Malmi, Lauri; Sheard, Judy; Sinclair, Jane; Kinnunen, Päivi; Simon

Domain-Specific Theories of Teaching Computing: Do they Inform Practice?

Published in:
Koli Calling '23: Proceedings of the 23rd Koli Calling International Conference on Computing Education Research

DOI:
10.1145/3631802.3631810

Published: 06/02/2024

Document Version
Publisher's PDF, also known as Version of record

Domain-Specific Theories of Teaching Computing: Do they Inform Practice?

Lauri Malmi
Aalto University
Finland
lauri.malmi@aalto.fi

Judy Sheard
Monash University
Australia
judy.sheard@monash.edu

Jane Sinclair
University of Warwick
United Kingdom
j.e.sinclair@warwick.ac.uk

Päivi Kinnunen
University of Helsinki
Finland
paivi.kinnunen@helsinki.fi

Simon
Unaffiliated
Australia
simon.unshod@gmail.com

ABSTRACT

Computing education research applies theories from the social sciences to build deep understanding of factors that influence students' learning process in different educational settings; but in recent years, computing education researchers have increasingly developed domain-specific theories and models that address the challenges and phenomena specific to computing education. Several literature surveys have addressed these developments. However, little attention has been given to whether these domain-specific theories and models have actually been applied to improve pedagogical practices in computing education.

In this paper, we explore domain-specific theories and models that are related to teaching computing. We present these constructs and report our findings about their impact on computing education pedagogies, based on our analysis of publications that cite the original papers presenting these theories or models. Based on the analysis of 1048 papers published between 2005 and 2022 in the International Computing Education Research Conference (ICER) and the journals Computer Science Education and ACM Transactions on Computing Education, we identified 31 papers that present relevant theoretical developments in these areas. We further analyze how these papers have been cited and discuss their identified pedagogical use cases in the citing papers. In general, our results show that while many papers refer to these theories and models, there are few concrete connections to presented pedagogical settings and few implications for further research. However, the papers themselves present interesting and relevant pedagogical ideas. We discuss these observations and make recommendations for reporting such connections and implications.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**.

Koli Calling '23, November 13–18, 2023, Koli, Finland

ACM ISBN 979-8-4007-1653-9/23/11.
https://doi.org/10.1145/3631802.3631810

KEYWORDS

computing education, theory, theoretical construct, research, method, pedagogy

ACM Reference Format:
Lauri Malmi, Judy Sheard, Jane Sinclair, Päivi Kinnunen, and Simon. 2023. Domain-Specific Theories of Teaching Computing: Do they Inform Practice?. In *23rd Koli Calling International Conference on Computing Education Research (Koli Calling '23), November 13–18, 2023, Koli, Finland*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3631802.3631810

1 INTRODUCTION

In recent years there has been substantial interest in the use of theory in computing education research (CER). Theories can guide research designs and support the interpretation of collected empirical data. Theories also have the potential to guide research-based pedagogical development work in many ways, from making sense of our everyday observations to guiding methodological choices [40, 71]. Several reviews have explored the state of the art as evidenced in the CER literature [30, 35, 66, 67]. In 2019 Computer Science Education published a special issue 'Advancing Theory about the Novice Programmer' [38], and in 2022-2023 ACM Transactions on Computing Education published two special issues 'Conceptualizing and Using Theory in Computing Education Research' [36, 72], which present reviews of specific theories or families of theories, along with novel theoretical approaches to address the teaching and learning of computing.

Most of the works mentioned above explore the use of theories from the social sciences, particularly education and psychology. There is, however, a growing body of literature that seeks to build domain-specific theories to support computing education in particular. Some examples include a model for students' programming process by Carter et al. [7], a theory of teaching programming by Xie et al. [82], and a theory of API knowledge by Thayer et al. [73]. Nelson and Ko promoted the need for such work in their position paper in ICER 2018 [41]. We have surveyed such developments in several reviews [32–34].

However, from the point of view of a computing educator, the value of theories may remain somewhat obscure. Educators are essentially interested in improving their teaching practice and supporting their students' learning. While theories can certainly give insights to help interpret observations on learning processes and learning outcomes, practitioners are more often looking for recommendations for selecting or tuning pedagogical practices. They

would especially value theories that give concise guidance in this regard. However, such theories are rare. In our recent review [34], we explored domain-specific theories in the broad area of students' learning process, underlying factors, and learning outcomes. We identified 80 theoretical constructs, but found only three that we classed as fitting Gregor's label of 'design & action' theories [23]. This finding raises the question of whether domain-specific theories have done anything to improve pedagogical practice in computing education. We therefore set out to explore the impact of these theories in some depth, while acknowledging the challenge of this endeavour. Research that involves surveying or interviewing computing teachers might reveal some of the impact, but while teachers might articulate when and in what course they adopted the idea for a change in their teaching, it is unlikely that they would refer explicitly to a theoretical development in a paper that they have read or heard about from a colleague. This is especially true for domain-specific theories, which are unlikely to be discussed in any pedagogical courses that teachers may have taken, and are not a common topic of discussion among teachers. We therefore chose to consider published papers as our data source. If a paper cites another paper that proposes a domain-specific theory, we are more likely to find explicit connections between theory and practice in the authors' chain of argumentation. Our overall research question is: *What domain-specific theories focusing on teaching have been developed in CER and what evidence is there of their impact on computing education pedagogical practice, as reported in the literature?*

Our previous work [32–34] focused mainly on aspects of *students' learning process*, such as students' understanding of computing topics, errors/misconceptions therein, and their behaviour, performance, emotions, and attitudes related to studying. While theories in these areas might inspire teachers contemplating pedagogical designs, we now complement the work by addressing domain-specific theories for *teaching, pedagogical content knowledge, goal setting, and assessment*, and seeking evidence in the literature as to whether these have impacted on pedagogical practice.

2 RELATED WORK

The relationship between an academic discipline and relevant theory has been investigated from a number of different perspectives. Perhaps most obviously, significant attention has been paid to the benefits (and indeed the necessity) of theory as a solid basis for research. The use of theory has been identified as an indicator of a mature research area [39] and the development of new theories within a domain is seen as one of the characteristics of an independent discipline [17]. Research feeds back by developing and refining theory in a process that shapes the development of the discipline itself [14, 76]. Another significant dimension to the relationship is practice: how do theory and practical application in a discipline relate to each other? How, and to what extent, can each inform the other? This section presents some of the existing work that informs current understanding of these relationships in general, and describes research that maps the use of theory specifically in the CER domain.

2.1 Theory in Research and Pedagogical Development

The development and use of theory have long been seen as fundamental to the conduct of research in the sciences, but even in that context, the concept of a 'theory' is hard to define: as Suppes wrote in 1964, "If someone asks 'What is a scientific theory?' it seems to me there is no simple answer to be given" [64, p63]. Nevertheless, scientific theory is understood to be a mechanism that can capture causal relationships, allowing phenomena to be understood and predicted and providing the accepted bedrock for research [14]. The interaction between theory and research, characterised by continuing rounds of investigation and refinement, may be seen as a framework for understanding the evolution of the subject itself [14, 76].

For disciplines such as education, in which multiple epistemological and methodological approaches have been applied, there are varying understandings of the role of theory. For instance, a case has had to be made for the beneficial role that theory can play in underpinning qualitative research [14, 49, 50, 65]. Recent work still indicates the need to "build upon the case that a balanced and centered use of the theoretical framework can bolster the qualitative approach" [11, p1]. A central issue here is the nature of theory: where human behaviour is involved, to what extent is it possible to identify causal theories that can effectively explain and predict [55]? For such areas of investigation the explanatory and interpretive roles of theory may be particularly beneficial, allowing aspects of human behaviour to be modelled and investigated [44, 81]. While such models cannot be fully predictive, they can assist the exploration and understanding of complex phenomena and of the beliefs and motivations that underlie them [44, 65]. Theories can provide different 'lenses' to help researchers to frame their work and organise the analysis of data and interpretation of results [49]. Tedre and Pajunen [69] discuss in more detail the nature of theory in computing education.

