The Roles and Challenges of Computing Terminology in Non-Computing Disciplines

Brett A. Becker

brett.becker@ucd.ie

University College Dublin & National Forum for the Enhancement of Teaching & Learning in Higher Education
Dublin, Ireland

ABSTRACT

Aspects of computing are used in all professions. Regardless of academic discipline, graduates require ever-improving computing competencies – applied in their own disciplinary contexts – to become proficient experts in their chosen careers, as well as informed members of society. However, teaching and learning contextually relevant computing competencies is challenging. One such challenge is the effective use of computing terminology. Terminology is fundamental to most human activities and is critical in making or breaking the interfaces where disciplines meet.

Notably, we found very little work on the use of computing terminology in non-computing contexts and none utilising interviews from educators in such areas. In this qualitative research, we interviewed 20 non-computing university-level lecturers from a very broad set of disciplines in Ireland. Interviews focused on the role of computing terminology in the teaching and learning of contextually relevant computing concepts in their own disciplines, including barriers presented and methods to overcome them.

This revealed a landscape as diverse as the disciplines themselves including a range of uses, issues, and approaches related to computing terminology. Although each story is unique, they all fit into a coherent spectrum, indicating that educators in one discipline would likely benefit from knowing how those in other disciplines have dealt with similar issues. They also reveal useful insights on contextual computing for computing educators. We also made several unexpected findings that are important considerations for how computing is taught and learned in other disciplines.

CCS CONCEPTS

• **Social and professional topics** → **Computing education; Computer science education.**

KEYWORDS

computing; computing across the disciplines; interdisciplinary; jargon; lexicon; nomenclature; non-majors; slang; terminology

ACM Reference Format:

Brett A. Becker. 2021. The Roles and Challenges of Computing Terminology in Non-Computing Disciplines. In *United Kingdom and Ireland Computing Education Research conference. (UKICER '21), September 2–3, 2021, Glasgow,*

This work is licensed under a Creative Commons Attribution International 4.0 License.

UKICER '21, September 2–3, 2021, Glasgow, United Kingdom

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8568-8/21/09.

<https://doi.org/10.1145/3481282.3481284>

United Kingdom. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3481282.3481284>

1 INTRODUCTION

The goal of most computing education research is improving how computing is taught and learned [11]. Computing competencies¹ are increasingly important in many disciplines. Although teaching computing to students in all disciplines has been discussed for decades [21] many practical realities had to come about for this to be realised to any meaningful extent. Today, inexpensive hardware and free/inexpensive software, reliable broadband, and the wide-ranging potential of technologies such as artificial intelligence, have resulted in the right atmosphere for computing to be well and truly applied in every discipline. Some subjects have arguably been revolutionized by computing in the last few decades, not just changing how they do what they do, but changing what they do.

We put forth that, for instance, the best lawyers of tomorrow will be those who have a command of computing complementary to the context of law. In other words, knowing a little computing, relevant to the practice of law, likely makes one a better lawyer. The same could be said for many, if not all, disciplines from art to zoology. In order to gain context-relevant computing competencies, terminology often poses a barrier. We do not advocate teaching only terminology but that terminology be specifically addressed, when introducing computing content.

As computing impacts so many disciplines it also affects how they are taught and learned. However, the teaching and learning of a discipline is highly context dependent. This is what makes disciplines distinct; without such specialization the world wouldn't have experts nor the need to train them. This context can make it difficult for competencies from other disciplines, including computing, to be incorporated into the teaching and learning of a given discipline. In fact, it is arguable that access to, and availability of, hardware and software are no longer the biggest barriers to the widespread use of computing in other disciplines. Rather, issues that are difficult to pin-down and hard to solve with raw economic or other brute-force approaches, present significant barriers. One such area is the terminology and vocabulary associated with a discipline such as computing. The vocabulary of computing was highlighted over 60 years ago as problematic [24] and still is today [6]. As the language educators use with students is exceedingly powerful and influential [43], it is important that the context of origin of a term is taken into account along with the context of use. When the context of origin and context of use differ, barriers can arise.

¹Inspired by CC2020 [6] we will use the term *competencies* to collectively refer to knowledge, skills, and dispositions required to be effective in a certain job, role, or task.

1.1 A Motivating Example

One way that computing is contributing to other fields is by providing models of discipline-specific contextual processes as information processes. In fact, some of the most innovative work is arguably being done by people who know both another discipline and computing well [9]. In many physical and social sciences, experiments are increasingly done *in silico* – using models of real-life systems rather than on the systems themselves.

