Factors Associated With High-Quality Computer Science Instruction: Data from a nationally representative sample of high school teachers

Eric Banilower
Horizon Research, Inc.
Chapel Hill NC USA
erb@horizon-research.com

Laura Craven
Horizon Research, Inc.
Chapel Hill NC USA
l.craven@horizon-research.com

ABSTRACT

This paper shares results from surveys administered in spring 2018 to a nationally representative sample of nearly 300 U.S. high school computer science teachers. It describes the nature of high school computer science instruction and the extent to which teacher background, classroom factors, and school context predict the type of instruction students experience. Data from the study were analyzed using path modeling—a form of regression analysis that estimates both direct and indirect effects (i.e., through intermediary variables)—to examine relationships between teacher, classroom, and school factors, and the extent to which teachers (1) emphasize reform-oriented instructional objectives (e.g., learning about real-life applications of computer science) and (2) engage students in computer science practices (e.g., recognizing and defining computational problems). Sample findings include that students are most commonly engaged in activities related to testing and refining computational artifacts, but are less often engaged in aspects of computer science related to end users (e.g., create a computational artifact to be used by someone else). The path analysis highlights several factors that are related to greater engagement of students in the computer science practices, including teacher participation in professional development and the use of coherent instructional materials.

CCS CONCEPTS

• Social and professional topics~K-12 education

KEYWORDS

High School Computer Science Education; Instruction; Computer Science Practices

SIGCSE '20, March 11–March 14, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6793-6/20/03…$15.00
https://doi.org/10.1145/3328778.3366831

ACM Reference format:
Eric R. Banilower and Laura M. Craven. 2019. Factors Associated With High-Quality Computer Science Instruction: Data from a nationally representative sample of high school teachers. In *Proceedings of The 51st ACM Technical Symposium on Computer Science Education, Portland, OR, USA, March 11 - March 14, 2020 (SIGCSE'20). ACM, New York, NY, USA, 6 pages.* https://doi.org/10.1145/3328778.3366831

1 INTRODUCTION

National interest in providing computer science education at the K–12 level has grown steadily since ACM founded the Computer Science Teachers Association (CSTA) in 2004. This trend is evident in SIGCSE conference sessions focused on K–12 education, President Obama's Computer Science for All initiative, the growth of online instruction and advocacy sites such as CS for All and code.org, and the development of K–12 computer science standards by CSTA and an increasing number of states. In this paper, we present findings from a large nationally representative survey of high school computer science teachers, a component of the 2018 NSSME+ [3], that provide an answer to the following research question:

> What factors predict high school computer science teachers' classroom teaching practices?

2 BACKGROUND

Efforts to increase access to computer science instruction in K–12 schools have identified a number of factors that affect the ability of schools and teachers to offer high-quality learning experiences. Although many studies of these issues are based on relatively small samples, the findings echo those found in other subjects such as science. Science education research has illuminated many of the factors that influence classroom science instruction. Research points to four broad categories of factors, each with a robust research base: teacher attributes, instructional resources, context-related factors, and inservice support. For example, teacher attributes include various forms of teacher knowledge (e.g., subject matter knowledge), teaching experience, type of teacher preparation program completed, teacher grade level, beliefs about teaching and learning. Several studies have found correlations between teacher attributes and teaching practice (e.g., Smith et al. [21]).

For example, Yadav, Gretter, Hambrusch, and Sands [25] reported that computer science teachers, particularly those new to the content area, lack both the computer science content knowledge and pedagogical knowledge needed to teach computer science effectively. Similarly in science, teacher subject matter knowledge has been found to predict teaching practice (reviewed in Abell [1]).

Teachers' beliefs about how the subject matter can/should be taught and in their own abilities to achieve successful student learning outcomes also influence teaching practice (reviewed in Jones and Carter [10]; see also Nespor [16]; Pajares [17]). Pedagogical approaches are shaped by beliefs about what constitutes effective instruction, which is informed by beliefs about the nature of the discipline and goals of education. Similarly, teaching reflects beliefs about learners in general and a group of students in particular. Beliefs about one's own ability to successfully teach a subject (self-efficacy) also influence teaching practice.

Access to instructional materials, including high-quality units and lessons as well as assessment tools, also have been shown to affect instruction (e.g., Yadav et al. [25]). Research about science teaching has found that the quality of instruction may be determined in large part by curriculum materials, particularly when teachers enact lessons and units with high levels of fidelity [23]. Availability of resources for lessons can create affordances and constraints for classroom pedagogy, and may be important determinants of what a science lesson entails (e.g., Johnson [9]). Finally, access to technological resources such as computers can impact science pedagogy (e.g., Lawless and Pellegrino [13]).