Another interaction relevant for our work is that between theory and pedagogical development. In common with educational research in general, CER has at its heart the very practical aim of improving student learning. As expressed by Nelson and Ko, "A primary goal of computing education research is to discover designs that produce better learning of computing" [41, p21]. Educational scholarship is not solely for the purposes of furthering understanding in research; it should also support the development of evidence-based teaching practice that can meet educational goals such as improving outcomes. The motivation and starting point for development work is often the need to find a working solution for a practical challenge encountered in teaching, not the intrinsic interest of testing the accuracy of some learning theory or framework. However, once a practical problem has been identified, learning theories and field-specific theoretical constructs and frameworks have great potential to guide pedagogical development work in many ways [71]. Theories may provide us with powerful lenses to make sense of what is happening in challenging situations. They help us to organise our observations and see beyond the immediate mundane observations muddled with a pool of details, clarifying the needs for development. Theories may also suggest suitable research methods to guide the research-based development work, as methodological and theoretical perspectives are at least partially

aligned [40]. Finally, if we aim to contribute to our community's knowledge about teaching and learning beyond our immediate classrooms or institutions we need to be able to communicate what we have learned in a way that is transferable. In this endeavour, theorising about one's research findings on pedagogical innovations and their effects is a powerful tool. Chick [9, p57] summarises the role of theory in development work as follows *"... researchers can benefit from being aware of their philosophical approach and theoretical assumptions about learning because it will help them ask new questions, design better studies, and more strongly articulate their findings, especially to colleagues with different world views."*

In this paper we adopt our earlier approach of viewing a theory as "a broad class of concepts that aim to provide a structure for conceptual explanations or established practice, and use such terms as 'theories', 'models', and 'frameworks' to describe particular manifestations of the general concept of theory" [35, p29]. The definition makes reference not only to conceptual explanations but to a relationship with practice. This paper seeks both to map the use and development of theory in computing pedagogic research and to conduct an initial examination of the relationship between that theory and pedagogic practice.

2.2 The CER Domain

Research in computing education obviously shares common theory with educational research in general [66, 67], with many of the theories and frameworks used in education and in other social sciences providing useful structure in CER. The 'grand theories' of education are certainly applicable, and there is clear relevance in approaches such as the 'design studies' methodology examined by diSessa and Cobb ("iterative, situated, and theory-based attempts simultaneously to understand and improve educational processes" [14, p80]).

However, theory relevant to CER is not purely a subdomain of general education theory: it has considerable crossover with computing in general in, for example, the development of novel software and hardware artifacts [34]. Computing research itself draws on theories and traditions of a number of other disciplines such as mathematics and engineering [70]. The development of artifacts and pedagogic frameworks also aligns strongly with research in design science, hence linking it to theory in that field [76]. Some topics are more specifically domain-based. For example, pedagogy for teaching programming, a central theme in computing education, requires a deep understanding of programming as well as knowledge of more general educational theories such as learning and cognitive load [34].

CER is a relatively new area of research that grew out of a combination of teaching practice and general pedagogic theory, with early papers often reporting teaching tools and initiatives [18]. By 2014, Malmi et al. were able to report that while CER was drawing extensively on theory from other disciplines, CER-specific theories were much less frequently observed [35]. Eight years later, evidence from our extensive review of CER publications indicated that a wide range of new theoretical constructs were emerging, but there was little refinement of or building upon that theory [34]. According to models of discipline development, as an emerging area grows and establishes its own identity, new domain-specific theory is generated [17, 39] and there is a systematic building of theory based on previous work [17]. It appears that CER is currently developing in the first aspect but is yet to fully establish the second.

2.3 Mapping the Use of Theoretical Constructs within CER

A number of papers have noted the increasing use of theory within CER [32–34, 41]. Based on our comprehensive review of papers from three major CER publication venues from 2005 to 2015, we categorised emerging theories into the following 11 subject areas [32]: 1) learning/understanding, 2) emotions/attitudes/beliefs/self-efficacy, 3) study choice/orientation, 4) performance/progression/retention, 5) learning behaviour/strategies, 6) perceptions of computer science/computing, 7) teaching/pedagogical content knowledge, 8) assessment/self-assessment, 9) content/curriculum/learning goals, 10) errors/misconceptions, and 11) computing education research.

We defined a *theoretical construct (TC)* as "a theory, model, framework, or instrument developed through application of some rigorous empirical or theoretical approach" [32, p188], and we examined citations to TCs in the areas of learning/understanding and learning behaviour/strategies, finding that domain-specific TCs are increasingly emerging, but 90% of papers that cite them do not actually use the TC. In our further work, we reviewed the development and use of theory in the areas of emotions/attitudes/beliefs/self-efficacy [33] and student learning, studying and progression [34]. Again, there were many good examples of new CER-specific TCs, but there was very little evidence of studies being replicated to provide stronger evidence for a suggested TC.

3 RESEARCH QUESTIONS AND METHOD

Our goal is to explore domain-specific theories and models in computing education that have had an impact on pedagogical practice. We build on our previous work [32–34], because the broad definition of *theoretical construct* (TC) fits very well with our current goals. We include first under this definition various statistical models explaining the relationships between identified concepts. However, we exclude simple hypothesis testing (which would cover too much literature to be practical for our purpose). Second, we include qualitative data-driven categorisations that seek to build higher-level abstract descriptions of the data, e.g., grounded theories, taxonomies or typologies, but not simple sets of categories with no further structure. Third, we include models presented as diagrams or equations with accompanying explanations and arguments. Fourth, we include validated instruments for measuring particular theory-based concepts, as instruments are often used as tools for building new theoretical constructs, and they can also be used in assessment, as we shall see below. Finally, we note that if a construct (in most cases an instrument) was initially developed in another discipline and adapted to computing education context, it was included as a domain-specific TC.

Our previous work focused on literature that revolves around students, their conceptions, behaviour, and factors affecting them. In contrast, we now focus on theories and models related to *teaching*. Using our above-mentioned categorisation of CER literature (Section 2.3), we are especially interested in literature focusing on *teaching/pedagogical content knowledge*, *content/curriculum/learning*

goals, and *assessment/self-assessment*. We are not aware of any detailed analysis of domain-specific theoretical developments in these areas. Our specific research questions are:

RQ1 What domain-specific theoretical constructs have been developed in the CER literature to address teaching, pedagogical content knowledge, goal setting, curriculum design, and assessment?
RQ2 How were these theoretical constructs developed?
RQ3 What are the main goals of these theoretical constructs?
RQ4 To what degree can we find evidence that these theoretical constructs have informed pedagogical practice?

3.1 Data

Our data pool covers all full papers from the ICER conference and two journals, Computer Science Education (CSE) and ACM Transactions on Computing Education (TOCE), from 2005 to the end of 2022. These venues were selected as they all publish long research papers, where theoretical developments are likely to be presented. Moreover, during our previous work [32, 34], we had already analysed the papers in these three venues till the end of 2020. We complemented the data with papers published in 2021-2022. The total pool comprises 1048 papers: 385 from ICER, 281 from CSE, and 382 from TOCE.

3.2 Method

From our previous work, we have taken the published lists of theoretical constructs within our areas of interest from 2005 to 2015 [32, Table 2] and 2005 to 2020 [34, Appendixes A and B (Area of focus: assessment/self-assessment)], and used our previous analysis of these TCs to partially address our RQ1, RQ2, and RQ3. We augment the results with an analysis of more recent literature, and complement the previous work by analysing TCs that were not covered in the published papers.

For the more recent literature, published in 2021-2022, two people read through all papers and identified candidate papers that appeared to present some new theoretical development. These papers were individually discussed until consensus was reached. We excluded pedagogical models and frameworks and design principles for learning resources, unless they were based on rigorous empirical evaluation to validate the model or framework, such as in Xie et al. [82]. An example of an excluded TC is the Block model [56] because in the paper introducing it the empirical study included only 10 students, split between treatment and control groups.

The final agreed set of constructs was then listed for the subsequent analysis. For each of these new constructs, two researchers discussed their focus area and the research methods by which they were developed until consensus was reached. The same technique was used in the categorisation process for addressing the goals of TCs (RQ3) and their use cases (RQ4). We use the term *source papers* for papers that introduce a TC.

For RQ3, we used, as in our earlier work [34], the taxonomy developed by Gregor [23] to classify the nature of theories. While this taxonomy was developed for information systems research, its five categories appear well suited to classifying theories and models in CER. *Analysis theories* focus on 'what is': analysis and description of a phenomenon with no purpose of identifying causal relationships or making predictions. *Explanation theories* focus on 'what is, how, why, when, and where', thus explaining some aspects of a phenomenon. However, they do not aim at predicting anything. *Prediction theories* state 'what is and what will be'. They offer some form of predictions of future findings, but with no clear explanation of why this happens. Gregor agrees that these cases are rare; they describe observed regularities with no clear explanation (yet). *Explanation & prediction theories* combine prediction with explaining what is happening. *Design & action theories* state or guide how something should be carried out.

To address RQ4, for each source paper in our list that presents a TC, we collected details of up to 30 most recent publications (as at the end of 2022) listed as its *citing papers* in Google Scholar (GS). If our previous work had included such citation analysis for a TC, we included it in our current results. When searching for citing papers, we included theses, and excluded short abstracts, papers that we could not access due to paywalls, papers in languages other than English, and duplicate publications. We continued the analysis back in time until at least 30 citing papers had been analysed, if there were that many. For each citing paper, one researcher scanned the full paper searching for citations of the corresponding source paper. In all identified cases, the citing text was read to identify whether the paper was cited because of the TC, or possibly for some other reason, such as its conclusions. When a TC was clearly addressed, the use case was categorised using our existing scheme [33] (see Table 1). This scheme categorises the type of TC usage according to one of 13 descriptors organised into four groups. Of particular pertinence is the category (A6) of a TC 'used to design a new pedagogical method'. However, we shall also report other use cases where appropriate. For each identified case, another researcher read the same paper and the case was discussed to reach consensus on the categorisation or to omit the use case if it was too vague.