Consider the life sciences – medicine and biology as two specific cases. These fields routinely employ computing for applications ranging from protein structure prediction to drug discovery/design to genetic sequencing, to diagnosing, treating, and preventing disease. Many of these practices extend beyond human health and involve veterinary sciences and zoology for instance. To be a biology student today is in many ways fundamentally different to being one one or two decades ago, due to the way that computing has been used – largely by biologists themselves – to reshape much of their discipline. In short, in order to be the best biologists they can be, today's biology students need to understand and apply several computing competencies in the context of the discipline of biology. Similar can be said for students of many other disciplines. However, educating biology students to become better biologists by learning to employ computing competencies is not easy, and terminology can complicate this already difficult scenario.

As a case in point, take the term *model*. We already used this term in describing *in silico* experiments. In a computing context – specifically in machine-learning – the term is often used in a way that can seem casual or vague to non-experts, yet machine-learning experts know what is meant when they hear it in context. However, finding a stand-alone definition for the term in this context is difficult. Many textbooks use it without defining it. The Amazon AWS Machine Learning Developer Guide defines it by way of an explanation: “The process of training a (machine learning) model involves providing an algorithm (that is, the learning algorithm) with training data to learn from. The term model refers to the model artifact that is created by the training process” [1].

In the context of biology the term *model* is also frequently used (in a non-computing sense). To the uninitiated this use often seems similar to its use in machine learning – it can seem vague and casual. Yet, biologists normally know what is meant in context when they hear the term being used. Again, finding a and definitive definition is difficult, and it is often used without definition. *Nature* defines this as: “an experimental system that recreates aspects of human tissue function or disease. For example, certain tumour cell lines may serve as cancer models, and transgenic mice that express human beta-amyloid protein may serve as animal models of Alzheimer's disease” [31].

In this case, the surface issue is that the same term is used differently in two different contexts. A deeper issue is that combining them by say, employing a machine learning model in the context of biology, could quickly become difficult to discuss particularly in a classroom. However, this is just one way that terminology can be problematic. Others include the way terminologies change, how they are translated, how they are used, and even, how they are abused. Although slang/jargon adds color and richness to language, it can also alienate and arouse emotion. Metaphor can be used for

describing, and teaching, but only if used in a way that has the intended effect. Although examples such as these may seem trivial in isolation, multiplied across many terms, many contexts and cultures, and many teachers and students, they can result in non-trivial educational barriers. As an example, in Section 2 we'll discuss a debate about the use of computing language started by Dijkstra decades ago, that is still alive and well today. First, following from our motivating example, we present potential barriers presented by terminology, followed by our research questions.

1.2 Barriers and Research Questions

From a language perspective, several *potential* barriers can arise due to the use of terminology when applying computing competencies in other disciplines including:

- (1) the same term is used in both disciplines, to mean different things [22, 40]
- (2) different terms are used in both disciplines, that mean the same thing [22, 40]
- (3) computer scientists themselves argue over the correct usage of terminology [17]
- (4) a computing term is not used in another discipline and/or the term's meaning in computing differs from how the word is used in everyday life, which is how those in other disciplines might interpret it [4, 35]
- (5) student level [8], native language [3], and other external factors may cause different barriers for different students
- (6) teacher computing knowledge / lack of training [38], and other external factors

However, examples of these being documented in the context of teaching computing competencies to non-computing students are extremely scarce. Further, these potential barriers are all permutations on the same idea - difficulty in students understanding the meaning of a term in relation to its context, for various reasons. Although they have been documented in the literature to some extent, it is possible that other barriers exist. Therefore, in this work we sought to answer the research questions by interviewing disciplinary experts:

- RQ1 *What computing terminology is used in non-computing disciplines, and in what ways are these used?*
- RQ2 *What terminology barriers are faced by educators in non-computing disciplines, when teaching computing competencies within their disciplines?*
- RQ3 *What can be done to overcome these terminology barriers, and are there approaches taken by one discipline that could be applied in another?*

In this paper we define a computing discipline as those under ISCED field 6 (Information and Communication Technologies) [15].

2 RELATED WORK

The role of terminology in computing is a vast topic with many facets and implications. In the present work we are specifically interested in the use of computing terminology in non-computing disciplines. In this section we situate this application in the wider context of computing terminology in general.

The terminology of computing has drawn increased interest in the last few years. In the last year, Mark Guzdial has written two *Communications of the ACM* Blog posts on the topic. The first [19] argues that Dijkstra, in “On the Cruelty of Really Teaching Computer Science” (which is contained in full in an article by Denning [10]) was wrong in warning educators away from using metaphor as an explanatory tool. As evidence that Dijkstra was wrong, Guzdial mentions (among other evidence) recent research demonstrating that embodied metaphor can be used to effectively communicate computing constructs and processes [28]. In his a more recent post, Guzdial addresses what students in liberal arts and sciences need to know about computing. However, the first task – one of the main subjects of the post – is establishing working definitions for what educators in that context do with computing and what their students need [20]. That is, in a specific context, discussing the application of computing in non-computing disciplines, has required honing the understood definition of computing, more or less from first principles, to provide a common ground from which to start, exemplifying the importance of terminology.

