *deduction* second, and another individual the reverse, then the LTS needed to reflect the difference. This meant that simply summing the rank positions, e.g. 1+2 = 2+1, would not be effective. Further, if an individual ranked *induction* and *deduction* 5th and 6th, then the LTS needed to be a value at the opposite end of the scale from those ranking *induction* and *deduction* 1st and 2nd.

The approach adopted to calculate LTS was based on each participant's rank order of *induction*, *deduction* and *visualization* (because these were the top three ranked tactics in the population as shown in Table 4). The equation used was patterned on an approach described in Dagsvik and Liu [1], which preserves category order and normalizes the value. The equation used was:

$$LTS = \frac{14 - (Ind. + Ded.^{0.75} + Vis.^{1.15})}{8.5} - 0.5$$

The resulting scale ranged from -0.5 to +0.5. The value of the exponents was determined by trial and error. The criterion used for the exponents was to maximize the apparent correlation between LTS and PFA (see [12]). The reason for using this criterion was that LTS and PFA were expected to be independent. After setting the exponents to maximize the apparent correlation, a linear regression demonstrated there was no relationship between LTS and PFA (adjusted R^2=.000). The researchers

anticipated that this task could yield a random set. Therefore, the determination of LTS effectiveness was based on the percent of variance accounted for in a linear regression model.

Data Analysis

LTS values were analyzed two ways: first, qualitatively to understand the characteristics of distribution in terms of tactical patterns and scores; and second, as part of a stepwise linear regression analysis used to evaluate the effectiveness of LTS (and PFA) for predicting students' course grades.

RESULTS

Demographic Results

Table 5 presents the final disposition of participants in this study (as reported in [12]). Of 90 participants, 24 withdrew and 66 completed. Of the 66, 23 had previous programming experience, 43 were first time programmers, but 5 of the 43 were determined to be outliers [11] [12].

Table 5: Participant Experience and Completion

	Comp-leted	With-drew	Sub-total	Poor, Fail	With-drew, Poor
No Exper-ience	43	20	63	9	29
	48%	22%	70%	10%	32%
Exper-ienced	23	4	27	3	7
	26%	4%	30%	3%	8%
Totals	66	24	90	12	36
	73%	27%		13%	40%

Detailed demographic results are reported in [12]. For the purposes of this study, the Analysis of Variance tests revealed no significant differences across the sample population except for programming experience between course sections. As this research was focused on students with no prior experience in programming, the only impact was that very few students were included from two sections in the no-experience group (the 38 were primarily from 3 sections at one university).

LTS Distribution Characteristics

Figure 2 (bottom of report) presents the distribution of *Learning Techniques* jointly with LTS values as a 3-dimensional histogram. LTS values are represented every 0.1 units (actual scores were rounded to the nearest tenth unit). Each score interval represents a similar pattern of 1st and 2nd rank choices. *Learning tactics* are represented along the z-axis (expressive mode tactics are not shown as only the ranking of *visualization* was included in the calculation of LTS). The distribution shows the population ranking along the far left side (ranks summed across the LTS dimension). The population use of a particular set of tactics is summed across the LT axis and is displayed along the back surface of Figure 2. There are two clear modes (mounds): one between +0.1 and +0.5, and one between -0.3 and -0.1.

To understand the actual patterns used, a "Learning Landscape" can be envisioned along the lines of an *Information Landscape* proposed by Skovira [13]. In a sense, this is a "zooming in" on the 3-D histogram of Figure 2 to expose the qualitative aspects of the LT dimension. A portion of this learning landscape is presented in tabular form in Table 6 (bottom of report). The complete table was too large to fit in the space permitted for this publication, but the main tactic patterns are presented for each LTS band.

The tactic choices from -0.5 to +0.2 are diverse and the frequency of *induction* and *deduction* being ranked first or second is lower than LTS scores above +0.2. Tactics such as *memorize*, *experiment* and *abduction* occur frequently. Expressive mode tactics are also diverse. What is informative are the mean CSGs (Course Grades on a 4.0 scale) in each score range which go from failing, to 1.5, 2.5, 2.7, 2.8 to 2.4 with an average of 2.5. Further, 90% of the poor performers (12% of the total population), and 67% of the withdrawers (18% of the population) are in this range.

What is notable about the range from +0.3 to +0.5 is that *induction* and/or *deduction* are either the first or second ranked tactic. This is true even when some other tactic is first (such as *memorize*, *experiment* or *relation*). The mean CSG is 3.1, only 10% of the poor performers (1% of the population) are in this range. Of the participants that withdrew, only 33% (corresponding to 9% of the population) were in this LTS range. Expressive tactics tend to favor visualization for positive LTS, while other expressive tactics seem more randomly selected toward negative LTS.