Each citing paper with a use case of A6 was read more closely to identify how the TC had informed the reported pedagogical design or whether it was discussed in the context of 'pedagogical implications' or some similar part of the paper. In most source papers, the developed TC was validated in some particular pedagogical setting and context. We do not report these validations, but focus instead on pedagogical implications for other contexts that we found in the source papers by carefully reading their discussion and conclusion sections.

4 RESULTS

We present our results of analysis of the literature to address our research questions.

4.1 Domain-Specific TCs in our Areas of Focus (RQ1)

Our analysis of the literature in the three venues of CSE, ICER and TOCE from 2005 to 2022 found 31 source papers reporting the development of TCs in the areas of teaching/pedagogical content knowledge, content/curriculum/learning goals, and assessment/self-assessment. One paper [6] in the area of teaching/pedagogical content knowledge reported two outcome spaces for their phenomenographical study of conceptions of successful and unsuccessful teaching. Another paper in the same area [63] included two models, one for describing the involvement of external stakeholders in a project

Table 1: Categories of use of TCs: D – description; A – application; C – construction; V – validation (Malmi et. al [33])

Code	Description
D	Cited in related work, possibly described and critiqued but not used in the work
A1	Used as a framework to scope a study
A2	Used to develop a data collection instrument
A3	Used as a framework for data analysis
A4	Used to predict results of a study
A5	Used to interpret/compare/explain results of a study
A6	Used to design a new pedagogical method
A7	Used as an instrument in a study
C1	Modified and/or extended existing construct for use in a new context
C2	Developed new construct from existing construct and empirical work
C3	Developed new construct from existing construct and argumentation
V1	Used in a test to improve or discount an existing/new construct
V2	Tested the construct in a new context

course and another one for designing such an involvement. Finally, one paper [4] in the area of assessment/self-assessment reported the development of two instruments. In total we found 34 TCs in the 31 source papers.

The numbers of papers for each area of focus reported in each venue are shown in Table 2. The total number of TCs is fairly evenly distributed across the venues, but less so within each focus area.

There were 14 papers reporting the development of a TC in the teaching/pedagogical content knowledge area, five papers in the content/curriculum/learning goals area, and 12 papers in assessment/self-assessment. In Appendixes A and B we list the TCs, giving a brief description of each, its method of development, and the number of citations for its source paper. Twelve papers reporting the development of instruments are included in a separate table.

Table 3 shows the numbers of papers found in each three-year period from 2005 to 2022 for each area of focus. More than half (58%) of the TCs were developed in the past three years (2020–2022), indicating a recent high increase in interest in the development of TCs in the area of teaching and assessment. Finally, we note that we did not find any TCs that fell outside the focus areas initially identified in our earlier work [32].

4.2 How Were the TCs Developed? (RQ2)

When investigating the source papers we identified a variety of approaches to developing the TCs. For 32% of the TCs, the development involved the use of an existing theory, model, or instrument. Table 4 lists the source papers for each focus area where such a construct was used in developing the new construct.

In most cases the development of the TCs involved a combination of methods. Almost half the cases (45%) used literature analysis or argumentation, often in combination. Quantitative methods were used in 55% of the cases, with factor analysis and regression the most used techniques. Qualitative methods were used in 18% of the cases, with approaches such as the Delphi method, phenomenography, grounded theory, and thematic analysis. The methods used to develop each TC are shown in Appendixes A and B.

4.3 Main Goals of the TCs (RQ3)

The counts of TCs that match each purpose in Gregor's taxonomy [23] are shown in Table 5. It is not surprising that most TCs in the area of assessment/self-assessment are instruments and therefore out of scope of the taxonomy. In addition, the most common type of TC in the area of teaching/pedagogical content knowledge is design & action, as we would expect. We did not find any TCs of type prediction, which is understandable as these cases are rare [23].

4.4 Evidence that the TCs Have Informed Pedagogical Practice (RQ4)

We found 479 papers that cited the 31 source papers that introduced the theory or model. Analysing these citing papers according to the 'type of use' classification scheme (Table 1), we found that a huge majority of citing papers just mentioned or described the TC. In only 11% of the papers was the TC applied in some form, and in very few cases was it used in construction (1%) or validation (1%). Table 6 summarises the levels of use found for each area of focus. Note that some citing papers have several use cases of the same type, which does not show in the table.

Of the 31 source papers, only 16 (Table 7) had cases where the TC was used beyond mere description. In looking for evidence of a TC informing pedagogical practice we carefully examined the citing papers categorised as A6, 'TC used to design a new pedagogical method'. Somewhat surprisingly, we found only six TCs that had been applied in actual pedagogical design or discussion of pedagogical implications. However, when counting the total number of different use cases, the A6 cases were most frequent. For the six source papers, we identified 19 citing papers where the TC was used to inform pedagogy. The second most common use case was A5, 'TC used to interpret/compare/explain results' (13 citing papers for seven different source papers). Other frequent use cases were A2 'TC used to develop an instrument' and A7 'TC used as instrument'. BDSI [47] was clearly the most used instrument.

In the following, we present a detailed discussion of the pedagogical use cases, splitting our findings into three parts. First, we report on the cases (A6) where the impact was clearly visible in the citing papers. However, as many of the source papers are very recently published and there has been little time for evidence of their impact, we also present TCs where the pedagogical implications are discussed in the source paper itself in some form. Finally,

Table 2: Numbers of source papers reporting the development of a TC in one of the three areas of focus in each venue

Focus Area	CSEd	ICER	TOCE	Total
teaching/pedagogical content knowledge	4	4	6	14
content/curriculum/learning goals	0	1	4	5
assessment/self-assessment	5	5	2	12
Total	9	10	12	31

Table 3: Numbers of source papers reporting the development of a TC in one of the three areas of focus in each three-year period from 2005 to 2022

Year	teaching/pedagogical content knowledge	content/curriculum/ learning goals	assessment/ self-assessment	Total
2005-2007	1	0	0	1
2008-2010	1	1	0	2
2011-2013	0	1	2	3
2014-2016	0	0	1	1
2017-2019	3	0	3	6
2020-2022	9	3	6	18
Total	14	5	12	31

Table 4: Source papers reporting the development of a theoretical construct based on an existing theory, model, or instrument

Theoretical construct involved in development	teaching/pedagogical content knowledge	content/curriculum/learning goals	assessment/ self-assessment
Theory (apply, combine, extend, modify, use)	[2, 13, 15]		[1, 37, 43, 82, 84]
Model (extend)		[80]	[52]
Instrument (adapt, extend, validate)	[29]		

Table 5: Purpose of the theoretical constructs in each of the three areas of focus; note that there are 37 TCs listed as three source papers each contained two TCs, and in three cases the TC matched two categories; note also that as there were no cases found for prediction, this has not been shown in the table. N/A cases denote TCs that are instruments.

Area of Focus	Analysis	Explanation	Explanation & prediction	Design & action	N/A
assessment/self-assessment	1	0	2	0	10
content/curriculum/learning goals	3	0	1	1	1
teaching/pedag. content knowledge	3	3	2	7	3
Total	**7**	**3**	**5**	**8**	**14**

we briefly list the TCs for which we found no clear pedagogical implications.

4.4.1 *Pedagogical impact reported in citing papers.*

Teaching/pedagogical content knowledge. Xie et al's. work, *A theory of instruction for introductory programming skills* [82], is a very good example of developing a domain-specific theory in CER, thus providing an extensive analysis of previous research and theoretical approaches to explain the development of programming skills. The authors construct a theory of instruction for four programming skills across two dimensions: reading/writing and semantics/templates. Reading semantics denotes that when given code, a student can determine the intermediate and final states of the code. Writing semantics means that when given an unambiguous description, the student can translate it to code. Reading templates involves recognising structures/patterns in the code and using that information to conclude what the code does. Finally, writing templates corresponds to the case that when given a problem description, the student is able to construct a plan that uses some structure/pattern to solve the problem. The core idea of the theory is to give a clear structure for learning: reading before writing and semantics before templates. The authors present an extensive discussion of how to instruct each of these skills, followed by an empirical study providing evidence that students learn better when instructed using these guidelines. The source paper for this work is the most cited work among the papers that we found to be presenting TCs.

Table 6: Counts of the levels of use of TCs in the citing papers for the three areas of focus; note that some citing papers use a TC in more than one way. Description is not counted as using a TC.

Area of Focus	Description	Analysis	Construction	Validation
assessment/self-assessment	169	20	4	4
content/curriculum/learning goals	73	13	0	0
teaching/pedagogical content knowledge	174	21	1	0
Total	**416**	**54**	**5**	**4**

Table 7: Source papers in each focus area that are cited for each level of use; note that there may be more that one citing for each paper listed, and that levels A4 & C1 are not shown in the table as there were no cases for these.

