Section 1:
Computing in context

Measuring the enacted K-12 computing curriculum

Elizabeth Cole (University of Glasgow, UK),
Monica McGill (CSEdResearch.org, USA),
Keith Quille (Technological University Dublin, Ireland),
and Rebecca Vivian (University of Adelaide, Australia)

Cole, E., McGill, M., Quille, K. and Vivian, R. (2021). Measuring the enacted K-12 computing curriculum. In Understanding computing education (Vol 1). Proceedings of the Raspberry Pi Foundation Research Seminar series.

Available at: rpf.io/seminar-proceedings-2020

Raspberry Pi

Section 1: Computing in context

Measuring the enacted K-12 computing curriculum

Elizabeth Cole (University of Glasgow, UK), Monica McGill (CSEdResearch.org, USA), Keith Quille (Technological University Dublin, Ireland), and Rebecca Vivian (University of Adelaide, Australia)

Abstract

The *intended curriculum* — the curriculum that is intended to be taught through policy, curriculum documents, or other required mandates — and the *enacted curriculum* — the curriculum that is actually taught in classrooms by teachers — are ideally aligned. However, often there is a chasm between the two. With computing education being relatively new to schools and teachers across many countries, we wanted to learn if a chasm existed and, if it did, how wide it is across different countries. Working as part of an international team, we created a set of templates for measuring intended curricula and a survey instrument, MEasuring TeacheR Enacted Curriculum (METRECC), to measure enacted curricula. The original pilot investigated the enacted curriculum in seven countries (with 244 teacher participants). Our research found that both visual and text-based programming languages are being used across K-12, warranting further research into potential impact on student learning and motivations. Unplugged activities are commonly used across K-12, extending into later years despite not being explicitly defined in intended curricula. Further, teachers' motivations for programming language choice are consistent across countries and our study revealed that student-driven factors motivate selection. This initial study was followed by additional analysis with respect to teacher self-esteem that was found to differ across multiple factors such as experience in teaching CS in years and gender. We punctuate our work with the adaptation of the instrument for use in South Asia and a call to the community to consider middle- and low-income nations in future research.

Introduction

While there has been some efforts to collate intended curricula, internationally, nationally, and regionally (Porter and Smithson, 2001; Hubwieser, Armoni, and Giannakos, 2015; Gander et al., 2013; Balanskat and Engelhardt, 2014; The Royal Society, 2012; The Royal Society, 2017; Hong et al., 2016; Moller and Crick, 2016; Sysło and Kwiatkowska, 2015), there is a need to understand the enacted curriculum and how well it aligns with the curriculum as intended to be taught. In June 2019, a working group led by Sue Sentance, Katrina Falkner, and Rebecca Vivian at the Association of Computing Machinery's (ACM) 2019 conference on Innovation and Technology in Computer Science Education (ITiCSE), conducted an international study of K-12 computer science (CS) implementation across Australia, England, Ireland, Italy, Malta, Scotland, and the United States. The purpose of the study was to develop instrumentation that would provide descriptive data about the intended and enacted computing education curriculum in K-12 schools. The genesis of the work arose from the analysis of previous attempts to assess and identify K-12 CS implementation efforts internationally.

This chapter describes three major efforts related to this research:

- Results originating from a detailed collaborative international process which resulted in the design, pilot, and evaluation of an international survey instrument for Measuring Teacher Enacted Computing Curriculum (METRECC) (Falkner, 2019a; Falkner, 2019b)
- Results of a deeper analysis of the self-esteem scale that was included as part of the METRECC survey (Vivian et al., 2020)
- The adaptation of METRECC for usage in middle- and lower-income nations, with a focus on South Asian classrooms (Anwar et al., 2020)

While some quantitative and qualitative reports existed with respect to measuring the intended and enacted curriculum, none succinctly provided a tool that could be applied at scale and easily provide the descriptive data for international comparison. During the initial planning phase of our study, it became apparent that we needed a formal method for framing the intended curriculum as well as instrumentation to capture the curriculum as it is taught in the K-12 classrooms. Both instruments also needed to be framed in a way that took into account the various differences in primary and secondary structures across seven countries.

In addition to providing evidence of the validity of the survey instrument, the pilot phase reviews of the curriculum landscapes from Europe, the UK, and the US provide some information on international CS K-12 efforts. This work also laid the foundation for several follow up studies, one of which looks specifically at teacher computer science self-esteem (Vivian, 2020) across the seven jurisdictions using the pilot data. Teacher self-esteem was found to differ across multiple factors such as experience in teaching CS in years and gender. This work might be useful to CS educators who are designing CPD or pre-service teacher programmes.