Terminology can also alienate newcomers as noted recently: “Teachers new to computing who are not familiar with technical jargon can feel like they have landed in a foreign world, making them reluctant to take on the subject, and potentially leading to misconceptions and misunderstandings in the classroom” [30, p1].

2.1 Challenges Posed by Terminology

Terminology has caused issues within computing education as a discipline itself. Given this, it is reasonable to expect that terminology may pose more varied issues when applied in other disciplines. The authors of Computing Curricula 2020 saw it necessary to include “a glossary of terms as well as a crosscutting nomenclature as used in different parts of the world”, complete with translations into several languages as an appendix [6, p23]. In fact, CC2020 directly states that the scope of terminology is a problem in computing: “Unfortunately, there is no universal terminology in computing or in computing education, even within the English-speaking world” [6, p70]. This says nothing of the surely greater difficulties imposed by computing terminology when brought to non-computing contexts.

Simon et al. conducted a working group at ITiCSE 2015 focusing only on the use of the following terms, limited to the scope of computing education: Computing; Information Technology; Informatics; Information and Communication Technology; Computer Science; and Information Systems [40]. A principal conclusion was that different interpretations of different terms – using multiple terms to mean the same thing, or the use of the same term to mean multiple things – can have negative consequences in matters as serious as obtaining employment, a point also made by Hasselbring [22]. The reader is directed to Simon et al. [40] for an in-depth background of computing terminology use in computing education.

Papamitsiou et al. [33] found that computing education authors use different terminology in paper keywords leading to inconsistency. In addition they found that in computing education research specifically, the use of generic terminology leads to a long-tailed distribution of keywords (which from this perspective is representative of the terminology used in computing education). Although focused on research specifically, that work brings into question how

the community uses terminology. The authors make several recommendations in terms of keyword choice for papers with many of the the parallels to terminology use in general being fairly obvious.

Computing terminology is also an issue in K-12 education with the The K-12 Computer Science Framework finding terminology and vocabulary to be problematic with ‘confusing’ being a common mention from framework reviewers, particularly where computing terms that have meanings different to those in other fields or everyday usage [8]. Some stated that the vocabulary used was too ‘technical’ while others voiced concern that it was not technical enough. The Australian curriculum uses a vocabulary list of nearly 300 terms that are defined in a glossary [42]. Sentance & Csizmadia surveyed hundreds of teachers as computing was being introduced in the UK second-level curriculum and found that “Teachers identified the need to encourage students to develop, embrace and articulate a common Computing vocabulary so that students are aware of technical keywords and use them correctly” [38, p483]. Others have used vocabulary intentionally as part of scaffolding when teaching computing to middle-school students [16].

Neil Selwyn has referred to the terminology of digital technologies as clumsy and awkward [37] and devoted a conversation in the preface to “Digital Technology and the Contemporary University: Degrees of Digitization” to explaining this [37]. This view applies to the terminology of computing and is likely exacerbated when these terms are used in non-computing contexts.

Perhaps the most related work to ours is by Fidoten & Spacco who surveyed computing and non-computing liberal arts faculty, asking questions regarding their perception of personality traits of computer science students, topics they think are covered in computer science classes, and their overall impressions of computer science. Of critical relation to this work, Fidoten & Spacco did not explore teaching computing concepts - their scope was more similar to Simon et al. [40] in that their goal was to situate computer science among other disciplines. One finding of interest to the present work was that different terms were used by those in computing faculties and those in other faculties, when asked “what does a computer scientist study?” [14], again highlighting that one’s own context has a large effect on how one sees, and interfaces with, other disciplines.

2.2 A Two-Way Street

It should be noted that not only has computing impacted many disciplines, but many disciplines have impacted computing. Mathematics, Ethics [26], Education, Law, Business, Engineering, Art [39], History, Biology [27] and more have all influenced computing and will continue to do so. Terminology of course plays a large role in this. However, it is the application of computing in other contexts that is the focus of the current work.

2.3 The Wider Context

Challenges due to terminology within computing and computing education extend far beyond the immediate context we discuss in this paper, particularly as terminology is a facet of vocabulary which itself is a facet of language. Several other areas have been researched that involve the role of terminology in computing including metaphor [28, 36], metonymy [29] the relationship of programming and natural language [3], how non-native English speakers

learn computing [18], and even programming keywords [41] and identifier names [23]. The GNU-Licensed *Free Online Dictionary of Computing* (FOLDOC) [25] is 35 years old and currently lists 15,205 terms (many of which have more than one definition). Clearly, the language of computing is complex.

