Developing a Concept Inventory for Computer Science 2

Lea Wittie
Department of Computer Science
Bucknell University
Lewisburg, PA 17837
Email: lwittie@bucknell.edu

Anastasia Kurdia
Department of Computer Science
Tulane University
New Orleans, LA 70118
Email: akurdia@tulane.edu

Meriel Huggard
School of Computer Science and Statistics
Trinity College Dublin
Dublin 2, Ireland
Email: Meriel.Huggard@tcd.ie

***Abstract*—Computer Science 2 is one of the core elements of any computer science program; however, there has been little progress towards the development of a concept inventory for this key introductory course for CS majors. Plans to address this gap using the Delphi method are presented. Instructors and researchers with knowledge and experience of CS2 are being recruited to help identify concepts from CS2 that are both important and challenging. These will then be distilled, through an iterative research methodology, to create a concept inventory that will undergo validity and reliability checking through large scale field testing, qualitative interview analysis and appropriate psychometric testing.**

I. Introduction

Instructors have long sought to explore their students' understanding of the concepts and phenomena that lie at the heart of the subject area they are teaching [1], [2], [3], [4]. These explorations rely on a clear articulation of the key concepts that underpin the curriculum, and the common student misconceptions that are associated with them. One way of establishing a coherent set of concepts that capture the quality of the learning achieved within classrooms is through the development of conceptual frameworks [5].

Researchers have tried to explore these conceptual frameworks through the identification of threshold concepts [6]. These concepts are not only part of the set of core concepts of the subject area, they are also (i) transformative, (ii) integrative, (iii) irreversible, (iv) troublesome and (v) boundary markers [6]. Once the threshold concepts for a discipline have been established they can then assist in the organization and focus of student learning [7]; however, identifying such concepts has proved challenging [8] and there is little consensus on the concepts that have been proposed in the literature [9].

Conceptual frameworks have also been developed as a way of fostering peer learning [10]; for example, Mazur's [11] multiple-choice based "ConcepTests" for introductory physics were intended to provide the instructor with immediate feedback in each class session. The students' responses allowed the instructor to gauge the class' understanding and to identify misconceptions they may have acquired. They were also used to foster cooperative learning [12] within the classroom using a think-pair-share approach [13]. These ConcepTests were shown to help students develop a better understanding of the fundamental concepts of introductory physics [11].

A more practical approach to capturing conceptual frameworks is in the development of "research-based distractor driven multiple-choice instruments" known as concept inventories [14]. These have been developed for a wide range of specific courses and fields; including the seminal work on the Force Concept Inventory for Physics [1], as well as concept inventories for Physiology [15], Chemistry [16] and Statistics [17].

Existing concept inventories for computer science related topics currently include Introductory Programming in C [18], Introductory programming (i.e. CS1) [19], [20] , Binary Search Trees versus Hash Tables (3 questions) [21], an upper level course in Algorithms and Data Structures [22], Software Engineering [23], Discrete Math [24], Computer Architecture [25], Operating Systems [4], and Digital Logic [26], [27]. There is also ongoing work on the development of a concept inventory for Algorithm Analysis [28].

One of the most notable absences from this list relates to one of the key introductory sequence courses for all computer science majors, namely a course on basic data structures. This is often referred to in the literature as CS2 [29]. This is the gap that the authors of this work are seeking to address.

It should be noted that while the terms CS1 and CS2 are standard in computer science, the actual content of CS1 and CS2 courses varies widely [29]. Hence the authors expect to encounter a wide variation in the topics that experts consider to be important in a CS2 course. The CS2 topics of particular interest in this study relate to basic abstract data structures such as list, stack, tree, and queue. However, the authors will be guided by the viewpoints expressed by the experts who participate in the study.

II. Concept Inventories

A concept inventory (CI) is a research-based multiple-choice test that seeks to measure a student's knowledge of a set of concepts while also capturing conceptions and misconceptions they may have about the topic under consideration. Each multiple choice question includes one correct answer and a set of incorrect answers that would result from one or more misconceptions. One of the earliest concept inventories in computer science is that of Almstrum et al. [24].

978-1-5090-5920-1/17/$31.00 ©2017 IEEE

At its simplest, a concept inventory is a straightforward assessment instrument that is easily administered, albeit one that is highly tailored to the specific subject domain under consideration. It can be used for formative purposes e.g. for diagnostic testing or for identifying appropriate teaching and learning activities [30]. A CI can be used to assess the impact of changing instruction methods on student understanding, at both the individual and the collective level. Given that a CI presents common misconceptions as options, it can also be used to give feedback to students on which topics they have yet to fully grasp and to provide them with suggestions as to where their future learning efforts are best directed. Instructors may use a CI to identify topics that need a different instructional approach or as a way of comparing instructional methods [24]. They can also be used for summative purposes; for example, to evaluate overall learning and instructional effects [30]. However, it is important to note that a CI is not an appropriate instrument for assessing teaching performance or for determining grades [24].