Many of the countries examined in the original study consisted of high-income nations, while little is known about primary and secondary computing education efforts in both middle- and low-income nations. Another follow-up study, therefore, reinterpreted METRECC for use in middle- and low-income countries in South Asia (Anwar et al., 2020). This pilot of the METRECC South Asia instrument is a step towards the validation of the instrument across nations with varying socio-economic demographics. This work sets a solid foundation for the continued longitudinal implementation of the METRECC instrument to further investigate international enacted curricula.

Measuring the intended and enacted curriculum

An initial body of work focused on the alignment between the intended and enacted curriculum in the areas of topics taught and programming languages used (Falkner, 2019b). These critical areas of CS curricula require further analysis and monitoring not only in terms of alignment and its ensuing benefits, but also in relation to our assumptions as tertiary educators on prerequisite knowledge and experience.

Two instruments were developed for the study by the 2019 ITICSE working group: a country report template to capture the *intended curriculum* and a teacher survey instrument to capture the *enacted curriculum*. The following explanation briefly describes the instruments with more detail available in the published report (Falkner et al., 2019a).

COUNTRY/USA STATE	AUSTRALIA (AUS)	COLORADO (US-CO)	ENGLAND (ENG)	IRELAND (IRL)	ITALY (ITA)	ILLINOIS (US-IL)	MALTA (MLT)	MINNESOTA (US-MN)	SCOTLAND (SCO)
Population (million)	25.09	5.69	55.62	4.70	60.50	12.7	0.47	5.6	5.44
No. of schools	9477	1900	29972	3961	8636	4266	170	2066	2400
No. Primary schools				3246			108		2031
No. secondary schools				715			62		359
No. of students	3893834	911536	8378809	920867	8422419	2072880	46247	862971	693251
No. of teachers (FTE)	288583	59989	498100	66327	872268	135701	2976	57262	51959
No. of Primary teachers (FTE)				36773					
No. of secondary teachers (FTE)				29554					
CS State or country plan	√	⊗	√	⊗	∅	⊗	√	⊗	√
CS Curriculum k-6 standards defined	√	⊗	√	∅	∅	⊗	√	⊗	√
CS Curriculum: Y7+ standards defined	√	√	√	∅	∅	⊗	√	⊗	√
CS Standalone subject	√	√	√	∅	∅	⊗	∅	∅	√
CS Formal Reporting	V	⊗	⊗*	⊗	∅	⊗	∅	∅	∅
CS in pre-service training Primary	E	E	√	E	√	E	⊗	⊗	E
CS in pre-service training Secondary	E	E	√	E	E	E	√	⊗	√
CS training for inservice Primary?	V	√		V				√	
CS training for inservice secondary?	V	√		V				√	
Year endorsed	2015	2018	2013/14	⊗	⊗	⊗	2018*	⊗	2016*
CSTA CONCEPTS (2017)	Intended Curriculum								
Computational Thinking	√	√	√	√	√	⊗	√	⊗	√
Computer Systems	√	∅	∅	√	√	⊗	∅	⊗	√
Networks and Internet	√	∅	√	√	√	⊗	√	⊗	√
Data & Analysis	√	√	√	√	√	⊗	√	⊗	√
Algorithms and Programming	∅	∅	NA	√	√	⊗	∅	⊗	√
Impact of Computing	√	√	√	√	√	⊗	√	⊗	√
Other areas not covered above									

(i) Yes (√) No (⊗) Additional information (∅) (ii) Pre-service training - Varies(V) Compulsory (√), Elective (E)
(iii) CSTA standards covered Explicit (√) Implicit (∅) Not covered (⊗) *Date previous CS curriculum refreshed.

COUNTRY/USA STATE	AUS			US-CO			ENG			IRL			ITA		
Age*/US Grade	TC	PL	CC	TC	PL	CC	TC	PL	CC	TC	PL	CC		PL	CC
2+															
3-4															
4-5 Kindergarten	√	⊗	√	√	⊗	O	√	VP	√						
5-6 Grade 1	√	⊗	√	√	⊗	O	√	VP	√	⊗	⊗	NA	S	VP	C
6-7 Grade 2	√	⊗	√	√	⊗	O	√	VP	√	⊗	⊗	NA	S	VP	C
7-8 Grade 3	√	VP	√	√	⊗	O	√	VP	√	⊗	⊗	NA	S	VP	C
8-9 Grade 4	√	VP	√	√	⊗	O	√	VP	√	⊗	⊗	NA	S	VP	O
9-10 Grade 5	√	VP	√	√	⊗	O	√	VP	√	⊗	⊗	NA	S	VP	O
10-11 Grade 6			√						√	⊗	⊗			VP /G	