2.4 Even Wider: Computing in Other Disciplines

Incorporating computing competencies into other disciplines requires at least an inspection of the term computing [20]. Simon et al. found that *computing* appears to be the least value-laden term (among others such as computer science, informatics, etc.) and the term that is understood in the greatest number of contexts [40]. However, the scope of their work was only *within* computing education. Stepping outside the discipline of computing, the term *digital* has a particular prominence, exemplified in the term *digital literacy*. Deloitte has defined digital literacy as “a basic understanding of the digital media and major digital platforms relevant to a particular domain” [13, p13]. This is in rather stark comparison to Simon et al. who, despite not formally defining digital literacy, said [40, p183]:

We did not encounter digital literacy or digital skills as the name of a degree, and we would not expect to, because we have only ever seen this term used to mean the computing skills that every reasonably educated person is expected to have: word processing, spreadsheets, presentation software, and perhaps simple webpage construction.

A report by The Committee on European Computing Education, jointly established by Informatics Europe and ACM Europe, defined Computing and Digital Literacy side-by-side [32, p6]:

- Informatics (described in the report as a synonym for computing and computer science): “knowledge and competencies about computational structures, processes, artefacts, and systems.”
- Digital Literacy: “basic user skills, i.e. fluency with standard tools such as word processors, web browsers, spreadsheets, etc.”

Obviously, future lawyers, artists, and zoologists will be better lawyers, artists and zoologists if they possess more than digital literacy as defined here. Indeed, it could be argued that all of these professionals should have “knowledge and competencies about computational structures, processes, artefacts, and systems”. Using terms such as structures, processes, artefacts and systems might in itself be problematic without defining them, but it is hard to see how the following examples would not fall under this broad umbrella in the following contexts, placing them clearly beyond the realm of digital literacy, and indicating that *computing* competencies might be crucial to disciplines such as:

- Archaeology: using digital imaging and scanning to restore ancient artwork [12]
- Literature: using data mining and text analysis to discover historical metaphors [34]
- Zoology: using technology to enhance zoological parks [5]

Clearly, this was the vision of Denning et al. who stated that computing students should be able to “work effectively with people in

other disciplines to help design modes of effective action in those disciplines.” [7, p13].

3 METHOD

We interviewed 20 educators from the following disciplines / departments: Adult & Community Education, Archaeology, Architecture, Art, Biochemistry, Dentistry, Economics, Education, Educational Development, English, Library Studies, Marketing & Entrepreneurship, Music, Nautical Science, Occupational Safety & Health, Organic & Medicinal Chemistry, Pharmacy, Physics, Sociology, and Student Skills Centre. Participants were recruited by calls on social media, and verified as university-level educators. We fortunately ended up with participants from every UNESCO International Standard Classification of Education (ISCED) [15] top-level field other than *Information and Communications Technologies* which we intentionally did not recruit from as this is the top-level categorization for the computing disciplines. All participants are third-level (university) lecturers at institutions in Ireland. None of the interviewees teach computing other than what is required for their (non-computing) degree programs. Interviews were conducted online one-on-one using the university videoconferencing system and were semi-structured, with each following the same protocol prompts with flexibility to allow those being interviewed to express their own experiences in their own contexts. Interviews focused on the use of computing competencies in the disciplinary context of the interviewee and featured terminology as a focus. This study was conducted with approval from the National Forum for the Enhancement of Teaching and Learning in Higher Education.

4 RESULTS

4.1 RQ 1: What Terms are Used and How

Our first research question was: *What computing terminology is used in non-computing disciplines, and in what ways are these used?* Interviewees were asked for specific computing words used in their disciplinary contexts and for examples of their use. Figure 1 shows all 58 terms mentioned during this phase of the interviews. Words that were mentioned by more than one interviewee are in larger font size. If a particular interviewee mentioned a specific word more than once, it was only counted once. Table 1 shows all 13 words that were mentioned by more than one interviewee, along with the number of interviewees who mentioned a given term.

Table 1: All terms mentioned by more than one interviewee

term	# interviewees	term	# interviewees
digital	7	digital learning	2
databases	4	digital native	2
digital literacy	4	download	2
programming	4	import	2
artificial intelligence	3	lab	2
modeling	3	simulation	2
		upload	2

These terms fall under three largely distinct categories of use: 1) the **application** of core computing topics such as *databases*, *artificial intelligence*, *modeling*, *programming*, and *simulation* in

Figure 1: All computing terms mentioned by interviewees. Larger font size indicates words mentioned by more than one interviewee. The smallest font corresponds to one interviewee and the largest corresponds to seven.

other disciplines; 2) the general **use of computers** exemplified by *upload* and *download*; and 3) general **“digital” education** such as *digital*, *digital literacy*, *digital learning*, *digital native*, etc.