Professional development for teachers can also have a positive effect on instruction. However, a number of studies indicate that short-duration approaches typically have limited success (e.g., Dickinson and Caswell [7]). Other studies indicate that on-going, content-focused, job-embedded experiences are much more likely to have positive impacts (e.g., Supovitz and Turner [22]; Desimone et al. [5]; Luft [14]).

Teaching is also situated within larger contexts whose features affect instruction. For example, policy considerations at the school, district, state, and national levels have impacts on what is and is not done in classrooms (e.g., Knapp & Plecki [11]; Shaver et al. [20]).

3 METHODS

3.1 Study Design

The analyses presented in this paper use data from a nationally representative survey of high school computer science teachers, with a computer science course defined as one that teaches programming or has programming as a prerequisite. The study, conducted during the 2017–18 school year, is based on a national probability sample of 2,000 schools and approximately 10,000 computer science, mathematics, and science teachers in grades K–12. It includes public and private schools in the 50 states and the District of Columbia. The sample was designed to yield national estimates of course offerings and enrollment, teacher background preparation, textbook usage, instructional techniques, and availability and use of facilities and equipment. Every eligible school and teacher in the target population had a known, positive probability of being sampled. A two-stage sample design was utilized. First, a random sample of schools stratified by region of the country, community type, and grades served was selected. Second, a stratified random sample of teachers within schools that agreed to participate was drawn. This sample design resulted in a sample that is both nationally representative and that has sufficient number of cases of various subgroups (e.g., race/ethnicity groups, SES) to allow the data to be disaggregated.

The sampling frame for the school sample was constructed from the Common Core of Data and Private School Survey databases—programs of the U.S. Department of Education's National Center for Education Statistics—which include school name and address and information about the school needed for stratification and sample selection. The sampling frame for the teacher sample was constructed from lists provided by sample schools, identifying current teachers and the specific computer science, mathematics, and science subjects they were teaching. Data collection concluded in July 2018. A total of 7,600 computer science, mathematics, and science teachers in 1,273 schools across the United States participated in this study, a response rate of 78 percent. The data were cleaned (e.g., checked for inconsistencies), and weights reflecting the complex sample design were applied for statistical analyses.

The questionnaires were based on previously used surveys, with new items developed and validated by expert review and cognitive interviews [6]. The study collected a wide range of data about computer science teachers, including teaching experience, degrees earned, college-level computer science courses taken, perceptions of pedagogical and content preparedness, beliefs about teaching and learning, professional development experiences, and adequacy of resources for instruction. In addition, each teacher was asked about instruction (e.g., objectives, instructional practices) in a randomly sampled class. A school-level survey gathered data such as computer science course availability and demographics of the student body.

3.2 Analytic Approach

This paper provides nationally representative data about the state of the computer science teaching force and instruction in the U.S. It then describes an analysis examining how various teacher and school characteristics relate to two teaching practice outcomes measured on the teacher questionnaire. Each outcome was a composite score[1] based on several related survey items. The first, reform-oriented objectives, consists of six items that asked teachers about the emphasis they place on different instructional

[1] Factor analysis was used to create several composite variables related to key constructs measured on the questionnaires. Composite variables, which are more reliable than individual survey items, were computed to have a minimum possible value of 0 and a maximum possible value of 100. Definitions of all of the composite variables used in this study can be found in the Report of the 2018 NSSME+ [3].

goals consistent with the vision for K–12 computer science described in the K–12 Computer Science Framework [11]. The second, engaging students in the practices of computer science, includes 14 items associated with the practices described in the Framework. The items comprising each composite follow.

Reform-Oriented Instructional Objectives Composite: *By the end of the course, how much emphasis will each of the following student objectives receive? [Response options were: None, Minimal, Moderate, Heavy]*

- Learning how to do computer science (e.g., breaking problems into smaller parts, considering the needs of a user, creating computational artifacts)
- Understanding computer science concepts
- Developing students' confidence that they can successfully pursue careers in computer science
- Increasing students' interest in computer science
- Learning how to develop computational solutions
- Learning about real-life applications of computer science

Engaging Students in the Practices of Computer Science Composite: *How often do you have students do each of the following in this class? [Response options were: Never, Rarely (e.g., a few times a year), Sometimes (e.g., once or twice a month), Often (e.g., once or twice a week), All or almost all lessons]*