Use of Construct	teaching/pedagogical content knowledge	content/curriculum/ learning goals	assessment/ self-assessment
A1 – framework to scope study			[47]
A2 – develop an instrument	[6]	[73, 80]	[13, 62]
A3 – data analysis framework		[12, 22]	[47]
A5 – interpret/compare/explain results	[6, 63, 74, 82, 83]	[12, 22]	
A6 – design a new pedagogical method	[1, 52, 63, 82]	[12]	[47]
A7 – use as an instrument		[22]	[47]
C2 – new construct from empirical			[4]
C3 – new construct from argumentation	[1]		[4, 13, 62]
V1 – improve/discount existing theory			[13, 68]
V2 – tested construct in new context			[47]

Fowler et al. [20], inspired by Xie's work, propose interesting pedagogical opportunities where each skill can be supported with appropriate instruction, practice, and feedback for students. This would support the building of different suites of exercises testing different skill orderings. Note that while Xie's theory states that students should learn skills in a certain order, this does not mean that students would first learn only reading semantics, followed by learning writing semantics, etc. The skills should be exercised repeatedly with new content, allowing possibilities for different types of exercise suite, as Fowler et al. suggest. On the other hand, Margulieux et al. [37] cite Xie's theory when considering gaps in learning. They note that formative assessments can be used to identify gaps in discrete programming skills. While this may at first glance seem obvious, it is worth noting that formative assessments, whether multiple choice quizzes or exercises, are often not designed to match the progress of skills, as guided in Xie's theory, but rather to match the general course content progression.

Finnie-Ansley et al. explored how students categorise simple algorithmic problem statements in a card sorting exercise [19]. They were interested in the arguments used by students when categorising the statements; where experts recognise certain patterns in such statements, novices often recognise something else, possibly only surface-level observations. When discussing pedagogical implications, they note that the explicit teaching of patterns has been reported to improve learning results in different settings. They note that this fits well with Xie's theory where learning to read and write templates are central steps in learning programming skills. A similar argument is given by Weinman et al. [78], who studied teaching and learning programming patterns with faded Parsons problems.

PRIMM (Predict, Run, Investigate, Modify, Make) is a structured pedagogical approach to teaching programming [59]. It builds on the *abstraction transition taxonomy, ATT* [12], which separates programming discussion into three levels: English, CS speak, and code (ATT is described in more detail below). When explaining the theoretical background of PRIMM, the authors describe "ATT as a discourse intensive teaching model of student understanding of programs, ... A clear recommendation of this research, which drew on situated cognition, was to support learners to be able to transition across all levels." [p140]. When explaining the practice of PRIMM, they continue: "In PRIMM, students transition from the program or code level to the execution level and may also summarise in English to the problem or with CS speak to the algorithm. During the run stage they check to see if their prediction was correct using English, CS speak and code as they accommodate or assimilate their understanding with language and vocabulary becoming the oil to facilitate the transitions." [p148]. Sentance et al. [58] note that "discussion of the question should ideally take place in pairs or groups to enable students to develop the vocabulary they need to talk about the program." [p478]. Another paper applying PRIMM pedagogy also builds on ATT, noting that the goal is to "improve students' programming vocabulary and facilitate CS speak" [46, Section 3.2].

Rich et al. have carried out substantial work in defining *learning trajectories* to support learning computational thinking, modelling the gradual development of K-8 students' understanding of variables [52], sequence, repetition, conditionals [54], debugging [53], and decomposition [51]. Gane et al. [21] used these to build a set of computational thinking assessments for this age group, and Luo et al. [31] report how they designed the 'action fractions' lessons,

integrating mathematics and computational thinking, based on the same learning trajectory works.

Many capstone projects integrate external collaborators with teaching, which is beneficial for students' learning. However, this integration is not always successful, which may lead to frustration among the different stakeholders. This challenge is explored by Steghöfer et al. [63], who present *a model of how external stakeholders can be involved in course design.* This model has been used to identify common risks, aims, and mitigation strategies; through discussion with course teachers, they have built another, conceptual, model of external stakeholder involvement that can guide course design. Thereafter they discuss experiences of applying the model in several different project courses mainly in software engineering. Steghöfer's models are used by Hashim et al. [25], who report on developing a first-year engineering design course. They note that engaging the stakeholders is a key element in the course, and give a detailed description of the objectives of stakeholder consultations based on this model. In another example, Chiang [8] discusses the pedagogical implementation of an experiential learning project, emphasising how important it is to consider these challenges carefully: "Therefore, guidelines that can be used to guide the action of lecturers (internal stakeholders) and external stakeholders in the entire process of collaboration are included in the developed pedagogical model" [p48].

Ahmad et al. [1] build a *gamification framework*, based on literature analysis, which integrates different types of goal (goal orientation, achievements, reinforcements, competition, and fun orientation) and various game elements. The framework can be used in designing gamified environments for computer science courses. Ahmad's framework was explicitly applied by Ishaq et al. [28], who present a serious games design model for language learning in the cultural context, giving arguments for selecting the constructs for their own design: goal orientation, accomplishments, reinforcements, and fun orientation.

Content/curriculum/learning goals. Exploring introductory programming lecture materials, Cutts et al. [12] examined how students are asked to answer questions. They identified three different types of abstraction, English, CS speak, and code, and based on this they defined the *abstraction transition taxonomy, ATT*, which helps to interpret different types of task set for students. The taxonomy clarifies different levels of expression that students need to learn to be able to fluently discuss programming topics and to solve tasks. Especially relevant is understanding the challenge that students have when they need to work in two different abstraction levels. ATT has been used as a building block in some pedagogical models, as mentioned above.

Assessment/self-assessment. BDSI, Concept Inventory for Basic Data Structures [47] focuses on linked lists and binary trees. Webb et al. [77] present the instrument in depth, question by question, recommending how it should be used. Two papers report the use of BDSI for evaluating prerequisite skills for further computing studies in industry-collaboration programs [3, 24]. Taking another perspective, a 2020 ITiCSE working group [42] explored assessment questions in basic data structures and algorithms, seeking to identify the impact of prerequisite knowledge on students' performance in such a course. Poor performance may be due to shortcomings in prerequisite knowledge rather than poor learning of actual course topics; they therefore tried to build an assessment tool that could identify such shortcomings. As a validated instrument, BDSI was used extensively in the study. Its questions were analysed to determine whether they can diagnose difficulties with prerequisite skills; many questions are not suitable for this. Their contribution to mitigate this problem is defining patterns and principles for building *differentiated assessments*, advanced topic assessment questions that can also diagnose relevant prerequisite knowledge.

4.4.2 *Pedagogical implications of theoretical constructs discussed in the source papers.*

Teaching/pedagogical content knowledge. Margulieux et al. [37] point out that results concerning the impact of direct instruction versus constructivism are inconsistent. While the former method can produce better short-term results, the same does not hold for retention and transfer. Constructive approaches appear better in this sense, but this result is unreliable. The authors therefore propose the *multiple conceptions theory*, stating that "learners develop better conceptual knowledge when they are guided to compare multiple conceptions of a concept during instruction" [p184]. They discuss several instructional techniques that guide students to compare multiple conceptions. These include, as direct instruction methods, test-enhanced learning, erroneous examples, analogical reasoning, and refutation texts; and as constructivist techniques, productive failure, ambitious pedagogy, problem-based learning, and inquiry learning. The authors compare how these techniques support comparison of different conceptions (correct, incomplete, incorrect and misconception) and several concept-building mechanisms (vicarious failure, self-explanation, inductive or case-based reasoning, conceptual growth, and conceptual change). They further present concrete guidance on how these mechanisms can be supported in instruction, and discuss how their theory can explain success in two well-known pedagogies, peer instruction and Parsons problems.

Clarke et al. [10] also introduce an approach that combines different learning and engagement strategies (collaborative learning, gamification, problem-based learning, and social interaction), seeking to integrate them in classroom and online learning. However, they take the bold view that the combination could be presented and evaluated in a quantified way, using a mathematical model. The model gives weights, which should be empirically evaluated, by which different strategies can be combined to improve student learning and engagement. The model was tested in long-term research over many semesters in a software testing course, and resulted in improved student performance. Such an engineering approach to pedagogical design is very rare in literature.

A quite different educational context is analysed by Sharma et al. [60], who have developed a grounded theory presenting systemic misalignments in implementing undergraduate research experiences (UREs). While UREs are generally considered effective ways to motivate students for research work and graduate studies, their implementation involves numerous challenges. Students and senior researchers have different goals and expectations, which are likely to cause conflicts or frustration. The paper presents an in-depth analysis of the existing tensions, followed by concrete design implications for how these barriers could be overcome.