COUNTRY/USA STATE	US-LO			MLT			US-MN			SCO		
Age*/US Grade	TC	PL	CC	TC	PL	CC	TC	PL	CC	TC	PL	CC
2+												
3-4											VP	
4-5 Kindergarten							√			S	VP	√
5-6 Grade 1	⊗	⊗	⊗	√	⊗	O	√	⊗	⊗	S	VP	√
6-7 Grade 2	⊗	⊗	⊗	√	⊗	O	√	⊗	⊗	S	VP	√
7-8 Grade 3	⊗	⊗	⊗	√	⊗	O	√	⊗	⊗	S	VP	√
8-9 Grade 4	⊗	⊗	⊗	√	⊗	O	√	⊗	⊗	S	VP	√
9-10 Grade 5	⊗	⊗	⊗	√	⊗	O	√	⊗	⊗	S	VP	√
10-11 Grade 6				√								√

Figure 1. Country report templates for the METRECC

Capturing the intended curriculum

The country report template is designed to be completed by the survey administrator and captures the country demographics relating to schools and the intended curriculum. For analysis and comparison, the broad curriculum and information for each country is organised as follows:

1. Demographics (e.g. such as total population, number of schools, number of teachers)
2. CS curriculum state or country plan standards and requirements
3. Year level (with age for comparisons) mapped to prescribed curriculum and programming requirements
4. General CS topics covered

Although the aim is to analyse intended and enacted curricula for individual states/countries, the templates show similarities and differences across states/countries. However, a key consideration for high-level country by country intended curricula comparison is the CS content variations. To overcome the issue of variance in the short term, the country report template includes a list of broad CS topics based on available literature. Future work will refine this list based on the pilot study results.

Capturing each country's enacted curriculum
The Measuring Teacher Enacted Computing Curriculum (METRECC) instrument captures the enacted curriculum. The survey measures what teachers are doing in the classroom, taking account of their context. A set of key categories of interest internationally in terms of the enacted CS education curriculum were curated and, where possible, refined from existing surveys with evidence of reliability and validity. The developed survey includes 11 sections and a total of 53 questions, including:

1. Introduction
2. Demographics
3. Current work (position)
4. Qualifications
5. Student composition
6. Support and resourcing
7. Assessment of student learning
8. Classroom practice and motivation
9. Self-efficacy/self-esteem
10. Professional development
11. Consent for publishing data

The intention was that each section can be administered independent of each other, with survey administrators being able to piece together a survey to suit their needs.

Findings

The final dataset includes 244 responses across 7 countries: USA (n=115), England (n=52), Italy (n=20), Ireland (n=19), Scotland (n=18), and Malta (n=6). The highlights of the demographic data showed that the respondents had these characteristics:

- 61% female; 37% male
- 87% ages 30 to 59
- 49.6% teaching for 12 or more years
- 89% from government/public schools
- 36% from disadvantaged schools
- 29% rural/remote areas; majority urban/metro
- All were teaching computing in school in some capacity across age 3 to 19
- The majority of respondents are teaching across middle school and secondary (age 13 to 17)

The pilot study sample below investigates the questions relating to curriculum topics and programming languages enacted in the classroom for comparison against intended curriculum requirements.

Country reports (intended) curriculum broad topics covered

Firstly, we examined topics that featured in the intended curriculum across the several countries as a comparison point for what teachers were working with when it comes to their enacted curriculum. Table 1 presents the results for the intended curriculum — broad topics that are explicitly or implicitly defined in country or state curriculum documents as identified in the country report snapshot. This information was used to not only compare countries or states but to also map similarities and gaps in enacted curriculum.

Concepts	AUS	US-CO	ENG	IRL	ITA	US-IL	MLT	SCO
Computational Thinking	✓	✓	✓	✓	✓	×	✓	✓
Computer Systems	✓	❖	✓	✓	✓	×	❖	✓
Networks and Internet	✓	❖	✓	✓	✓	×	✓	✓
Data & Analysis	✓	✓	✓	✓	✓	×	✓	✓
Algorithms and Programming	❖	❖	✓	✓	✓	×	❖	✓
Impact of Computing	✓	✓	✓	✓	✓	×	✓	✓

Table 1. Country report information of topics featured explicitly (✓), implicitly (❖), or not at all (X) in intended curriculum across countries

CS Topics	Australia	England	Ireland	Italy	Malta	Scotland	USA
Algorithms	**79%***	**100%***	**68%***	**70%***	**33%***	**100%***	**82%***
Artificial Intelligence	7%	44%	32%	10%	0%	6%	**30%***
Computational Thinking	**57%***	96%	68%	**45%***	17%	**89%***	72%
Cybersecurity	71%	83%	16%	35%	17%	**72%***	**57%***
Data analysis and visualisation	**29%***	44%	26%	25%	0%	11%	**43%***
Data representation (e.g. digital data, binary)	**57%***	**88%***	**53%***	**45%***	**33%***	100%	**68%***
Databases	14%	71%	42%	**45%***	**17%***	89%	*27%
Design process (or Design Thinking)	**86%***	**54%***	**58%***	**20%***	17%	56%	72%
Ethics	**29%***	**88%***	58%	35%	0%	**56%***	**75%**
Hardware	**26%***	90%	**68%***	**55%***	**50%***	**94%***	**61%***
Information Systems	**50%***	58%	21%	**30%***	33%	**72%***	35%
Machine Learning	7%	23%	26%	5%	17%	11%	21%
Networks and Digital Systems	**64%***	90%	16%	40%	**17%***	**39%***	**45%***
Privacy	**64%***	77%	42%	40%	**17%***	**61%***	**64%***
Programming skills and concepts	**79%***	**100%***	**100%***	**80%***	**50%***	**100%***	**87%***
Robotics	79%	33%	42%	40%	**50%***	11%	47%
Web Systems	36%	62%	**37%***	**50%***	17%	**94%***	38%
Total sample (n)	14	52	19	19	6	18	115