- Create computational artifacts (e.g., programs, simulations, visualizations, digital animations, robotic systems, or apps)
- Write comments within code to document purposes or features
- Consider how a program they are creating can be separated into modules/procedures/objects
- Identify and adapt existing code to solve a new computational problem
- Provide feedback on other students' computational products or designs
- Systematically use test cases to verify program performance and/or identify problems
- Identify real-world problems that might be solved computationally
- Use computational methods to simulate events or processes (e.g., rolling dice, supply and demand)
- Explain computational solution strategies verbally or in writing
- Create instructions for an end-user explaining how to use a computational artifact
- Compare and contrast the strengths and limitations of different representations such as flow charts, tables, code, or pictures
- Create a computational artifact designed to be used by someone outside the class or other students
- Get input on computational products or designs from people with different perspectives
- Analyze datasets using a computer to detect patterns

Descriptive data about schools, teachers, and classes were generated in WesVar v5.1 [24], using the Jacknife 2 replicate method. In order to estimate the relationship between potential predictors of classroom practice and the outcomes, a path analysis [19] using maximum likelihood estimation was conducted using MPlus v8.2 [15]. Path analysis is a form of regression analysis that estimates both direct and indirect effects (i.e., through intermediary variables) on an outcome. The reform-oriented instructional objectives composite was regressed on a set of independent variables concerning teacher attributes including perceptions of preparedness, availability of instructional resources, stakeholder support, and professional development participation. The engaging students in the practices of computer science composite was regressed on the same set of independent variables, as well as the reform-oriented instructional objectives composite. Prior to analysis, categorical variables in the model were dummy coded and continuous variables checked for normality (transformations were used if normality assumptions were not met). In addition, correlations among the independent variables were examined to guard against multicollinearity issues (highly correlated variables were combined).

4 RESULTS

4.1 Descriptive Statistics

Tables 1–7 show descriptive statistics on the variables used in the model. Because the analyses were conducted using design weights that reflect sampling and subsequent non-response adjustments, the results are nationally representative. The average high school computer science teacher in the U.S. has about 6 years of experience teaching computer science and only one-quarter hold a degree in computer science or computer science education. Thus, it is not surprising that these teachers feel only somewhat prepared when it comes to computer science instruction, in terms of both content and pedagogy (average perceptions of preparedness composite scores around 70). While teachers hold strong reform-oriented teaching beliefs, belief in more traditional instructional methods (e.g., providing students with definitions of key vocabulary at the beginning of instruction) is also prevalent. More than a third of teachers have had full-time job experience in a computer science field prior to teaching. Over half of high school computer science teachers have had more than 35 hours of computer science-related professional development in the last three years.

Table 1: Descriptive Statistics on Teacher-Level Continuous Independent Variables

	Min	Max	Mean	SD
Years of CS Teaching Experience	0	33	6.15	6.51
Perceptions of Preparedness				
Content Preparedness	0	100	64.64	22.80
Pedagogical Preparedness	0	100	67.73	20.49
Implement Instruction in Unit	0	100	71.56	20.44
Traditional Teaching Beliefs	17	100	66.59	18.54
Reform-Oriented Teaching Beliefs	45	100	82.22	11.34

Table 2: Descriptive Statistics on Teacher-Level Categorical Independent Variables

	Percent of Teachers
CS-Related Degree	25
Job Experience in a CS Field	35
Amount of CS PD in Last 3 Years	
None	17
Less Than 35 Hours	28
35 or More Hours	55

Nearly half of high school computer science classes are introductory programming classes and about a third are Advanced Placement (AP) classes; the remainder are specialized or elective courses that have programming as a prerequisite (e.g., robotics, game development). On average, high school computer science classes contain 16 students, and despite the fact that about half of the student body comes from race/ethnicity groups historically underrepresented in STEM (HUS), only 28 percent of computer science students come from these groups.[2] In many classes, the teacher feels strong control over the curriculum, which is reflected in the instructional materials they use. In a large majority of classes, teachers mainly use materials they developed or pulled together from multiple sources; only 12 percent of classes use existing curriculum packages.