Generally the CI is administered both before and after an instructional period; for example, a topic unit or a course. In computer science, unlike many other fields, students' preconceptions may arise from prior instruction rather than from a naive understanding based on their experience of the world around them [19] [31]. Indeed, some have argued that CS students often enter the classroom with no prior conceptions of the field and so administering a CI before teaching may not be useful [32]. However, this may not be the case in the future as topics in computer science are increasingly becoming part of the mainstream K-12 curriculum; for example through initiatives such as the K-12 computer science framework [33] or short courses designed to incentivize students to pursue computer science programmes [34], [35].

III. Constructing a Concept Inventory for CS2

Most concept inventories are constructed using some variant of the Delphi method [36]. Although they may differ in terms of some of the specifics, they share a common theme of obtaining a set of topics, creating questions, conducting field tests, carrying out validity checks and reliability testing.

The Delphi method is a user-friendly and systematic process to gain consensus among a group of experts. Since its development in the 1950's [37], it has been used in curriculum development, adult education, nurse education, determining educational effectiveness, distance education, vocational training, and many other education-related tasks [38].

The Delphi method tries to overcome the weaknesses in other approaches including using a single expert, taking a plain average of expert opinions, and having a expert roundtable come to agreement together. A single expert's judgment is never sufficient in subjects that use qualitative or other opinion-based answers. The averaging of experts does not allow the experts to reflect on, and respond to, what others have said. Roundtable groups tend to "polarize opinions" and come to more extreme conclusions that individual experts [38].

The strengths of the Delphi method include the anonymity of its experts which allows them to make decisions free from social peer pressure [38] and reduces the ability of dominant individuals to sway other experts [39]. Its several phases of controlled feedback reduce noise in the results and allow the experts to reflect, form opinions, and justify their decisions [39]. Using quartile statistics on the data set allows the experts to reflect while still preserving all of their varied opinions [39].

Many of the CI's mentioned above, e.g. [40], [32], and [24], use a similar iterative process to gather expert opinions and achieve consensus. The experts are first asked to list a set of topics they consider to be both important and difficult for a given subject. The experts are then asked to rank the overall set of topics on difficulty and importance. Several more iterations occur where the experts are given more and more statistical information on how their chosen rankings relate to the overall group choices and also anonymized justifications for rankings outside of the mainstream.

The following sections set out the methodology that will be followed in this work.

IV. Methodology to Obtain the CS2 Concepts

For our purposes, the works of Dalkey [37], Pill [39], and Clayton [38] show that the Delphi method has been a long accepted and much studied process for gaining consensus. Our aims are perfectly aligned with the purposes of the Delphi methodology as we will be using it to create a tool to determine educational effectiveness. We plan to follow the suggestions for best practice set out in the literature; for example, by Lindell et al. [14] and Goldman et al. [40].

Following Clayton's recommendation [38], we are seeking to enlist the help of a group of 15-30 experts in the field. Ideally, we would like to capture a wide range of diverse perspectives from our expert group. We are seeking diversity in terms of the number of times they have taught a CS2 course, the size of their educational institution, typical class sizes, gender, race, geographical location, undergraduate and Ph.D. granting institutions, liberal arts colleges and engineering institutions, etc. Our aim is to bring together a group of experts who will provide a broad view of the topics that belong in a CS2 classroom. We also welcome experts from the CS and Engineering Education domains as well as those who have published educational materials for CS2. As in the standard Delphi method, our experts' specific decisions will remain anonymous. We will collect anonymized aggregate diversity data on our experts to show the breadth and depth of the expert pool.

In our first step, the experts will be asked to list 10-15 topics they consider both important and difficult for CS2. After we receive this data from the experts, the researchers associated with this project will independently compile a list of all topics mentioned by more than one expert. We will then concatenate the compiled sets to form the complete set. It may be necessary to combine topics where there is significant overlap.