Content/Curriculum/Learning Goals. Thayer et al. [73] present *a theory of robust API knowledge*, seeking to identify core knowledge components related to APIs. These include domain concepts, execution facts, and usage patterns. Moreover, each of these classes is considered from the perspective of its role in understanding code, and what the design space of the component would include, that is, what concepts and potential API calls may be available with APIs and potential programs that could use an API. Together these three components form a coherent whole that supports understanding of APIs. The paper concludes with concrete implications for designing pertinent learning material.

A broader aspect in programming is discussed by Sharmin [61], who carried out a literature survey of creativity in computer science. Through the analysis the author builds a categorisation of creative components in CS1, which include collaboration, relevance, autonomy, ownership, learning by doing / iterative learning, and visual feedback. Each of these components is discussed from the perspective of relevant creativity/education theories, with numerous examples from literature for each component. This categorisation can be highly useful for teachers who seek to include creative activities in their courses.

Finally, Werner et al. [80] analysed the projects of middle school students who had developed 3D games with Alice. They analysed different features used in the games (objects, operations, interaction types, variables) to build a measure of game computational sophistication covering game complexity and richness. The framework can be used to support assessment frameworks in game design, as well as when discussing various game features with students.

4.4.3 Other theoretical constructs. Here we list TCs for which we found no clear pedagogical implications, either in the source paper or in any citing paper. Most of these are validated instruments and would appear to have potential to support pedagogical development [4, 6, 13, 15, 26, 27, 29, 43, 48, 62, 74, 83, 84]. See Appendix B for more information.

5 DISCUSSION

5.1 General Observations

We have explored a significant share of the computing education literature, covering 18 years of publications in three major venues. Our work complements and also contrasts our previous surveys of TCs [32, 34]. Our previous work focused on exploring student perspectives, that is, learning/understanding, errors/misconceptions, learning behaviour and strategies, assessment, learning results and performance, as well as factors relating to emotions/attitudes/beliefs and self-efficacy [33]. In the current research, we have explored literature from the teacher's perspective, including assessment. Naturally, these areas have much overlap, as teaching practices influence students' behaviour and learning results, and student feedback, along with teachers' observations of students' work and learning outcomes, encourages reflections and improvements in future teaching. It is beyond the scope of this paper to compile a holistic compilation of TCs from both of these perspectives. We leave that for future research, and here only summarise some contrasting findings.

In our data pool we found 31 papers presenting new constructs, three of them presenting more than one. Fifteen of these papers were published in 2021–2022, which strongly highlights the fast growing community interest in developing TCs. Our previous analysis identified 73 papers presenting TCs related to student learning during 2005–2020 (excluding assessment/self-assessment, which are counted in the current analysis). While these aspects were not in our focus, when searching for TCs we also found many cases related to the student perspective, and we recorded 15 of these (which are not discussed in this paper). The total counts highlight the prevalence of research on the student perspective in earlier work, while our current findings suggest an increasing balance in the research. We appreciate this finding, as the instructional process is a whole in which students and teachers are equally relevant.

Another significant difference between our current findings and our earlier results [32, 34] is that previously we found only three TCs of the type design & action, while now we have found eight such cases in a much smaller pool of TCs. Moreover, the previous analysis found 15 indications of TCs used to inform pedagogy (case A6), while we found 19. These numbers suggest that teaching-oriented TCs are more often used to inform pedagogy than student-oriented TCs. However, we do find it concerning that in both cases the numbers are low.

We have made some additional observations. Almost all TCs that we found relate solely to course-level teaching. This is not surprising, as pedagogy most often informs classroom practice and online learning. However, pedagogy can also inform wider educational practices at the program and curriculum levels. We hope to see more curriculum/program-level TCs in the future. Another observation is that even though there is a strong emphasis on student-centered pedagogical practices, we found no evidence of cases where students have been integrated into pedagogical development. Co-creating pedagogies with students seems a promising idea that deserves more attention.

5.2 Are TCs Informing Pedagogy?

Our findings show that at least in the literature, there is still limited evidence that domain-specific TCs are informing the development of pedagogy. While we found many interesting cases of new TCs emerging in the area, as we reported when answering RQ4, we found pedagogical use cases for only a fraction of the TCs we had identified. There are several possible reasons for this. Many CS teachers do not read, and perhaps do not even have access to, scientific publications in computing education; and even if they do, most of them do not write papers for computing education venues. We recognise a growing risk that a reader familiar only with science papers might find that CER papers introduce new challenges (e.g., understanding and accepting multiple epistemological stances, educational theories, and methodological choices and their consequences). Reading a long theoretically and methodologically rigorous paper takes effort, and if there are no clear practical implications discussed, there may be no message conveyed to the teacher. This does not mean that CER should retreat to publishing mainly 'practice papers'. Rather, we would encourage authors to include sections such as 'practical implications' or 'how this research informs pedagogy' in structured abstracts and within discussion or conclusion sections.

It appears likely that those with the greatest potential to apply domain-specific TCs and publish the results are themselves active researchers in CER. However, our observations suggest that

when writing papers for the three venues covered in this paper, the authors may be more focused on writing about their research results than on practical applications of their work. Such applications could easily be covered in discussion and conclusion sections, but we found limited evidence of this. It is possible, but seems implausible, that the practical applications of presented TCs are not considered relevant from their perspective. It is possible that more evidence of the applications can be found in shorter conference papers and in practice-oriented sessions and submission categories, but again, our citation analysis did not uncover much evidence of this, even though the search for citing papers was not limited to the three source venues, ICER, CSE, and TOCE.

5.3 Research, Theory, and Practice

Computing education researchers are themselves generally computing educators, so in one sense CER is certainly grounded in practice. However, wider classroom practice is obviously different from research publication, and knowledge developed through experience by practitioners in the field is far wider than can be investigated by a limited number of researchers. Several authors point to possible dangers of a narrow focus on theory as a driver for good pedagogy. For example, Nelson and Ko [41] discuss the tension between the depth of theory and the breadth of practice. Collins and Stockton [11] point out that theoretical results should never be used uncritically. This may lead us to consider what a healthy relationship between research, theory, and practice would look like. In their work on adult education, Usher and Bryant refer to the relationship between theory, research, and practice as a 'captive triangle' [75]. Their thesis is that theory and research have commonly been viewed as the base elements in the relationship, with practice built on as the apex; but this conception can limit our thinking (making us 'captive' to a particular view of how pedagogy should develop). They make the case for a different view of the triangle in which no single element is privileged. This is especially important in computing education, which emphasises, far more than other disciplines, educational domain-specific tools – which can support new types of pedagogical actions that are not clearly informed by theory.

Currently it appears that although the importance of theory in CER is increasingly recognised, the evidence for the influence of theory on practice is weaker. Overall, work mapping the use and reuse of theory in CER [32–35] paints a picture of TCs being increasingly developed but less often built on, tested by further practical trials, or applied to influence wider teaching practice within the community. This resonates with the situation described by Disessa and Cobb in their consideration of educational design studies: "Theories concerning educational matters seem to replace one another, rather than subsume, extend, or complement other theories. Although the state of the art constantly changes, it is often difficult to tell that progress is being made" [14, p79]. The picture emerges of a sort of primordial soup of CER theories, many of which will not be widely useful, and indeed on further investigation may not even be correct. Allowing theories to be tested and retested by practice (or rather, ensuring that this happens) can certainly help to identify the species that will survive. However, this in itself is a reinforcement of the captive triangle. It is also necessary to recognise the importance of experiential knowledge as the starting point for theory and to find ways in which the triangle can be rebalanced.

5.4 Limitations

Our data pool in searching for TCs is limited to papers in three highly research-oriented venues. CER is being published in many other conferences, sometimes with special tracks or sessions focusing on teaching practice. It was beyond our resources to extend our analysis to papers in SIGCSE, ITiCSE, or Koli Calling, for example, or to carry out backward snowballing from the source papers. On the other hand, selecting the same venues as in our previous surveys [32, 34] allowed us to make some better comparisons.

The definition of TC that we used is broad and subject to interpretation. Using such a definition was, however, a practical decision, as there is no common consensus of the concepts of 'theory', 'model', or 'theoretical framework'. Moreover, as the field has few, if any, rigid theories, taking a strict stance would have given us a very thin recall. We note that Tedre and Pajunen [69] also promote the advantage of developing models, which are more flexible and still highly useful, whereas theories can have additional load from the natural science tradition, such as a requirement of falsifiability. In practice, many of the TCs we have identified are models, but they are always models that are based on clear empirical research. We mitigated the challenge of identifying TCs by using two researchers, who independently read the papers and identified potential TCs which were jointly discussed until a consensus was reached. The same method was applied when categorising the TCs in various ways, as well as when categorising the use cases of TCs in the citing papers.