Table 3: Descriptive Statistics on Class-Level Continuous Independent Variables

	Min	Max	Mean	SD
Class Size	1	43	16.29	8.57
Percent HUS	0	100	28.33	30.08
Curriculum Control	0	100	78.41	23.98
Extent Computer Issues are Problematic	0	100	14.69	22.61

Table 4: Descriptive Statistics on Class-Level Categorical Independent Variables

	Percent of Classes
Type of CS Course	
Introductory	48
AP	35
Specialized/Elective	18
Prior Achievement Level of Students	
Mostly Average/Mix of Levels	63
Mostly High Achievers	37
Type of Instructional Material Used	
Mainly Curriculum Packages (e.g., commercially published textbooks)	12
Mainly Materials Pulled Together on Their Own	41
Mix of Both	47

High schools that offer computer science are typically located in suburban or urban areas, and, on average, have an enrollment of 1,170 students. Roughly a third of students in these schools are eligible for free/reduced-price lunch (FRL).

Table 5: Descriptive Statistics on School-Level Continuous Independent Variables

	Min	Max	Mean	SD
School Size	70	4664	1170.35	832.01
Percent of Students Eligible for FRL	0	100	32.09	29.09

Table 6: Descriptive Statistics on School-Level Categorical Independent Variables

	Percent of Schools
Community Type	
Rural Area	17
Suburban Area	42
Urban Area	41
Public School	75
Block Scheduling	45

Overall, scores on the reform-oriented instructional objectives composite are fairly high, indicating that high school computer science classes are likely to emphasize these objectives. However, the mean score of 55.62 on the engaging students in the practices of computer sciences composite indicates that, on average, students are engaged in this set of activities relatively infrequently. Further, the range and standard deviation indicate that substantial variation exists among classes in this regard.

Table 7: Descriptive Statistics on Outcome Variables

	Min	Max	Mean	SD
Reform-Oriented Instructional Objectives	33	100	81.97	13.34
Engaging Students in the Practices of CS	7	100	55.62	16.70

4.2 Path Analysis Results

The emphasis on reform-oriented objectives in high school computer science classes is significantly related to several factors; effect sizes[3] (in standard deviations) are shown in Table 8. Teachers' perceptions of preparedness to teach computer science is the strongest predictor of an emphasis on reform-oriented objectives, with an effect size of 0.415 standard deviations (in other words, every 1 standard deviation increase in perceptions of preparedness predicts a 0.415 standard deviation increase in the outcome).

[2] Includes students identified as American Indian or Alaskan Native, Black or African American, Hispanic or Latino, or Native Hawaiian or Other Pacific Islander.

[3] Because different studies may use different instruments or report different kinds of scores, it is difficult to compare results across studies. Effect sizes are used to report results that can be more easily compared across studies. In addition, effect sizes take into account the amount of variation in scores, aiding interpretation of results. Effect sizes of about 0.20 are typically considered small, 0.50 medium, and 0.80 large [4].

Teachers with more than 35 hours of computer science-related professional development in the previous three years were also more likely than those who had not participated in professional development to emphasize these objectives. Classes consisting of mostly high prior achievers and those with higher proportions of students from race/ethnicity groups historically underrepresented in STEM were also more likely to have an emphasis on these objectives (though it is important to note that "higher" is a relative term as most computer science classes have very few students from these groups). Interestingly, classes in which teachers reported having a full-time job in a computer science field prior to teaching and classes in public schools were less likely to emphasize these objectives.

Table 8: Effect Sizes for Reform-Oriented Objectives

	Effect Size
Teacher-Level Predictors	
Perceptions of Preparedness	0.415*
Amount of CS PD in Last 3 Years (vs. None)	
Less Than 35 Hours	0.114
35 or More Hours	0.222*
Job experience in a CS field	-0.145*
Class-Level Predictors	
Mostly High Prior Achievers (vs. Average)	0.170*
Percent HUS	0.130*
School-Level Predictors	
Public School	-0.180*

* $p < 0.05$

Table 9 shows the indirect (I)—through the objectives outcome—, direct (D), and total (T) effect sizes for variables predicting how frequently high school classes were engaged in aspects of the computer science practices. (Total effects combine the direct and indirect effects to provide an estimate of the entire relationship between the independent variable and the outcome.) Again, the effect sizes are measured in standard deviations of the outcome.

Interestingly, participation in 35 or more hours of professional development is, by far, the strongest predictor of this outcome, with a total effect size of 0.561 standard deviations. Teachers' perceptions of preparedness and class size were also positive predictors of engagement with the computer science practices. And consistent with previous research related to science and mathematics instruction [2, 18], teachers who most often use already developed instructional materials (e.g., commercially published textbooks, self-paced online units) as the basis of instruction tend to emphasize these practices more than teachers who pull together their own instructional materials (e.g., findings lessons online, writing their own lessons).