Inspecting the other terms that were only mentioned by one interviewee each reveals that these divisions generally apply to most, if not all, terms mentioned. For instance, *command line*, *coding*, *algorithm*, and *computational thinking* seem to relate to the **application** of core computing concepts while others such as *file extensions* and *online* seem to relate to the **use of computers** while *e-learning* and *digital skills* relate to **general “digital” education**.

It is worth commenting on *digital* on its own, typified by its use in *digital literacy*, etc. In this use, *digital* is a general term for computing technology as noted by [40], not the technical term *digital* relating to data stored or transmitted as a sequence of discrete symbols from a finite set, most commonly taken to be binary. Several interviewees singled out *digital* as being a source of confusion.

Several software types were mentioned in interviews, predominantly: office, video conferencing and learning management systems. A few task-specific examples were mentioned such as STATA and CAD. The only references to hardware was one mention of Raspberry Pi and a handful of discipline-specific mentions of specific laboratory or medical equipment. We discuss the programming languages mentioned in Section 6.2.

4.2 RQ 2: Identifying Barriers

Our second research question was: *What terminology barriers are faced by educators in non-computing disciplines, when teaching computing competencies within their disciplines?*

Interviewees were directly asked if computing terminology ever poses barriers to their teaching. Of the 20 interviewees, 15 answered “yes” and 5 answered “not much” or “not a serious barrier”

or similar. No interviewee simply said “no” or any other firm negative response. When asked to describe these barriers, a number of interesting manifestations arose. Several interviewees mentioned that vagueness and simple confusion surrounding the meanings of certain terms were problematic:

- *Digital is so vague.*
- *Digital and digital literacy are problematic as it is difficult to determine what these really mean.*
- *The plain English meaning of terms vs. the computing meaning of the same terms is problematic.*

Others noted that constantly changing terminology is problematic:

- *What do we call things? It shifts – it’s confusing.*
- *There is a fashion. Do people ever use new terms because they are better, or just because they are fashionable?*

Other interviewees mentioned that how it can’t be assumed that students will know computing terminology:

- *It never ceases to amaze me how little [technical] knowledge students have.*
- *Digital native does not mean digitally literate.*
- *In our discipline we are hypocrites in that we teach our students not to use acronyms and jargon but then we teach with them.*
- *We can’t presume knowledge. We have to check understandings before we get too deep.*
- *Students have associations with terminology that can be hard to break.*

Nonetheless at least one interviewee was more positive about terminology and cited deeper issues as a barrier: *The basic digital literacy is there. The problem is application of design and principles.*

Other interesting experiences and observations were expressed by several interviewees. One mentioned that terminology can be an issue for dyslexic students. Another mentioned student *Digital confidence* as a barrier. We discuss more in Section 5.

4.3 RQ3: Overcoming Barriers

Our third research question was: *What can be done to overcome these terminology barriers, and are there approaches taken by one discipline that could be applied in another?*

Interviewees revealed several strategies to deal with terminology in teaching computing competencies. One interviewee stated that they *make a conscious decision to have a common language from day one with regard to computing and technology terms, and that this is aimed at getting students to be talking the same language as staff*. Another simply stated that they *teach the terminology of the technology before the disciplinary stuff*. Another mentioned metaphor: *... relate technologies to old, more familiar and commonly understood things. Talking about technology in the abstract isn’t enough – you need concrete examples*. Two interviewees mentioned that their students take mandatory courses specifically on computer skills.

Several interviewees mentioned consistency, glossaries, and other systematic ways to address terminology issues. One said: *There is so much out there. Pick the most benign terms and don’t nitpick. Define it and use it. Have a glossary and use it consistently*. Another mentioned reinforcement: *Reinforcing something that someone knows is redundant but but leaving someone out is a tragedy*.

Two interviewees noted a 'difference' between the language of students and the language of staff and the need for the agreement of a middle ground. Another noted that it can be useful to identify students in the class who do know more than others and empower those students to help those that need it, stating: *The students will listen to other students in a way that they won't listen to me. Students often explain to other students with stories or examples that students can relate to - it's a better way of learning.*

5 UNEXPECTED FINDINGS

We made several unexpected observations during the course of these interviews. Despite not directly addressing research questions, these findings illustrate the diversity of various disciplines as well as the complexity and nuance that is involved in the terminology of computing when used in non-computing contexts.

5.1 Linguistic Terminology

A striking number of linguistic terms were mentioned, indicating the nuance and complexity surrounding computing terminology, including: acronym speak; fluency; jargon; lexicon; lingo; natural language; parlance; pigeon language; slang; translate; vernacular.

5.2 It's Really Not All About Coding and Computational Thinking

Coding, programming, and computational thinking were mentioned surprisingly infrequently. Four interviewees mentioned *programming*, one mentioned *coding*, and one mentioned *computational thinking*. In terms of programming languages, two mentioned *R*, while *Python* and *Excel* (programming) came up once each. No task-specific programming languages were mentioned, although specific software packages were mentioned as discussed in Section 4.1.