Table 2. Percentage of teachers teaching CS topics across countries (asterisk denotes the topic is part of the country's intended curriculum)

In the example above we can easily see differences, such as where formal curricula are not available for CS topics and that in a number of countries topics taught are implicitly defined in the curriculum rather than explicitly.

Survey responses (enacted) curriculum broad topics covered

In Table 2 data from the country reports is mapped against the percentage (%) of teachers who indicated "yes, I teach this [topic]" in the survey. An asterisk indicates that the topic is part of the intended curriculum, providing a comparison point in terms of what topics are expected to be covered and what teachers are explicitly teaching.

Topics mostly being taught across countries are algorithms, programming, computational thinking, and data representation. Topics taught less frequently are machine learning and artificial intelligence. We can also see gaps, particularly in Malta and some topics within countries.

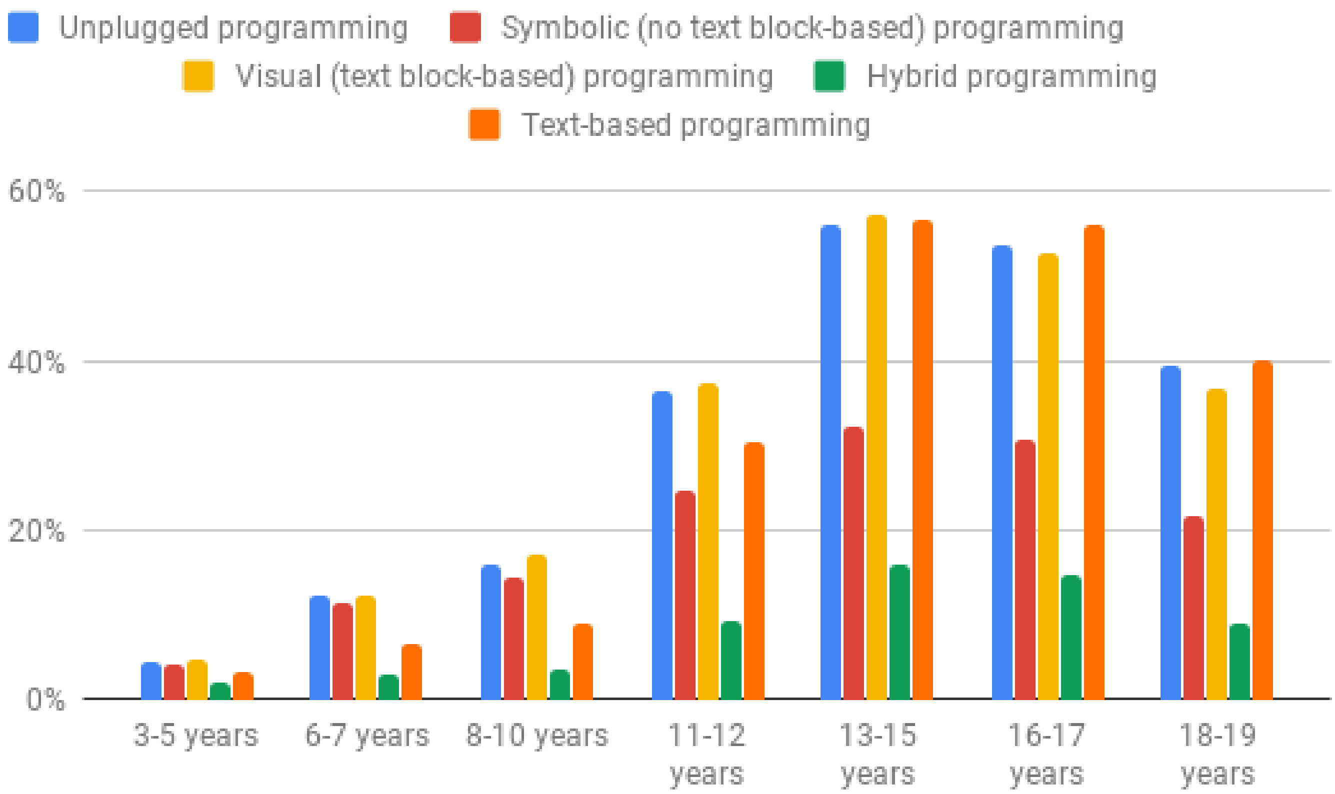

Figure 2. Percentage of teachers using programming environments according to year level bands