Our data pool for finding TCs was intentionally focused on research-oriented venues, whereas we had no such venue limitation for the citing papers. We thus found many use cases from more practice-oriented venues. When searching for pedagogical applications of the TCs, we decided to limit the scope of work by considering as our guideline about the 30 most recent citing papers for each source paper. Many source papers had fewer citations listed in Google Scholar, while some had considerably more. We acknowledge the missed opportunity to find pedagogical use cases of TCs in citing papers that we did not analyse. However, as there is clearly a growing interest in using theories and developing domain-specific theories, the most recent citing papers would presumably reflect this interest and better reveal whether and how TCs have been used. Our analysis of the literature covered the period from 2005 to 2022 and the Google citations were downloaded at the end of 2022. Understandably, we found very few citations for source papers from 2022, which comprise 29% of our dataset. It is likely that the future will bring evidence of use of these TCs.

Finally, we note that our research approach cannot capture pedagogical solutions in classrooms or other settings that are not reported in publications. As already mentioned in the introduction, such an endeavour would be quite different, and it remains unclear whether surveys or interviews of practising teachers would be any better at discovering TCs that are being used to inform pedagogy.

6 CONCLUSION

The goal of this work was to find teaching-related theoretical constructs developed in computing education and to investigate how

these have influenced pedagogy. Our search and analysis of the computing education literature from three prominent computing education research venues found a variety of theoretical constructs. Although some of the source papers reporting the developments were highly cited, we found few cases where the theoretical constructs had been used and fewer cases where they had directly influenced pedagogy. While this finding was disappointing and concerning, and could be seen as showing a lack of maturity of the computing education research field, it is encouraging that we found evidence of a recent increase in development of theory.

We challenge researchers, scholars, and teachers in computing education to seek out and use a theoretical basis for any pedagogical developments, thus building a solid theoretically-founded understanding of how we can enhance and improve teaching practice, ultimately enhancing the learning experiences of computing students.

We conclude with a few recommendations:

(1) An *Implications for teaching and learning* section should be added to papers and to guidelines for authors.
(2) When citing theoretical work, authors should explicitly explain how they use the theory.
(3) More replication and longitudinal studies are needed to confirm and/or develop further existing theoretical constructs in different educational contexts.
(4) More research is needed with theoretical development on program- and curriculum-level pedagogical practices.
(5) More research could be targeted to co-creating pedagogical practices with students.

REFERENCES

[1] Adnan Ahmad, Furkh Zeshan, Muhammad Salman Khan, Rutab Marriam, Amjad Ali, and Alia Samreen. 2020. The impact of gamification on learning outcomes of computer science majors. *Transactions on Computing Education (TOCE)* 20, 2 (2020), 1–25.

[2] Satu Alaoutinen. 2012. Evaluating the effect of learning style and student background on self-assessment accuracy. *Computer Science Education* 22, 2 (2012), 175–198.

[3] April Alvarez, Legand Burge, Shameeka Emanuel, Ann Gates, Sally Goldman, Jean Griffin, Harry Keeling, Mary Jo Madda, Bianca Okafor, Alycia Onowho, et al. 2020. Google tech exchange: an industry-academic partnership that prepares black and latinx undergraduates for high-tech careers. *Journal of Computing Sciences in Colleges* 35, 10 (2020), 46–52.

[4] Satabdi Basu, Daisy W Rutstein, Yuning Xu, Haiwen Wang, and Linda Shear. 2021. A principled approach to designing computational thinking concepts and practices assessments for upper elementary grades. *Computer Science Education* 31, 2 (2021), 169–198.

[5] Kevin Buffardi. 2020. Assessing individual contributions to software engineering projects with git logs and user stories. In *Proceedings of the 51st ACM technical symposium on computer science education*. 650–656.

[6] Angela Carbone, Linda Mannila, and Sue Fitzgerald. 2007. Computer science and IT teachers' conceptions of successful and unsuccessful teaching: a phenomenographic study. *Computer Science Education* 17, 4 (2007), 275–299. https://doi.org/10.1080/08993400701706586

[7] Adam S Carter, Christopher D Hundhausen, and Olusola Adesope. 2017. Blending measures of programming and social behavior into predictive models of student achievement in early computing courses. *ACM Transactions on Computing Education (TOCE)* 17, 3 (2017), 12.

[8] Wong Shaw Chiang. 2021. Involvement of External Stakeholders in Designing Pedagogy for Experiential Learning at University Level: A Case Study. *EDUCATUM Journal of Social Sciences* 7, 2 (2021), 45–56.

[9] Nancy L Chick. 2019. Theory and the scholarship of teaching and learning: Inquiry and practice with intention. In *The Grounded Instruction Librarian: Participating in the Scholarship of Teaching and Learning. Atlanta*, Mallon Melissa, Lauren Hays, Cara Bradley, Rhonda Huisman, and Jackie Belanger (Eds.). ALA, 55–64.

[10] Peter J Clarke, Debra L Davis, Ingrid A Buckley, Geoff Potvin, Mandayam Thirunarayanan, and Edward L Jones. 2021. Combining learning and engagement strategies in a software testing learning environment. *ACM Transactions on Computing Education (TOCE)* 22, 2 (2021), 1–25.

[11] Christopher S Collins and Carrie M Stockton. 2018. The central role of theory in qualitative research. *Int. journal of qualitative methods* 17, 1 (2018).

[12] Quintin Cutts, Sarah Esper, Marlena Fecho, Stephen R Foster, and Beth Simon. 2012. The abstraction transition taxonomy: developing desired learning outcomes through the lens of situated cognition. In *[Eighth] International Computing Education Research Conference (ICER 2012)*. ACM, 63–70. https://doi.org/10.1145/2361276.2361290

[13] Laura E De Ruiter and Marina U Bers. 2022. The coding stages assessment: development and validation of an instrument for assessing young children's proficiency in the ScratchJr programming language. *Computer Science Education* 32, 4 (2022), 388–417.

[14] Andrea A DiSessa and Paul Cobb. 2004. Ontological innovation and the role of theory in design experiments. *The Journal of the Learning Sciences* 13, 1 (2004), 77–103.

[15] Rodrigo Duran, Jan-Mikael Rybicki, Juha Sorva, and Arto Hellas. 2019. Exploring the value of student self-evaluation in introductory programming. In *15th International Computing Education Research Conference (ICER 2019)*. 121–130.

[16] Rodrigo Duran, Juha Sorva, and Otto Seppälä. 2021. Rules of program behavior. *ACM Transactions on Computing Education* 21, 4 (2021), 1–37.

[17] Peter F Fensham. 2004. *Defining an identity – the evolution of science education as a field of research*. Springer.

[18] Sally Fincher and Marian Petre. 2004. *Computer Science Education Research*. CRC Press.

[19] James Finnie-Ansley, Paul Denny, and Andrew Luxton-Reilly. 2021. A Semblance of Similarity: Student Categorisation of Simple Algorithmic Problem Statements. In *Proceedings of the 17th ACM Conference on International Computing Education Research*. 198–212.

[20] Max Fowler, David H Smith IV, Mohammed Hassan, Seth Poulsen, Matthew West, and Craig Zilles. 2022. Reevaluating the relationship between explaining, tracing, and writing skills in CS1 in a replication study. *Computer Science Education* 32, 3 (2022), 355–383.

[21] Brian D Gane, Maya Israel, Noor Elagha, Wei Yan, Feiya Luo, and James W Pellegrino. 2021. Design and validation of learning trajectory-based assessments for computational thinking in upper elementary grades. *Computer Science Education* 31, 2 (2021), 141–168.

[22] Ken Goldman, Paul Gross, Cinda Heeren, Geoffrey L Herman, Lisa Kaczmarczyk, Michael C Loui, and Craig Zilles. 2010. Setting the scope of concept inventories for introductory computing subjects. *ACM Transactions on Computing Education (TOCE)* 10, 2 (2010), 5.

[23] Shirley Gregor. 2006. The nature of theory in information systems. *MIS Quarterly* (2006), 611–642.

[24] Jean Griffin, Legand Burge, Sally Goldman, Diego Aguierre, Juan Alonso Cruz, April Alvarez, Albert Cervantes, Shameeka Emanuel, Ann Gates, Daniel Gillick, et al. 2022. Innovative Courses that Broaden Awareness of CS Careers and Prepare Students for Technical Interviews. *Journal of Computing Sciences in Colleges* 38, 5 (2022), 54–64.

[25] Hashim A Hashim, Catherine Tatarniuk, and Brad Harasymchuk. 2022. First Year Engineering Design: Course Design, Projects, Challenges, and Outcomes. In *2022 IEEE Frontiers in Education Conference (FIE)*. IEEE, 1–9.

[26] Sally AM Hogenboom, Felienne FJ Hermans, and Han LJ Van der Maas. 2022. Computerized adaptive assessment of understanding of programming concepts in primary school children. *Computer Science Education* 32, 4 (2022), 418–448.

[27] CD Hundhausen, Phillip T Conrad, AS Carter, and Olusola Adesope. 2022. Assessing individual contributions to software engineering projects: a replication study. *Computer Science Education* 32, 3 (2022), 335–354.

[28] Kashif Ishaq, Fadhilah Rosdi, Nor Azan Mat Zin, and Adnan Abid. 2022. Serious game design model for language learning in the cultural context. *Education and Information Technologies* 27, 7 (2022), 9317–9355.