Table 9: Effect Sizes for Engaging Students in CS Practices

	I	**D**	**T**
Teacher-Level Predictors			
Perceptions of Preparedness	0.061*	0.199*	0.260
Traditional Teaching Beliefs	0.009	0.174*	0.183
Amount of CS PD in Last 3 Years (vs. None)			
Less Than 35 Hours	0.017	0.160	0.177
35 or More Hours	0.032	0.529*	0.561
Class-Level Predictors			
Class size	0.020	0.187*	0.207
Type of Instructional Material Used (vs. Mainly Materials Pulled Together on Their Own)			
Mainly Curriculum Packages (e.g., commercially published textbooks)	0.002	0.146*	0.148
Mix of Both	-0.010	0.034	0.024
Reform-Oriented Instructional Objectives	N/A	0.146*	0.146
School-Level Predictors			
Community Type (vs. Suburban)			
Rural	-0.007	0.201*	0.194
Urban	-0.002	0.128	0.126
Block Scheduling	-0.002	0.115*	0.113

* $p < 0.05$

5 DISCUSSION

There is growing momentum in many countries to expand access to computer science in grades K–12, and the U.S. is no exception. However, there has been a lack of large-scale studies examining both the nature of instruction provided to students, and the extent to which it aligns with standards, and the factors that affect teachers' pedagogical practices. Previous studies have identified a number of these factors, though much of this research, in computer science and other disciplines, is based upon small sample sizes. The study from which this analysis stems is the first to provide in-depth data about high school computer science instruction in the U.S., the preparation and background of those teaching it, and contextual factors that can affect the quality of students' learning opportunities. The analyses presented in this paper, though not causal, both highlight several factors that relate to the nature of instruction, including the strength of those relationships, and suggest steps that can be taken to achieve the vision of instruction laid out in the K–12 Computer Science Framework [12].

Not surprisingly, teachers who feel better prepared to teach computer science, both in terms of content and pedagogy, are more likely to engage students in the computer science practices, a finding consistent with research in other content areas. The larger study from which the data used in these analyses come indicates that many computer science teachers have limited preparation to teach computer science, but that they are much more likely to participate in professional development than their mathematics and science teacher counterparts [3]. Thus, the strong relationship between participation in professional development and the outcomes used in these analyses point to the value of investing in such experiences for computer science teachers.

Another important finding is that the type of instructional materials used by teachers has a sizeable relationship with the nature of instruction. Classes in which teachers mainly use instructional units or a whole curriculum developed by others (e.g., commercially published textbooks, self-paced online units) are more likely to engage students in the practices of computer science than those classes in which teachers are pulling together instructional materials from a variety of sources and/or developing their own. This finding parallels ones from mathematics and science education research and point to the importance of providing teachers with high-quality instructional materials. It also illustrates an area of need as very few classes are based on already developed instructional materials. Investing in the development and dissemination of high-quality instructional materials, and providing professional development to help teachers understand the content and pedagogical storylines of the materials, appears to be a reasonable strategy for increasing the quality of computer science instruction at scale [8].

5.1 Suggestions for Future Research

A limitation of this study is that it draws solely on survey data. Surveys are good for measuring how often teachers do things in instruction, but do not provide a direct measure of the quality of instruction. Consequently, the results presented here should be interpreted as the extent students have opportunities to be engaged in the practices of computer science, which are necessary, but not sufficient, to ensure instruction is high quality. An important focus for future research would be to examine the relationship between the factors identified by this study and direct measures of quality of instruction (based on observations, artifact analysis, etc.).

Another limitation is that the analyses presented in this paper are correlational, not causal. Future research could examine the causality of the relationships described in this paper (e.g., via randomized control trials).

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under grant number DGE-1642413. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] S. Abell. 2007. Research on science teacher knowledge. In S. Abell & N.L. Lederman (Eds.), *Handbook of research on science education*. Lawrence Erlbaum Associates, Mahwah, NJ, 1105–1149.

[2] E. R. Banilower, S. E. Boyd, J. D. Pasley, & I. R. Weiss, 2006. Lessons from a decade of mathematics and science reform: A capstone report for the local systemic change through teacher enhancement initiative. Horizon Research, Inc., Chapel Hill, NC.

[3] E. R. Banilower, P. S. Smith, K. A. Malzahn, C. L. Plumley, E. M. Gordon, & M. L. Hayes, 2018. Report of the 2018 NSSME+. Horizon Research, Inc., Chapel Hill, NC.

[4] J. Cohen, 1988. Statistical power analysis for the behavioral sciences. Lawrence Erlbaum Associates, Hillsdale, NJ.