Using LTS to predict course grades

LTS (and PFA variables) were submitted as the independent variables to a stepwise linear regression procedure. Table 7 presents the results for 38 of the 43 participants that reported having no previous programming experience (5 participants determined to be anomalies were excluded). Both LTS and PFA entered the equation at the p.05 level. LTS accounted for 16% of the variance, and PFA 32%of the

variance. A joint 3-D plot is presented in Figure 3 (also at the bottom) showing LTS and PFA effects on course grades. Actual grades are shown as circles and predicted grades are shown as letters adjacent to each circle. LTS is the perspective from the right side and PFA is the perspective from the left side.

Table 7: Stepwise Linear Regression: Course Grade

Descriptive Statistics				
Variables		Mean	Std. Deviation	N (no outliers)
Dep.	CSG	2.874	1.1037	38
Indep.	PFA	3.148	.8904	38
Indep.	LTS	.1608	.24520	38
Dependent Variable: Course Grade (CSG) Model Summary				
Model	Adjusted R Square	df1	df2	Sig. F Change
1-PFA	.324	1	36	.000
2-LTS	.485	1	35	.001
Predictors: (Constant), PFA, LTS				

A second analysis was run with all 43 participants reported as non-experienced (included the 5 anomalies). The results of this regression showed LTS accounting for 21% of the variance and PFA 11% for a total of 32%.

An alternative method of determining how well LTS and PFA predicted course grades (CSGs) was to see how well the final exam grade, treated as an independent variable, predicted students' course grades. The final exam grade entered the stepwise regression at the p.05 level and accounted for 51% of the variance in CSGs (N=38). This result is only 3% more than LTS and PFA jointly explain in participants with no prior programming experience.

CONCLUSIONS

This study demonstrated a rank ordering and scoring procedure (LTS) to assess the primary learning technique pattern used by students entering their first course in computer programming. The value of LTS represents the degree to which a participant's rankings aligns or differs from the dominant ranking in the sample population. LTS effectively predicted between 16% and 21% of the variance in students' final course grades (for first time programmers in their first programming course).

The LT distribution (Figure 2 and Table 6) and regression results provide evidence supporting the conclusions of Hughes and Peirsi [4] regarding *Learning Approach*. Participants with LTS scores at or above +0.3 use *induction* and/or *deduction* (Table 6) and these align with a *strategic* approach in the semiotic framework (Table 2). Other successful combinations occur in this range such as *induction* or *deduction* combined with memorization (as a second choice) that aligns with *high strategic-low surface*. These patterns are associated with an average CSG of 3.1. For LTS at or below +0.2 *memorization* aligns with a surface approach, and when combined with induction or deduction aligns with *high surface-low strategic*, both associated with poorer performance (average CSG of 2.5 or less).

The LT distribution reveals a richer view of other combinations that were missing in [4]. Approaches that use *abduction* or *relation* in combination with themselves and/or *induction*/*deduction* are revealed. These would correspond with *deep* or combined *deep-strategic* approaches not visible in [4].

The methodology, which used an overly simple rank ordering task, opened a window for viewing a learning landscape that helps assess the role of specific learning tactics in learning computer programming. These results provide supportive evidence that students capable of formulating questions (inherent in induction and deductive tactics) perform better than students who can not or do not use these techniques. LTS, along with PFA, can help determine which students should be placed in a CS0 course, or are ready for a CS1 course.

REFERENCES

1. Dagsvik, John K., and Liu, Gang. (2006). *A Framework for Analyzing Rank Ordered Panel Data with Application to Automobile Demand*. Discussion Papers No. 480, October, 2006. Statistics Norway, Research Department, pp. 1-20.
2. Dewey, John. (1939). *Intelligence in the modern world - John Dewey's philosophy* (Joseph. Ratner, Ed.). Toronto, Canada: Random House of Canada Limited.
3. Eco, Umberto. (1976). *A Theory of Semiotics* (1979 (Midland) ed.). Bloomington, Indiana, U.S.A: Indiana University Press by arrangement with Bompiani, Milan, Italy.
4. Hughes, Janet., & Peirsi, D. Ramaee. (2006). *ASSISTing CS1 students to learn: Learning approaches and object oriented programming*. In Implagliazzo & Goldweber & Salamoni (Eds.), SIGCSE Bulletin, 38(4), pp. 275-279.