Databases and *artificial intelligence* and a handful of other core computing topics were mentioned, but none with the frequency of other more general terms such as *digital* or with the plurality of terms such as *upload*, *download*, *import*, *navigate*, *online*, etc. and the "digital" terms such as *digital*, *digital literacy*, *digital native*, *digital learning*, etc. Computing educators should take these findings into account when working with non-computing educators and students, for instance when engaging in computational thinking or coding initiatives for non-computing students.

5.3 Barriers and Disagreements Among Staff

Several of the reported terms, as well as interview details with a few participants, revealed that terminology is not only a barrier in the classroom when dealing with students. A picture emerged of differences between colleagues: younger & older, and 'tech proponents' & 'stalwarts' / hold-outs' or simply the 'less knowledgeable'. One interviewee recalled how different interpretations of terminology among staff led to debates, and others described the terminology used on departmental websites and teaching materials being sticking points among staff. Another described needing to be strategic when using certain terms with colleagues.

5.4 Effect of Age

Although not explicitly asked, it is notable that several interviewees mentioned student age as a factor when discussing terminology as a barrier. Fewer mentioned staff age, but that was also unexpected. Further analysis revealed that 8 interviewees mentioned age and 8 did not. Several interviewees focused on observations of first-year students but others did mention graduate students and adult learners as also experiencing difficulties with computing terminology. Although this may seem obvious the reality may not be as simple as it seems, and this observation warrants further investigation.

6 LIMITATIONS

This study is not without limitations. Our interviewees were not globally dispersed, however there is little reason to think that they would be much different to those in several other North American, European, and Australasian countries for our purposes. Although not large on some standards, for qualitative interviews of this nature our sample size is considered appropriate [2], particularly given the contextual details such as our wide range of disciplines represented. In order to ensure that our results are as representative as possible, we interviewed educators from as many of the eleven UNESCO International Standard Classification of Education (ISCED) [15] top-level fields as possible. Our participants represent ten with all but one represented at least twice. The only one not represented is "06 Information and Communications Technologies" which we intentionally did not recruit from as this is the top-level categorization for the computing disciplines.

7 CONCLUSIONS

We present the results of interviews with 20 educators from non-computing disciplines at higher education institutions. Together these educators represent all non-computing ISCED top-level education fields on the use of computing terminology in their disciplinary contexts. These interviews revealed that the computing terminology used is diverse, but can be divided into three main categories: the *application* of core computing topics such as "programming"; the *general use of computers* such as "upload" / "download"; and *general digital education* terminology such as "digital literacy".

Of the 20 interviewees, 15 indicated that terminology poses a barrier to their teaching, and 5 indicated "not much" or "not a serious barrier" or similar. Several barriers caused by terminology were identified broadly categorised as: vague / confusing; constantly changing; and differences between student and staff understandings. Many interesting approaches to dealing with these barriers were revealed, most of which can be applied in other disciplines.

Many interviewees offered their approaches to dealing with and overcoming barriers presented by computing terminology. Most, if not all of these could be applied by educators in other disciplines.

We also revealed several unexpected findings that warrant further study including that age is a perceived factor in barriers presented by terminology, and that some core computing terms (e.g. *programming*) may play more minor roles than other more general terms such as *digital* regarding the barriers they present. These results demonstrate that the use of computing terminology in non-computing disciplines is complex and nuanced, and can take on several distinct modes of use. Future research is required on specific

facets of these phenomena to better understand how computing is used in non-computing disciplines and how that can be best achieved.

ACKNOWLEDGMENTS

This research was funded through a National Teaching and Learning Research Fellowship, awarded by the National Forum for the Enhancement of Teaching and Learning in Higher Education.