[29] Wanda M Kunkle and Robert B Allen. 2016. The impact of different teaching approaches and languages on student learning of introductory programming concepts. *Transactions on Computing Education (TOCE)* 16, 1 (2016), 1–26.

[30] Alex Lishinski, Jon Good, Phil Sands, and Aman Yadav. 2016. Methodological rigor and theoretical foundations of CS education research. In *12th International Computing Education Research Conference (ICER 2016)*. ACM, 161–169. https://doi.org/10.1145/2960310.2960328

[31] Feiya Luo, Maya Israel, and Brian Gane. 2022. Elementary computational thinking instruction and assessment: A learning trajectory perspective. *ACM Transactions on Computing Education (TOCE)* 22, 2 (2022), 1–26.

[32] Lauri Malmi, Judy Sheard, Päivi Kinnunen, Simon, and Jane Sinclair. 2019. Computing education theories: what are they and how are they used?. In *15th International Computing Education Research Conference (ICER 2019)*. 187–197.

[33] Lauri Malmi, Judy Sheard, Päivi Kinnunen, Simon, and Jane Sinclair. 2020. Theories and models of emotions, attitudes, and self-efficacy in the context of programming education. In *16th International Computing Education Research Conference*.

36–47.
[34] Lauri Malmi, Judy Sheard, Päivi Kinnunen, Simon, and Jane Sinclair. 2022. Development and use of domain-specific learning theories, models, and instruments in computing education. *Transactions on Computing Education (TOCE)* 23, 1, Article 6 (dec 2022), 48 pages. https://doi.org/10.1145/3530221
[35] Lauri Malmi, Judy Sheard, Simon, Roman Bednarik, Juha Helminen, Päivi Kinnunen, Ari Korhonen, Niko Myller, Juha Sorva, and Ahmad Taherkhani. 2014. Theoretical underpinnings of computing education research: what is the evidence?. In *Tenth International Computing Education Research Conference (ICER 2014)*. ACM, 27–34. https://doi.org/10.1145/2632320.2632358
[36] Lauri Malmi and Josh Tenenberg. 2023. Editorial: Conceptualizing and Using Theory in Computing Education Research. *Transactions on Computing Education (TOCE)* 23, 1, Article 1 (Jan 2023), 4 pages. https://doi.org/10.1145/3570729
[37] Lauren Margulieux, Paul Denny, Kathryn Cunningham, Michael Deutsch, and Benjamin R Shapiro. 2021. When wrong is right: the instructional power of multiple conceptions. In *17th International Computing Education Research Conference (ICER 2021)*. 184–197.
[38] Lauren E. Margulieux and Briana B. Morrison. 2019. Guest Editorial. *Computer Science Education* 29, 2-3 (2019), 103–105.
[39] Tim May and Beth Perry. 2022. *Social research: Issues, methods and process*. McGraw-Hill Education (UK).
[40] Kathleen McKinney. 2006. Attitudinal and structural factors contributing to challenges in the work of the scholarship of teaching and learning. *New Directions for Institutional Research* 2006, 129 (2006), 37–50.
[41] Greg L Nelson and Amy J Ko. 2018. On use of theory in computing education research. In *Proceedings of 14th International Computing Education Research Conference (ICER)*. 31–39.
[42] Greg L Nelson, Filip Strömbäck, Ari Korhonen, Marjahan Begum, Ben Blamey, Karen H Jin, Violetta Lonati, Bonnie MacKellar, and Mattia Monga. 2020. Differentiated assessments for advanced courses that reveal issues with prerequisite skills: A Design Investigation. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*. 75–129.
[43] Lijun Ni, Tom McKlin, Han Hao, Jake Baskin, Jason Bohrer, and Yan Tian. 2021. Understanding professional identity of computer science teachers: design of the computer science teacher identity survey. In *17th International Computing Education Research Conference (ICER 2021)*. 281–293.
[44] Mogens Allan Niss. 2007. The concept and Role of Theory in Mathematics Education: Plenary presentation. In *Relating Practice and Research in Mathematics Education: Proceedings of NORMA 05, Fourth Nordic Conference on Mathematics Education*. TAPIR Akademisk Forlag, 97–110.
[45] Miranda C Parker, Leiny Garcia, Yvonne S Kao, Diana Franklin, Susan Krause, and Mark Warschauer. 2022. A pair of ACES: an analysis of isomorphic questions on an elementary computing assessment. In *18th International Computing Education Research Conference (ICER 2022)*. 2–14.
[46] Alex Parry. 2020. Investigating the relationship between programming and natural languages within the PRIMM framework. In *Proceedings of the 15th Workshop on Primary and Secondary Computing Education*. 1–10.
[47] Leo Porter, Daniel Zingaro, Soohyun Nam Liao, Cynthia Taylor, Kevin C Webb, Cynthia Lee, and Michael Clancy. 2019. BDSI: a validated concept inventory for basic data structures. In *15th International Computing Education Research Conference (ICER 2019)*. 111–119.
[48] Seth Poulsen, Geoffrey L Herman, Peter AH Peterson, Enis Golaszewski, Akshita Gorti, Linda Oliva, Travis Scheponik, and Alan T Sherman. 2021. Psychometric evaluation of the cybersecurity concept inventory. *ACM Transactions on Computing Education (TOCE)* 22, 1 (2021), 1–18.
[49] Scott Reeves, Mathieu Albert, Ayelet Kuper, and Brian David Hodges. 2008. Why use theories in qualitative research? *BMJ* 337 (2008).
[50] Research Council of Norway. 2011. *The role of theory in educational research*. Technical Report. Research Council of Norway.
[51] Kathryn M Rich, T Andrew Binkowski, Carla Strickland, and Diana Franklin. 2018. Decomposition: A K-8 computational thinking learning trajectory. In *Proceedings of the 2018 ACM conference on international computing education research*. 124–132.
[52] Kathryn M Rich, Diana Franklin, Carla Strickland, Andy Isaacs, and Donna Eatinger. 2022. A learning trajectory for variables based in computational thinking literature: using levels of thinking to develop instruction. *Computer Science Education* 32, 2 (2022), 213–234.
[53] Kathryn M Rich, Carla Strickland, T Andrew Binkowski, and Diana Franklin. 2019. A K-8 debugging learning trajectory derived from research literature. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 745–751.
[54] Kathryn M Rich, Carla Strickland, T Andrew Binkowski, Cheryl Moran, and Diana Franklin. 2017. K-8 learning trajectories derived from research literature: Sequence, repetition, conditionals. In *Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER)*. 182–190.
[55] Alex Rosenberg. 2006. *Philosophy of Science: A Contemporary Introduction*. Routledge.
[56] Carsten Schulte. 2008. Block Model: an educational model of program comprehension as a tool for a scholarly approach to teaching. In *Proceedings of the Fourth International Workshop on Computing Education Research*. 149–160.
[57] Sue Sentance and Jane Waite. 2021. Teachers' perspectives on talk in the programming classroom: language as a mediator. In *17th International Computing Education Research Conference (ICER 2021)*. 266–280.
[58] Sue Sentance, Jane Waite, and Maria Kallia. 2019. Teachers' experiences of using primm to teach programming in school. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. 476–482.
[59] Sue Sentance, Jane Waite, and Maria Kallia. 2019. Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education* 29, 2-3 (2019), 136–176.
[60] Rhea Sharma, Atira Nair, Ana Guo, Dustin Palea, and David T Lee. 2022. "It's usually not worth the effort unless you get really lucky": barriers to undergraduate research experiences from the perspective of computing faculty. In *Proceedings of 18th International Computing Education Research Conference (ICER)*. 149–163.
[61] Sadia Sharmin. 2021. Creativity in CS1: a literature review. *ACM Transactions on Computing Education (TOCE)* 22, 2 (2021), 1–26.
[62] Judy Sheard, Simon, Angela Carbone, Donald Chinn, Mikko-Jussi Laakso, Tony Clear, Michael de Raadt, Daryl D'Souza, James Harland, Raymond Lister, Anne Philpott, and Geoff Warburton. 2011. Exploring programming assessment instruments: a classification scheme for examination questions. In *Seventh International Computing Education Research Workshop (ICER 2011)*. ACM, 33–38.
[63] Jan-Philipp Steghöfer, Håkan Burden, Regina Hebig, Gul Calikli, Robert Feldt, Imed Hammouda, Jennifer Horkoff, Eric Knauss, and Grischa Liebel. 2018. Involving external stakeholders in project courses. *Transactions on Computing Education (TOCE)* 18, 2 (2018), 1–32.
[64] Patrick Suppes. 1964. What is a scientific theory? US Information Agency, Voice of America Forum.
[65] Patrick Suppes. 1974. The place of theory in educational research. *Educational Researcher* 3, 6 (1974), 3–10.
[66] Claudia Szabo, Nickolas Falkner, Andrew Petersen, Heather Bort, Kathryn Cunningham, Peter Donaldson, Arto Hellas, James Robinson, and Judy Sheard. 2019. Review and use of learning theories within computer science education research: primer for researchers and practitioners. In *ITiCSE 2019 Working Group Reports*. 89–109.
[67] Claudia Szabo and Judy Sheard. 2023. Learning theories use and relationships in computing education research. *Transactions on Computing Education (TOCE)* 23, 1, Article 5 (Mar 2023), 34 pages. https://doi.org/10.1145/3487056
[68] Narjes Tahaei and David C Noelle. 2018. Automated plagiarism detection for computer programming exercises based on patterns of resubmission. In *14th International Computing Education Research Conference (ICER 2018)*. 178–186.
[69] Matti Tedre and John Pajunen. 2022. Grand theories or design guidelines? Perspectives on the role of theory in computing education research. *ACM Transactions on Computing Education* 23, 1 (2022), 1–20.
[70] Matti Tedre and Erkki Sutinen. 2008. Three traditions of computing: what educators should know. *Computer Science Education* 18, 3 (2008), 153–170.
[71] Agnes Tellings. 2011. Theories and research in the field of education: An indissoluble union. In *The Research Council of Norway, The role of theory in educational research: Report from the March seminar*. 8–15.
[72] Josh Tenenberg and Lauri Malmi. 2022. Editorial: Conceptualizing and Using Theory in Computing Education Research. *Transactions on Computing Education (TOCE)* 22, 4, Article 38 (dec 2022), 8 pages. https://doi.org/10.1145/3542952
[73] Kyle Thayer, Sarah E Chasins, and Amy J Ko. 2021. A theory of robust API knowledge. *ACM Transactions on Computing Education* 21, 1 (2021), 1–32.
[74] Jodi Tutty, Judithe Sheard, and Chris Avram. 2008. Teaching in the current higher education environment: perceptions of IT academics. *Computer Science Education* 18, 3 (2008), 171–185. https://doi.org/10.1080/08993400802332423
[75] Robin Usher and Ian Bryant. 2014. *Adult education as theory, practice and research: The captive triangle*. Routledge.
[76] John Venable. 2006. The role of theory and theorising in design science research. In *First International Conference on Design Science in Information Systems and Technology (DESRIST)*. Citeseer, 1–18.
[77] Kevin C Webb, Daniel Zingaro, Soohyun Nam Liao, Cynthia Taylor, Cynthia Lee, Michael Clancy, and Leo Porter. 2021. Student performance on the BDSI for basic data structures. *Transactions on Computing Education (TOCE)* 22, 1, Article 8 (Mar 2021), 34 pages. https://doi.org/10.1145/3470654
[78] Nathaniel Weinman, Armando Fox, and Marti A Hearst. 2021. Improving instruction of programming patterns with faded parsons problems. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–4.
[79] Linda Werner, Jill Denner, and Shannon Campe. 2014. Children programming games: A strategy for measuring computational learning. *ACM Transactions on Computing Education (TOCE)* 14, 4 (2014), 1–22.
[80] Linda Werner, Jill Denner, Shannon Campe, and David M Torres. 2020. Computational sophistication of games programmed by children: a model for its measurement. *Transactions on Computing Education (TOCE)* 20, 2 (2020), 1–23.
[81] Timothy Williamson. 2017. Model-building in philosophy. *Philosophy's future: The problem of philosophical Progress* (2017), 159–171.