[5] L. M. Desimone, A. C. Porter, M. S. Garet, K. S. Yoon, & B. F. Birman, 2002. Effects of professional development on teachers' instruction: Results from a three-year longitudinal study. *Educational Evaluation and Policy Analysis*, 24, 2, 81–112.

[6] L. M. Desimone & K. C. Le Floch, 2004. Are we asking the right questions? Using cognitive interviews to improve surveys in education research. *Educational Evaluation and Policy Analysis*, 26, 1, 1–22. DOI:10.3102/01623737026001001

[7] D. Dickson & L. Caswell, 2007. Building support for language and early literacy in preschool classrooms through in-service professional development: Effects of the Literacy Environment Enrichment Program (LEEP). *Early Childhood Research Quarterly*, 22, 2, 243–260.

[8] D. J. Heck, K. B. Chval, I. R. Weiss, & S. W. Ziebarth, 2012. Developing measures of fidelity of implementation for mathematics curriculum materials enactment. In D. J. Heck, K. B. Chval, I. R. Weiss, & S. W. Ziebarth (Eds.), *Approaches to studying the enacted mathematics curriculum*. Information Age Publishing, Inc., Charlotte, NC, 67–88.

[9] C. C. Johnson, 2006. Effective professional development and change in practice: Barriers science teachers encounter and implications for reform. *School Science and Mathematics*, 106, 150–161.

[10] M. G. Jones & G. Carter, 2007. Science teacher attitudes and beliefs. In S. Abell & N.L. Lederman (Eds.), *Handbook of research on science education*. Lawrence Erlbaum Associates, Mahwah, NJ, 1067- 1104.

[11] M. S. Knapp & M. L. Plecki, 2001. Investing in the renewal of urban science teaching. *Journal of Research in Science Teaching*, 38, 1089–1100.

[12] K–12 Computer Science Framework. 2016. Retrieved from http://www.k12cs.org.

[13] K. Lawless & J. Pellegrino, 2007. Professional development in integrating technology into teaching and learning: Knowns, unknowns, and ways to pursue better questions and answers. *Review of Educational Research*, 77, 4, 575-614.

[14] J. Luft, 2009. Beginning secondary science teachers in different induction programmes: The first year of teaching. *International Journal of Science Education*, 31, 17, 2355-2384.

[15] B. O. Muthen & L. K. Muthen, 2018. Mplus (Version 8.2) [Computer software]. Retrieved from https://www.statmodel.com/

[16] J. Nespor, 1987. The role of beliefs in the practice of teaching. *Journal of Curriculum Studies*, 19, 4, 317–328.

[17] M. F. Pajares, 1992. Teachers' beliefs and educational research: Cleaning up a messy construct. *Review of Educational Research*, 62, 3, 307–332.

[18] W. R. Penuel & L. P. Gallagher, 2009. Preparing teachers to design instruction for deep understanding in middle school earth science. *Journal of the Learning Sciences*, 18, 4, 461-508. DOI: 10.1080/10508400903191904

[19] R. E. Schumacker & R. G. Lomax, 1996. A Beginner's Guide to Structural Equation Modeling. Lawrence Erlbaum Associates, Mahwah, NJ.

[20] A. Shaver, P. Cuevas, O. Lee, & M. Avalos, 2007. Teachers' perceptions of policy influences on science instruction with culturally and linguistically diverse elementary students. *Journal of Research in Science Teaching*, 44, 725–746.

[21] T. Smith, L. Desimone, T. Zeidner, A. Dunn, M. Bhatt, & N. Rumyantseva, 2007. Inquiry-oriented instruction in science: Who teaches that way? *Educational Evaluation and Policy Analysis*, 29, 3, 169–199.

[22] J. A. Supovitz & H. M Turner, 2000. The effects of professional development on science teaching practices and classroom culture. *Journal of Research in Science Teaching*, 37, 9, 963–980.

[23] I. R. Weiss,, J. D. Pasley, P. S. Smith, E. R. Banilower, & D. J. Heck, 2003. Looking inside the classroom: A study of K–12 mathematics and science education in the United States. Horizon Research, Inc., Chapel Hill, NC.

[24] Wesvar, 2015 (Version 5.1) [Computer software]. Retrieved from https://www.westat.com/capability/information-technology/wesvar

[25] A. Yadav, S. Gretter, S. Hambrusch, & P. Sands, 2016. Expanding computer science education in schools: understanding teacher experiences and challenges. *Computer Science Education*, 26, 4, 235–254.