REFERENCES

- [1] Amazon AWS. 2020. Training ML Models. <https://docs.aws.amazon.com/machine-learning/latest/dg/training-ml-models.html>
- [2] Sarah Elsie Baker and Rosalind Edwards. 2012. How many qualitative interviews is enough. (2012).
- [3] Brett A. Becker. 2019. Parlez-vous Java? Bonjour La Monde != Hello World: Barriers to Programming Language Acquisition for Non-Native English Speakers. In *30th Workshop of the Psychology of Programming Interest Group - PPIG '19* (Newcastle, UK). 13 pages. www.brettbecker.com/publications
- [4] Jeffrey Bonar and Elliot Soloway. 1985. Preprogramming Knowledge: A Major Source of Misconceptions in Novice Programmers. *Human-Computer Interaction* 1, 2 (1985), 133–161.
- [5] Andrea W Clay, Bonnie M Perdue, Diann E Gaalema, Francine L Dolins, and Mollie A Bloomsmith. 2011. The Use of Technology to Enhance Zoological Parks. *Zoo biology* 30, 5 (2011), 487–497.
- [6] Alison Clear and Allen Parrish. 2020. Computing Curricula 2020 (Draft Version 44). (Nov 2020). <https://cc2020.nsparc.msstate.edu/wp-content/uploads/2020/11/Computing-Curricula-Report.pdf>
- [7] D. E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, Paul R. Young, and Peter J. Denning. 1989. Computing as a Discipline. *Commun. ACM* 32, 1 (Jan. 1989), 9–23. <https://doi.org/10.1145/63238.63239>
- [8] K-12 Computer Science Framework Steering Committee. 2016. *K-12 Computer Science Framework*. Technical Report. New York, NY, USA.
- [9] Peter Denning. 2000. Computer Science: The Discipline. In *Encyclopedia of Computer Science*, Anthony Ralston, Edwin D. Reilly, and David Hemmendinger (Eds.). Nature Publishing Group, UK, 9–23.
- [10] Peter J. Denning. 1989. A Debate on Teaching Computing Science. *Commun. ACM* 32, 12 (Dec. 1989), 1397–1414. <https://doi.org/10.1145/76380.76381>
- [11] Paul Denny, Brett A. Becker, Michelle Craig, Greg Wilson, and Piotr Banaszkievicz. 2019. Research This! Questions That Computing Educators Most Want Computing Education Researchers to Answer. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) (ICER '19). Association for Computing Machinery, New York, NY, USA, 259–267. <https://doi.org/10.1145/3291279.3339402>
- [12] Graeme Earl, Kirk Martinez, and Tom Malzbender. 2010. Archaeological Applications of Polynomial Texture Mapping: Analysis, Conservation and Representation. *Journal of Archaeological Science* 37, 8 (2010), 2040–2050.
- [13] Peter Evans-Greenwood, Tim Patston, and Amanda Flouch. 2019. The Digital Ready Worker: Digital Agency and the Pursuit of Productivity. (2019). https://www2.deloitte.com/content/dam/insights/us/articles/AU22848_The-digital-ready-worker/DI-The-digital-ready-worker.pdf
- [14] Hannah Fidoten and Jaime Spacco. 2012. What Do Computer Scientists Do? A Survey of CS and Non-CS Liberal Arts Faculty. In *Proceedings of the 17th ACM Annual Conference on Innovation and Technology in Computer Science Education* (Haifa, Israel) (ITiCSE '12). Association for Computing Machinery, New York, NY, USA, 279–284. <https://doi.org/10.1145/2325296.2325362>
- [15] Unesco Institute for Statistics. 2014. ISCED Fields of Education and Training 2013 (ISCED-F 2013): manual to accompany the International Standard Classification of Education. (2014).
- [16] Shuchi Grover, Roy Pea, and Stephen Cooper. 2016. Factors Influencing Computer Science Learning in Middle School. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education* (Memphis, Tennessee, USA) (SIGCSE '16). Association for Computing Machinery, New York, NY, USA, 552–557. <https://doi.org/10.1145/2839509.2844564>
- [17] Fred Gruenberger. 1960. A Terminology Proposal. *Commun. ACM* 3, 2 (Feb. 1960), 72. <https://doi.org/10.1145/366959.366962>
- [18] Philip J. Guo. 2018. Non-Native English Speakers Learning Computer Programming: Barriers, Desires, and Design Opportunities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3173574.3173970>
- [19] Mark Guzdial. 2020. Dijkstra Was Wrong About 'Radical Novelty': Metaphors in CS Education. <https://cacm.acm.org/blogs/blog-cacm/248985-dijkstra-was-wrong-about-radical-novelty-metaphors-in-cs-education/fulltext>
- [20] Mark Guzdial. 2021. What Liberal Arts and Sciences Students Need to Know About Computing. <https://cacm.acm.org/blogs/blog-cacm/249787-what-liberal-arts-and-sciences-students-need-to-know-about-computing/fulltext>
- [21] Mark Guzdial and Elliot Soloway. 2003. Computer Science is More Important than Calculus: The Challenge of Living up to Our Potential. *SIGCSE Bull.* 35, 2 (June 2003), 5–8. <https://doi.org/10.1145/782941.782943>