[82] Benjamin Xie, Dastyni Loksa, Greg L Nelson, Matthew J Davidson, Dongsheng Dong, Harrison Kwik, Alex Hui Tan, Leanne Hwa, Min Li, and Amy J Ko. 2019. A theory of instruction for introductory programming skills. *Computer Science Education* 29, 2-3 (2019), 205–253.

[83] Aman Yadav and Marc Berges. 2019. Computer science pedagogical content knowledge: characterizing teacher performance. *Transactions on Computing Education (TOCE)* 19, 3 (2019), 1–24.

[84] Ninger Zhou, Ha Nguyen, Christian Fischer, Debra Richardson, and Mark Warschauer. 2020. High school teachers' self-efficacy in teaching computer science. *Transactions on Computing Education (TOCE)* 20, 3 (2020), 1–18.

A Theoretical constructs, other than instruments, found in the study; ordered alphabetically by author name within area of focus, and showing number of citations in Google Scholar as at 31 December 2022

Source paper	Theoretical construct	Methods	Citations
Area of focus: assessment/self-assessment			
Hundhausen et al. [27]	Regression model (extending Buffardi's model [5]) of the effect of a team member's GitHub contributions to the project grade	regression	1
Sheard et al. [62]	Classification scheme to investigate characteristics of introductory programming exam questions	argumentation, empirical	48
Tahaei and Noelle [68]	Logistic regression model for detecting plagiarism in programming assessments based on patterns of resubmission	literature, argumentation, regression, empirical	22
Area of focus: content/curriculum/learning goals			
Cutts et al. [12]	Abstraction transition (AT) taxonomy to classify the knowledge and practices required to apprentice students into the programming community	argumentation, empirical	46
Goldman et al. [22]	Lists of expert-identified central concepts for programming fundamentals, discrete math, and logic design	delphi method	99
Sharmin [61]	Components of creativity-enhancing activities	literature, argumentation	5
Thayer et al. [73]	A theory of components of API knowledge	argumentation, regression	17
Werner et al. [80]	GCS 2.0 (game computational sophistication): measure of the relationship between different types of building blocks of computer games and game computational sophistication; new version of the GCS [79]	extended model, argumentation, empirical	8
Area of focus: teaching/pedagogical content knowledge			
Ahmad et al. [1]	A gamification framework from literature	extended theory, argumentation, empirical	42
Carbone et al. [6]	Two outcome spaces: IT academics conceptions of successful teaching (3 categories) and IT academics conceptions of unsuccessful teaching (5 categories)	phenomenography	44
Clarke et al. [10]	Model for maximizing the use of learning and engagement strategies	literature, argumentation, empirical	1
Duran et al. [16]	A framework for identifying, organizing, and communicating learning objectives that involve program semantics	literature, argumentation	6
Margulieux et al. [37]	Multiple conceptions theory - a framework which allows analysis of various generic pedagogies and explanation of their differences	used theory, literature, argumentation	9
Rich et al. [52]	Learning trajectory for variables	extended model, literature, argumentation	7
Sentance and Waite [57]	A model to frame understanding of how programming teachers use classroom talk to support the learning of programming	phenomenography, thematic analysis	4
Sharma et al. [60]	A model of systemic misalignments in implementing undergraduate research experiences	grounded theory	0
Steghöfer et al. [63]	Two models: A model for analysing involvement of external stakeholders in university courses and a model to design actions	action research	4
Tutty et al. [74]	Five categories of teaching experience and practice	phenomenography	25
Xie et al. [82]	A holistic theory for teaching programming by splitting it into several skills	used theory, literature, argumentation	87

B Instruments found in the study; ordered alphabetically by author name within area of focus, and showing number of citations in Google Scholar as at 31 December 2022

Source paper	Theoretical construct	Methods	Citations
Area of focus: assessment/self-assessment			
Alaoutinen [2]	Taxonomy-based scale for self-evaluation of programming knowledge	used theory, empirical	22
Basu et al. [4]	Two instruments to assess 4th-6th grade students' CT skills	literature, factor analysis, empirical	8
Duran et al. [15]	Instrument for self-evaluation of knowledge of programming concepts	adapted theory, exploratory & confirmatory factor analysis, empirical	12
Hogenboom et al. [26]	Computerized Adaptive Programming Concepts Test (CAPCT) to measure comprehension of basic sequences, loops, if statements, if-else statements, procedures, multiple agents, debugging, and generalization to a different programming syntax	empirical	5
Kunkle and Allen [29]	Instrument to measure understanding of fundamental and object-oriented programming concepts	modified instrument, exploratory factor analysis, empirical	85
Parker et al. [45]	New isomorphic versions of ACES assessment tool (Assessment of Computing for Elementary Students)	empirical	1
Porter et al. [47]	Basic Data Structures Inventory (BDSI): validated concept inventory for assessing knowledge of basic data structures concepts	qualitative, empirical, CTT (classical test theory), IRT (item response theory)	42
Poulsen et al. [48]	Cybersecurity concept inventory (validated in this paper)	delphi method, CTT, IRT	2
De Ruiter and Bers [13]	Instrument for assessing young children's proficiency in the programming language ScratchJr.	used & adapted theory, CTT, IRT	20
Area of focus: teaching/pedagogical content knowledge			
Ni et al. [43]	Instrument for assessing computer science teachers' professional identity	used theory, exploratory & confirmatory factor analysis	4
Yadav and Berges [83]	Instrument to measure teachers computer science pedagogical content knowledge (Rasch model used)	literature, content analysis, empirical	41
Zhou et al. [84]	Instrument to measure high school teachers' self-efficacy to teach computer science	used theory, confirmatory factor analysis,	17