Programming paradigms

Exploring programming paradigms as implemented by classroom teachers provides interesting insight as differences might be attributed to a range of reasons such as intended curriculum, but also the resources they have available and student needs. We present results from what programming environments teachers reported implementing in their classrooms according to age groups (see Figure 2). In our analysis we see that programming environments reflected what would be expected across age groups, for example a shift from symbolic programming tools (e.g. Scratch JR) toward block-based environments (e.g. Scratch) and text-based programming environments (e.g. Python). We also noted a strong presence of 'unplugged' programming experiences across age groups

We were then able to take this information about what teachers reported and compare country data we acquired in the country reports that identified any curriculum requirements around programming environments according to age groups.

An example of country-level programming paradigm analysis used England as a test case with the greatest representation of teachers across grade levels. We present teachers'

Ages	Unplugged	Symbolic (no text)	Visual (text)	Hybrid	Text-Based
3-5	8%	**6%**	**10%**	4%	10%
6-7	17%	**15%**	**17%**	6%	13%
8-10	21%	**17%**	**21%**	6%	**13%**
11-12	60%	40%	54%	10%	**58%**
13-15	65%	37%	58%	10%	**73%**
16-17	63%	31%	52%	8%	**67%**
18-19	48%	21%	33%	2%	**48%**

Table 3. Percentage of teachers in England using programming environments according to age group (highlighted cells indicates where the environment is advised in intended curriculum)

reported use in Table 3 with what is identified in the intended curriculum as yellow highlighted cells. Lower percentages of teachers reported using symbolic programming (no text — such as flowcharting, describing sequences as steps, etc.) and visual programming in elementary grades. The lack of teaching with symbolic and visual coding illustrates a mismatch in the intended and enacted curriculum. Higher percentages of teachers reported using text-based programming environments matching more closely the intended curriculum.

Self-esteem of CS teachers

The foundational research conducted by the working group in 2019 captured a wide range of data, thus allowing international comparisons on a multitude of factors, such as the aforementioned enacted curriculum. In 2020, the working group further examined one of the constructs in detail, the teachers' computer science self-esteem (Vivian et al. 2020).

Study background

The Bergin programming self-esteem (Bergin, 2006; Quille & Bergin 2019) construct has shown strong prediction capability in previous studies using CS1 students at third-level, and reported insights when CS1 subcohorts were compared such as by gender, performance, or age (Quille, Culligan, and Bergin, 2017; Quille & Bergin, 2020). The construct was adapted for K-12 teachers and included in the MEasuring TeacheR Enacted Curriculum (METRECC) pilot study. The study consisted of 219 teachers across the seven countries who completed this construct in the METRECC survey. This scale was designed to determine if there were any differences in teachers' computer science self-esteem by country, teacher age, teacher computer science experience, the age groups that are being taught by the teacher, teaching location (rural, metro, etc.), and by gender. The goal of this work was to identify insights that might inform future curriculum developments and teacher PD design and implementation.

Country	N	%
USA	98	45
England	49	22
Italy	18	8
Ireland	19	9
Scotland	17	8
Australia	14	6
Malta	4	2
Total	**219**	**100**

Age Range	N	%
18 - 29	7	3
30 - 39	52	24
40 - 49	70	32
50 - 59	70	32
60 or over	20	9
Total	**219**	**100**

Table 4. Participants per country and age demographics

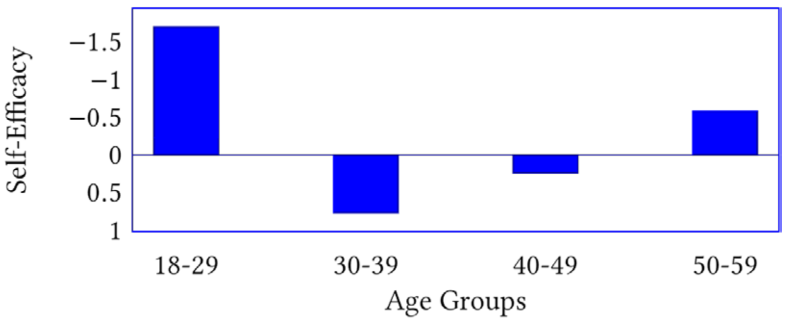

Figure 3. Teachers' computer science self-esteem by age group. (The y-axis scale is inverted, as a negative value represents a positive computer science self-esteem, and a positive value represents a negative computer science self-esteem. This is due to the data reduction algorithm applied, Principal Component Analysis -PCA.)