5. Ivins, Jim & Ong, Michele Poy-Suan. (2003). *Psychometric assessment of computing undergraduates*. (Department of Computing, pp. 1-15). Perth, Australia: Curtin University of Technology
6. Lister, Raymond, Adams, Elizabeth S., Fitzgerald, Sue, Fone, William, Hamer, John, Lindholm, Morten, McCartney, Robert, Mostrom, Jan Erik, Sanders, Kate, Seppala, Otto, Simon, Beth, and Thomas, Lynda. (2004). *A Multinational Study of Reading and Tracing Skills in Novice Programmers*. SIGCSE Bulletin, 36(4), pp. 119-150.
7. Liu, Kecheng. (2000). *Semiotics in Information Systems Engineering.* Cambridge, United Kingdom: Cambridge University Press.
8. Perrenet, Jacob, and Kaasenbrood, Eric. (2006). *Levels of Abstraction in Student's Understanding of the Concept of Algorithm: the Qualitative Perspective*. Implagliazzo, Goldweber and Paola (Eds.), SIGCSE Bulletin, 38(3), pp. 270-274.
9. Rountree, Nathan, Rountree, Janet and Robins. (2004). *Interacting Factors that Predict Success and Failure in a CS1 Course*. SIGCSE Bulletin, 36(4), pp. 101-104.
10. Schuyler, Stanley T., Skovira, Robert J. (2007). *Is the Problematic in CS1 a Student's Problem Solving Ability?* In Issues in Information Systems, IIS Vol. VIII, No 1-2, pp.112-119.
11. Schuyler, Stanley T. (2008). *Using Problem Formulation Ability to predict Student Performance in a First Course in Computer Programming*. Doctoral Dissertation, Robert Morris University.
12. Schuyler, Stanley T. (2008). *Problem Formulation Ability is a student's problem in CS1 courses*. To be published in IIS associated with IACIS 2008.
13. Skovira, Robert Joseph. (2007). *Ontological grounding of a knowledge mapping methodology: Defining data, information, and knowledge*. Unpublished manuscript, Robert Morris University, Moon, PA.
14. Wiedenbeck, Susan, LaBelle, Deborah and Kain, Vennila, N.R. (2004). *Factors Affecting Course Outcomes In Introductory Programming*. PPIG 2004 – 16th Workshop of the Psychology of Programming Interest Group, Retrieved April 23, 2007 from www.ppig.org.

Table 2: Integrated Framework relating learning techniques and semiotic transformations

Semiotic Dimension / Learning Technique	Sign Perception	Signific-ation	Interpret-tion	Semiosis	Semantic Model
Memorization	Surface [4]				
Deduction (use general rules to regenerate information)			Strategic [4]		
Induction (use collections of specific examples to regenerate general rules)					
Experimentation (guess and validate)			Experience **		
Relation (discovery of underlying meaning and relationships)				Deep [4]	
Abduction (imagination: create or design unique models)					
Discussion (oral communication, debates, arguments, presentations)	Dynamic Language			Discursive	
Reflection (written communication, notes, reports or stories)	Static Language			Reflective	
Visualization (use drawings, graphs, and/ or tables)	Spatial Linguistic Models			Spatial	

** The learning approach labeled "Experience" was not in [4]; it was added to acknowledge a gap between *Strategic* and *Deep* created by the insertion of *Experimentation* as a learning technique.