- [22] Wilhelm Hasselbring. 1999. Technical Opinion: On Defining Computer Science Terminology. *Commun. ACM* 42, 2 (Feb. 1999), 88–91. <https://doi.org/10.1145/293411.293488>
- [23] Johannes Hofmeister, Janet Siegmund, and Daniel V Holt. 2017. Shorter Identifier Names Take Longer to Comprehend. In *2017 IEEE 24th International conference on software analysis, evolution and reengineering (SANER)*. IEEE, 217–227.
- [24] J. E. Holmstrom. 1960. The Multilingual Terminology Project. *Commun. ACM* 3, 7 (July 1960), 409–412. <https://doi.org/10.1145/367349.367354>
- [25] Denis Howe. [n.d.]. *Free Online Dictionary of Computing*. <http://foldoc.org/>.
- [26] Chuck Huff and Laura Barnard. 2009. Good computing. *IEEE Technology and Society Magazine* 28, 3 (2009), 47–54.
- [27] William B Langdon and Riccardo Poli. 2013. *Foundations of Genetic Programming*. Springer Science & Business Media.
- [28] Andrew Manches, Peter E McKenna, Gnanathusharan Rajendran, and Judy Robertson. 2020. Identifying Embodied Metaphors for Computing Education. *Computers in Human Behavior* 105 (2020), 105859.
- [29] Craig S. Miller and Amber Settle. 2019. Learning to Get Literal: Investigating Reference-Point Difficulties in Novice Programming. *ACM Trans. Comput. Educ.* 19, 3, Article 28 (May 2019), 17 pages. <https://doi.org/10.1145/3313291>
- [30] Bhagya Munasinghe, Tim Bell, and Anthony Robins. 2021. Teachers' understanding of technical terms in a Computational Thinking curriculum. In *Proceedings of the 21st Australasian Computing Education Conference* (Virtual, SA, Australia) (ACE '21). Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/3441636.3442311>
- [31] Nature. [n.d.]. Biological models. <https://www.nature.com/subjects/biological-models>
- [32] The Committee on European Computing Education (CECE). 2017. *Informatics Education in Europe: Are We All In The Same Boat?* Technical Report. New York, NY, USA.
- [33] Zacharoula Papamitsiou, Michail Giannakos, Simon, and Andrew Luxton-Reilly. 2020. Computing Education Research Landscape through an Analysis of Keywords. In *Proceedings of the 2020 ACM Conference on International Computing Education Research* (Virtual Event, New Zealand) (ICER '20). Association for Computing Machinery, New York, NY, USA, 102–112. <https://doi.org/10.1145/3372782.3406276>
- [34] Brad Pasanek and D Sculley. 2008. Mining Millions of Metaphors. *Literary and linguistic computing* 23, 3 (2008), 345–360.
- [35] Kathryn Ptacek. 1979. The Spectre of the Machine. In *Proceedings of the 7th Annual ACM SIGUCCS Conference on User Services* (Los Angeles, California) (SIGUCCS '79). Association for Computing Machinery, New York, NY, USA, 112–113. <https://doi.org/10.1145/601889.601944>
- [36] Joseph P. Sanford, Aaron Tietz, Saad Farooq, Samuel Guyer, and R. Benjamin Shapiro. 2014. Metaphors We Teach By. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, Georgia, USA) (SIGCSE '14). Association for Computing Machinery, New York, NY, USA, 585–590. <https://doi.org/10.1145/2538862.2538945>
- [37] Neil Selwyn. 2014. *Digital Technology and the Contemporary University: Degrees of digitization*. Routledge.
- [38] Sue Sentance and Andrew Cizmadia. 2017. Computing in the Curriculum: Challenges and Strategies From a Teacher's Perspective. *Education and Information Technologies* 22, 2 (2017), 469–495.
- [39] Lior Shamir and Jane A. Tarakhovskiy. 2012. Computer Analysis of Art. *J. Comput. Cult. Herit.* 5, 2, Article 7 (Aug. 2012), 11 pages. <https://doi.org/10.1145/2307723.2307726>
- [40] Simon, Alison Clear, Janet Carter, Gerry Cross, Atanas Radenski, Liviana Tudor, and Eno Töniss. 2015. What's in a Name? International Interpretations of Computing Education Terminology. In *Proceedings of the 2015 ITiCSE on Working Group Reports* (Vilnius, Lithuania) (ITiCSE-WGR '15). Association for Computing Machinery, New York, NY, USA, 173–186. <https://doi.org/10.1145/2858796.2858803>
- [41] Andreas Stefk and Susanna Siebert. 2013. An Empirical Investigation into Programming Language Syntax. *ACM Trans. Comput. Educ.* 13, 4, Article 19 (Nov. 2013), 40 pages. <https://doi.org/10.1145/2534973>
- [42] ACARA Digital Technologies. [n.d.]. *A-Z Digital Technologies vocabulary F–6*. Technical Report. https://www.australiancurriculum.edu.au/media/5544/a-z-digital-technologies-vocabulary-k-6_v2.pdf
- [43] Jessica Vandenberg, Jennifer Tsan, Danielle Boulden, Zarifa Zakaria, Collin Lynch, Kristy Elizabeth Boyer, and Eric Wiebe. 2020. Elementary Students' Understanding of CS Terms. *ACM Trans. Comput. Educ.* 20, 3, Article 17 (June 2020), 19 pages. <https://doi.org/10.1145/3386364>