Results

The participants were primarily from the US and England and were predominantly 40 to 59 years old (see Table 4). Though the majority of teachers were from the US and the UK (n=67%), if student population were considered, the US would be significantly underrepresented. However, as a pilot study, we were evaluating the scale for evidence of reliability and validity.

A comparison was conducted across each of the seven countries. No statistically significant differences between the teachers' computer science self-esteem were found.

Since previous studies identified programming self-esteem differences by student age, this led the working group to investigate across teachers' age groups as used in the METRECC survey. Figure 3 presents the teachers' computer science self-esteem per age group. There was no statistically significant difference between the age groups (using a one-way ANOVA test where $F(4, 214) = 1.5485$, $p = 0.1893$. A Tukey HSD post-hoc test was also administered which confirmed the findings of the one-way ANOVA test) while acknowledging that visually there are

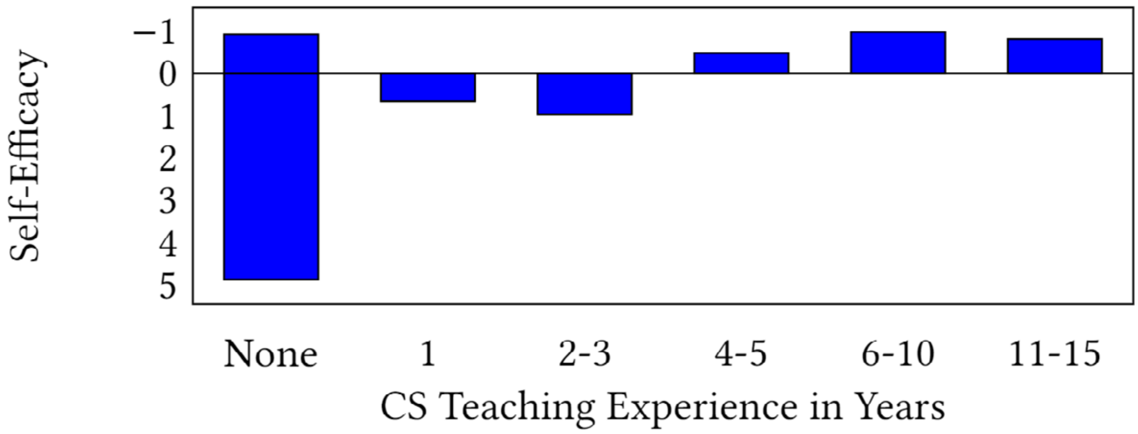

Figure 4. Teachers' computer science self-esteem by CS teaching experience

differences in Figure 3. An interesting finding is that teachers between the age of 30 and 49 have the lowest CS self-esteem. Though the sample size of teachers in the 18 to 29 age group is relatively small, they have the most positive CS self-esteem. Teachers 50 years or older also present a higher positive CS self-esteem to that of the 30 to 49 group. This could suggest a need to focus PD towards specific age groups, however, further work is required to investigate reasons for these age-related variations.

The working group further investigated if the CS teaching experience in years reported differences as reported in the METRECC study. We found a statistically significant difference between groups where a deeper analysis reported that the difference was between the "no experience group" and all other groups. This was expected as K-12 teachers who have not taught a formal CS class would not be expected to have a high CS self-esteem. The results of this comparison are presented in Figure 4. In addition, another insight from Figure 4, illustrates that teachers with four to five years of CS experience or more report a positive computer science self-esteem (where teachers with less than this experience report a negative computer science self-esteem), where perhaps this is the minimum time required to be confident to teach the subject.

Next the working group examined teachers' computer science self-esteem by the age groups being taught (this was binomial with this analysis categorising teachers by primary- or second-level education) and by teaching location (as categorised by metro, urban, rural, or remote). The difference is statistically significant with primary teachers reporting lower self-esteem than secondary teachers. Future studies could compare the level of CS required for primary- and second-level teachers (where secondary content and depth would be significantly more advanced), to unpack this finding. On a positive note, teaching location did not report significant teachers' computer science self-esteem differences, where PD availability was hypothesised to be a factor for more remote teachers.