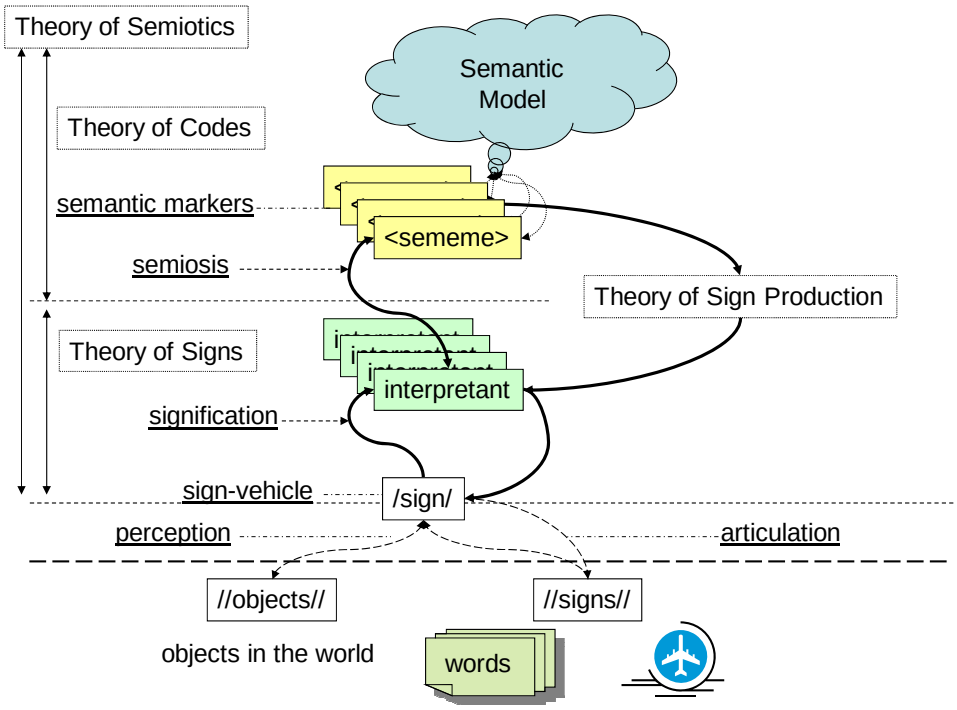

Figure 1: A Semiotic Model Diagramed from synthesizing Eco's theory [3]

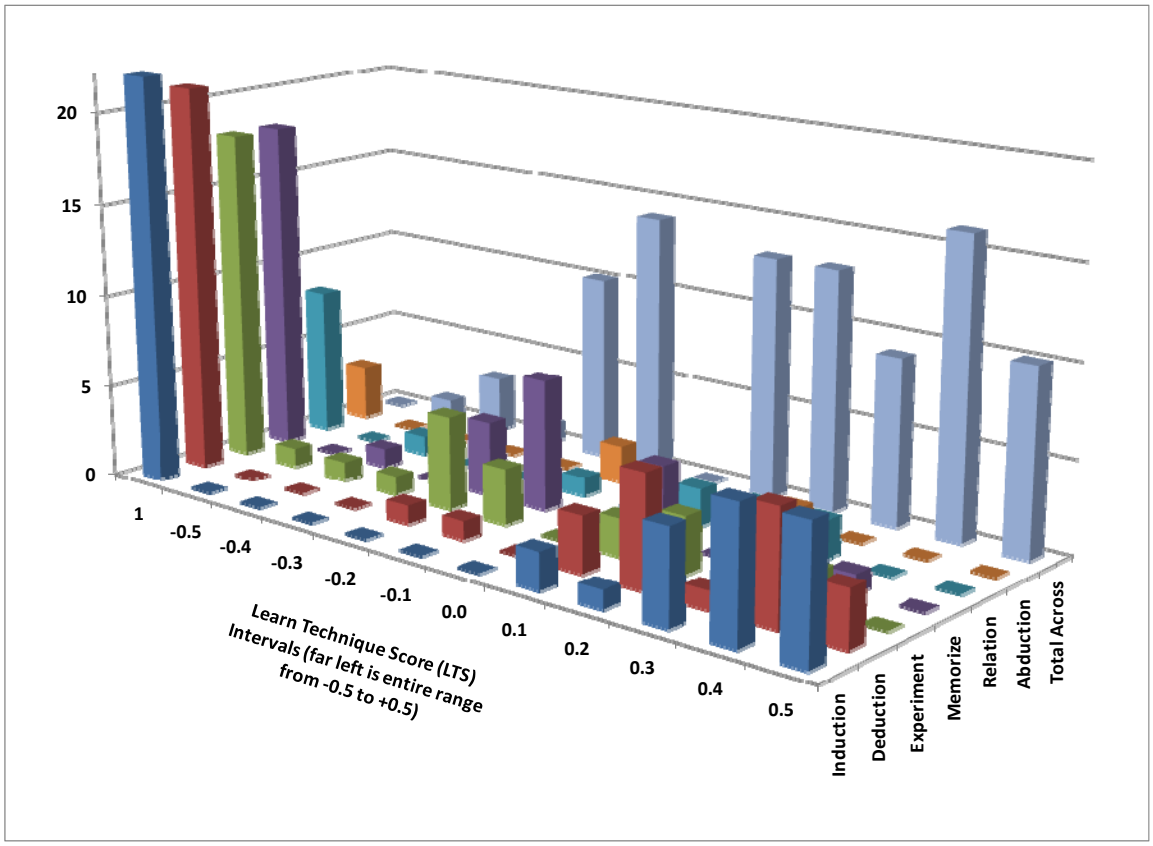

Figure 2: Course Grades Compared to Predicted Course Grades.