Finally, the working group investigated teachers' computer science self-esteem differences based on gender. Previous work has reported for CS1 students significant differences in programming self-esteem (Quille & Bergin, 2017), thus this work was to identify if the same findings were

Initial METRECC Study	Rank	Index
Ireland	3	0.918
Australia	6	0.923
United Kingdom	15 [sic]	0.916
United States	15 [sic]	0.899
Malta	28	0.818
Italy	29	0.793
Average	16	0.883

Table 5. 2018 Education index as specified by United Nations Development Programme for the countries involved in the original METRECC study

South Asia	Rank	Index
Afghanistan	170	0.413
Bangladesh	135	0.513
Bhutan	163	0.441
India	129	0.558
Maldives	104	0.564
Nepal	147	0.501
Pakistan	152	0.407
Sri Lanka	71	0.756
Average	134	0.519

Table 6. 2018 Education index as specified by United Nations Development Programme for South Asian countries

present for K-12 teachers' computer science self-esteem. The difference in teachers' computer science self-esteem reported is statistically significant. These findings align with the aforementioned research (although this study had no metric to examine teacher performance) and prompts questions for future research, such as: "Are male teachers overrating their CS self-esteem, while female teachers underrate theirs?" and "If males and females correctly rate their CS self-esteem, why do female teachers have significantly lower CS self-esteem than males?".

Adapting METRECC for the South Asian classrooms

The research conducted by our working group and highlighted in the previous sections focused on high-income nations, as designated by the World Bank (Table 5) (World Bank, 2020). But, little is known about primary and secondary computing education efforts in both middle- and low-income nations. When examined in aggregate, K-12 computing education research in low and lower-middle income countries is underrepresented in many international publication venues. This can introduce a false understanding as well as biases and prejudice against these nations and paint a deceptive picture that we are achieving more than we actually are on an international scale.

To investigate this, we conducted a follow-up study focusing on K-12 computing education in South Asia, in which limited research on computing education exists and with low- and middle-income countries (Anwar, et al., 2020) (Table 6). Rather than covering all eight countries in South Asia, we instead chose to focus on four: Bangladesh, Nepal, Pakistan, and Sri Lanka. These countries were chosen because they provide a mix of one low-income country (Nepal), two lower-middle income countries (Bangladesh and Pakistan), and one upper-middle income country (Sri Lanka). Further, it gave us a mix of three countries with lower ranked education indices (Bangladesh, Nepal, and Pakistan) and one from a higher ranked index (Sri Lanka). We also note that two authors of this study were from Pakistan, one was from Nepal, and three had connections with Sri Lanka and Bangladesh.

To gauge what types of enacted curriculum are taught in the classrooms, we carefully modified METRECC to meet the cultural and current educational climate of these countries. Before trialing the new METRECC South Asia instrument, we modified it based on our cultural knowledge of the countries, then acquired feedback from professionals in South Asia and modified the instrument again based on their input. We then tested this instrument across Nepal and Pakistan. Based on the results and the face validity performed on the instrument, we consider METRECC South Asia to be ready for larger scale usage with recommendations for a broader study. In addition, we consider the role and importance of computing education research in low-income countries in order to support the beliefs and values of the CS for All movement.

In light of the social justice and economic prospects promoted through the CS for All initiative across high-income countries, our literature fails to step back and understand the tens of millions of students in low and lower-middle income countries who deserve the opportunity to lift their families and their countries from their low-income status. Exploration of other ways to improve and to enable low-income countries to achieve their technological aspirations as noted in several plans (Ministry of Education Sri Lanka, 2015; Government of Pakistan, 2018) could prove to be invaluable in enabling these countries to address gaps in their education (Coloma and Harris, 2009; Wikramanyake, 2014). These nations' leaders understand that CS for All is vital to their nation's growth. Therefore, though our research into this area is just starting, we hope that this study gives readers a chance to reflect and consider how we can be more aware of the CS for All movement across the full international landscape.

Conclusion and future work

This chapter brings together the original enacted curriculum investigation work (the pilot study administered during the development of the METRECC instrument) with two follow-up bodies of work, including expanding to efforts in both middle- and low-income nations. This work seeks to understand the intended curriculum and enacted curriculum to determine if what is intended to be taught through policy, curriculum documents, or other required mandates are being implemented by classroom teachers. We identified in our country analysis differences between what is expected to be taught across countries in terms of CS topics and programming environments, as well as differences between what is expected to be taught and what teachers reported implementing in classroom learning.

This work provided some early insights on teachers' computer science self-esteem, but it was a pilot in its nature based on teacher sample size and jurisdiction representation. More work is required to conduct a broader study and re-validate the findings. In addition, a deeper analysis is required (such as qualitative data collection, interviews, etc.) to unpack the insights presented in this section of work. These preliminary findings, however, could be valuable to professional learning developers, for developing a more differentiated suite of professional learning sessions.

This work has demonstrated the value in including both middle- and low-income nations in the design and development of instruments to capture and monitor intended and enacted curriculum across the world, however, more work can be done to extend METRECC to be customised for other contexts. By continuing to capture more countries and to replicate METRECC, we will be able to gather more comparison data and to monitor changes over time. Throughout this work we have identified some differences between what is intended to be taught and what is taught, as well as some slight differences in teacher confidence. This opens up more questions and opportunities to further explore the METRECC dataset and to continue work to investigate the potential reasons for these differences, for example whether it be due to qualifications, professional learning, resources, or other factors.