Table 6: Characteristics of Learning Technique and Expressive Mode Rankings

LTS Range	N	Learning Technique (LT) Ranked 1st and 2nd		Expressive Mode (EM) *Ranked 1st*	Course Grade Avg.	Cum. Satis-factory	Cum. Poor /Fail	Cum. With-draw
-0.5	1	Experiment	Abduction	Visual	0.0	0%	1%	0%
-0.4	3	Exper.|Rel. Memorize	Memorize Abduction	Visual Reflect	1.5	1%	2%	1%
-0.2	10	Deduction Experiment Memorize	Experiment Abd|Mem|Rel Abd|Ded|Exp	Visual Di|Rf|Vi Di|Rf|Vi	2.5	4%	3%	9%
-0.1	14	Abduction Deduction Experiment Memorize Relation	Ded|Ind Abduction Abduction Ab|De|Exp|In Experiment	Vis|Dis Reflect Dis|Vis Di|Rf|Vi Reflect	2.7	16%	7%	10%
0.1	13	Deduction Experiment Induction Memorize Relation	Exp|Rel Abd|Ded Exp.|Rel. De|Ex|In|Re Abd.|Exp.	Di|Rf|Vi Dis|Vis Visual Dis|Vis Visual	2.8	22%	9%	16%
0.2	13	Abduction Deduction Experiment Induction Relation	Deduction Ind|Mem Abd|De|Ind Relation Ded.|Mem.	Visual Rf|Vi|Di Discuss Reflect Visual	2.4	31%	12%	18%
0.3	9	Deduction Induction Memorize Relation	Relation Ded|Mem|Rel Deduction Ded|Ind	Discuss Rf|Vi|Di Discuss Vis|Dis	3.1	39%	12%	20%
0.4	16	Deduction Exper.|Mem. Induction	Abd|Rel|Ind Induction Ab|Exp|Mem Ded|Rel	Di|Re|Vi Visual Visual Ref|Dis	3.2	52%	13%	23%
0.5	10	Deduction Induction	Induction Abd|Exp Ded.|Rel.	Ref|Vis Visual Di|Re|Vi	3.1	60%	13%	27%

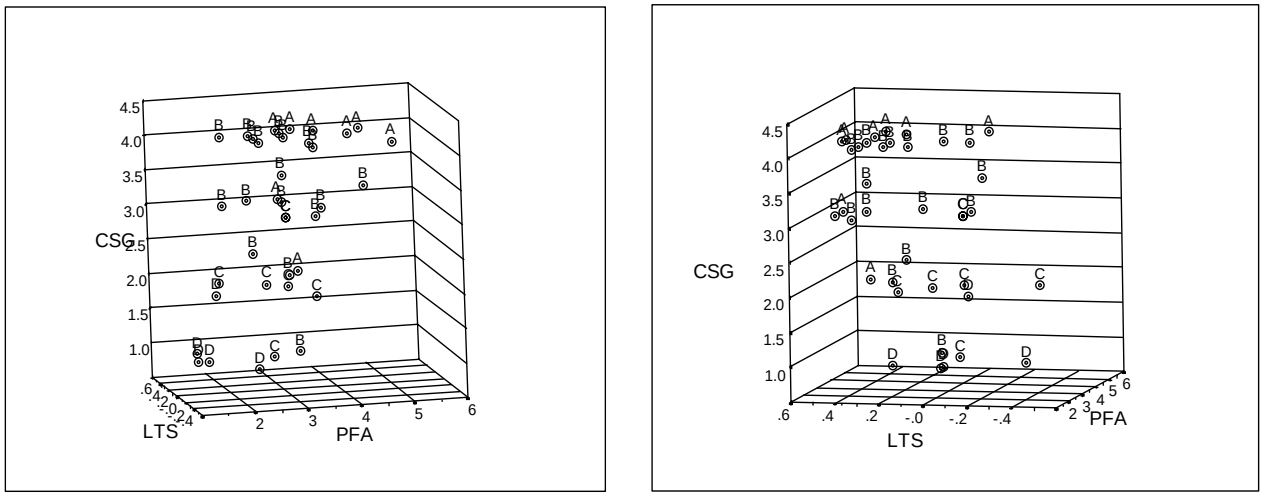

Figure 3: Actual CSG and Predicted CSG (letters) plotted as a function of PFA (left) and LTS (right)