This body of work highlights the importance of international researchers working together with a strategic effort to monitor what is happening in CS classrooms around the world. By working together and collaboratively developing and validating instruments, we are working toward a common goal and more robust and consistent ways of being able to capture and monitor our CS education landscapes.

References

Anwar, T., Jimenez, A., Bin Najeeb, A., Upadhyaya, B., & McGill, M. M. (2020, August). Exploring the enacted computing curriculum in K-12 Schools in South Asia: Bangladesh, Nepal, Pakistan, and Sri Lanka. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (pp. 79-90).

Balanskat, A., & Engelhardt, K. (2014). Computing our future: Computer programming and coding-Priorities, school curricula and initiatives across Europe. European Schoolnet.

Bergin, S (2006). *A computational model to predict programming performance*. PhD thesis, Department of Computer Science, Maynooth University.

Coloma, J., & Harris, E. (2009). From construction workers to architects: developing scientific research capacity in low-income countries. *PLoS Biol*, 7(7), e1000156.

Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., … Quille, K. (2019a). An international study piloting the measuring teacher enacted computing curriculum (METRECC) instrument. In *Annual Conference on Innovation and Technology in Computer Science Education*, ITiCSE (pp. 111–142). https://doi.org/10.1145/3344429.3372505

Falkner, K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., … Quille, K. (2019b). An international comparison of K-12 computer science education intended and enacted curricula. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1–10). New York, NY, USA: ACM. https://doi.org/10.1145/3364510.3364517

Government of Pakistan. (2018). *Ministry of Information & Broadcasting Government of Pakistan*. Retrieved from http://moib.gov.pk/ on February 20, 2020.

Hubwieser, P., Armoni, M., & Giannakos, M. N. (2015). How to implement rigorous computer science education in K-12 schools? Some answers and many questions. *ACM Transactions on Computing Education*, 15(2), 1–12. https://doi.org/10.1145/2729983

Ministry of Education Sri Lanka. (2015). Moe.gov.lk. Retrieved from https://web.archive.org/web/20191126145533/http://www.moe.gov.lk/english/images/subject_related/ICT/ALsyllabusrevised.pdf on March 10, 2021

Gander, W., Petit, A., Berry, G., Demo, B., Vahrenhold, J., McGettrick, A., … Meyer, B. (2013). *Informatics Education: Europe Cannot Afford to Miss the Boat. ACM Europe: Informatics Education Report*. New York.

Hong, H., Wang, J., & Moghadam, S. H. (2016). K-12 Computer science education across the U.S. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 142–154). https://doi.org/10.1007/978-3-319-46747-4_12

Crick, T., & Moller, F. (2016). A national engagement model for developing computer science education in Wales. In *9th International Conference on Informatics in Schools (ISSEP)* (pp. 1–13). Munster, Germany.

Porter, A. C., & Smithson, J. L. (2001). *Defining, developing, and using curriculum indicators. CPRE Research Reports*.

Quille, K., Culligan, N., & Bergin, S. (2017). Insights on gender differences in CS1. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 263–268). New York, NY, USA: ACM. https://doi.org/10.1145/3059009.3059048

Quille. K & Bergin, S (2019) CS1: how will they do? How can we help? A decade of research and practice. *Computer Science Education*, 29:2-3, 254-282, DOI: 10.1080/08993408.2019.1612679

Quille, K. & Bergin, S. (2020). Promoting a growth mindset in CS1: Does one size fit all? A pilot study. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (ITiCSE '20). Association for Computing Machinery, New York, NY, USA, 12–18. DOI: https://doi.org/10.1145/3341525.3387361

Sysło, M. M., & Kwiatkowska, A. B. (2015). Introducing a new computer science curriculum for all school levels in Poland. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*) (pp. 141–154). https://doi.org/10.1007/978-3-319-25396-1_13

The Royal Society. (2012). *Shut down or restart? The way forward for computing in UK schools*. Technical report. London. Retrieved from https://royalsociety.org/~/media/education/computing-in-schools/2012-01-12-computing-in-schools.pdf

The Royal Society. (2017). *After the reboot: Computing education in UK schools*. Technical report. London. Retrieved from https://royalsociety.org/~/media/policy/projects/computing-education/computing-education-report.pdf

Vivian, R., Quille, K., McGill, M. M., Falkner, K., Sentance, S., Barksdale, S., … Maiorana, F. (2020). An international pilot study of K-12 teachers' computer science self-esteem. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (pp. 117–123). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3341525.3387418

Wikramanayake, G. N. (2005). Impact of digital technology on education. In *24th National Information Technology Conference* (pp. 82–91).

World Bank. (2020). *World Bank Country and Lending Groups*. Retrieved from https://datahelpdesk.worldbank.org/knowledgebase/articles/906519-world-bank-country-and-lending-groups, February 26, 2020

Raspberry Pi