Second, we will send the complete set to each expert and ask them to rank each topic on a scale of 1 to 10 for difficulty and

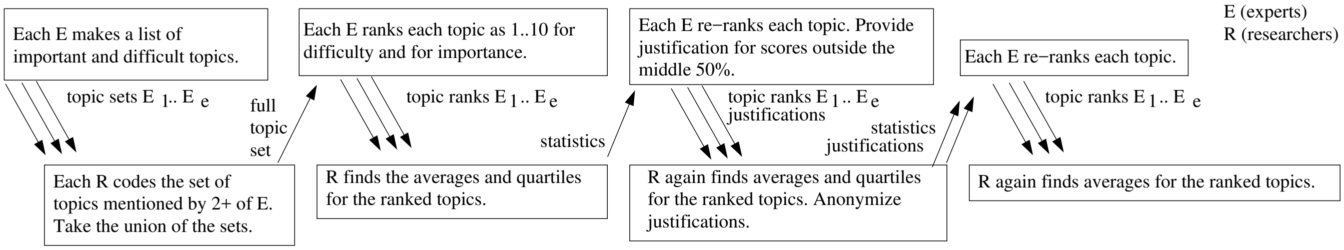

Fig. 1. Workflow for the proposed Delphi method.

also, separately, for importance. After we receive the rankings, the researchers will calculate the averages and inner quartiles to find the middle 50% of the rankings.

Third, we will provide the averages and quartile data of the complete set of rankings to the experts and ask them to re-rank it for difficulty and importance again. In this phase the experts will provide a justification for any of their scores that fall outside the middle 50% range from the previous phase. The researchers will again calculate the averages and inner quartiles of this phase and collect the full set of anonymized justifications.

Fourth, we will provide the new averages and quartile data along with the anonymized justifications to our experts and ask them to rank the topics for difficulty and importance one last time. Using these final rankings, we will compute the averages for difficulty and importance for each topic. Following the procedures in Goldman [40], we will compute the distance of each average as a point (importance, difficulty) from the maximum ranking (10, 10) and select (approximately) the 10 topics closest to the (10, 10) point. We aim to establish a clear division between the top topics that will be included in the CI and those to be disregarded. Once agreement has been reached on the list of fundamental concepts and the list of misconceptions to be included, the researchers will create the first, pilot version of the CI.

V. Field Testing, Validity and Reliability

Once the initial concept inventory has been developed it will then be field tested using a sample population of between 500 and 1000 students. It is intended that this student cohort will be drawn from a wide range of educational institutions worldwide. Statistics will then be calculated for each item in the inventory. For example, it will be necessary to determine what percentage of students answer each question correctly and how well each item discriminates between the quartiles of the class [14]. Items found to be invalid will be deleted or amended. It is expected that a number of field tests will be needed to ensure that all items satisfy the CI requirements.

Criterion, construct and content validity tests will then be conducted. While it is expected that these will be quite time consuming and onerous, they do not require any additional input from the experts who assist the researchers with the initial phase of the research. For example, in order to determine the criterion validity of the CI, it will be necessary to compare student performance on the CI with other measures of their performance e.g. standardized tests or course grades. Construct validity will be established through interviews with students who take the test; while content validity will require the opinion of a group of experts who were not involved in the construction of the CI.

Reliability will be determined though statistical methods used to determine the reliability of psychometric tests e.g. Cronbach's alpha test [26].

VI. Future work

This is a work in progress and we are actively recruiting experts to participate in the initial Delphi method described in this paper. Our experts will have the option to also participate in field testing of the Concept Inventory and will have access to the Concept Inventory to use in their own educational and research work. To volunteer as an expert, please contact the research team (info@cs2ci.org).

References

[1] D. Hestenes, M. Wells, and G. Swackhamer, "Force concept inventory," *The physics teacher*, vol. 30, no. 3, pp. 141–158, 1992.

[2] M. T. Chi, J. D. Slotta, and N. De Leeuw, "From things to processes: A theory of conceptual change for learning science concepts," *Learning and instruction*, vol. 4, no. 1, pp. 27–43, 1994.

[3] K. C. Midkiff, T. A. Litzinger, and D. L. Evans, "Development of Engineering Thermodynamics Concept Inventory instruments," in *31st Annual Frontiers In Education Conference (FIE)*, vol. 2, 2001, pp. F2A–F23.

[4] K. C. Webb and C. Taylor, "Developing a pre- and post-course concept inventory to gauge operating systems learning," in *Proceedings of the 45th ACM technical symposium on computer science education*. ACM, 2014, pp. 103–108.

[5] N. Entwistle, "Concepts and conceptual frameworks underpinning the ETL project," 2003.

[6] J. Meyer and R. Land, *Threshold concepts and troublesome knowledge: Linkages to ways of thinking and practising within the disciplines*. University of Edinburgh, 2003.

[7] A. Eckerdal, R. McCartney, J. E. Moström, M. Ratcliffe, K. Sanders, and C. Zander, "Putting threshold concepts into context in computer science education," in *ACM SIGCSE Bulletin*, vol. 38, no. 3. ACM, 2006, pp. 103–107.

[8] D. Shinners-Kennedy and S. A. Fincher, "Identifying Threshold Concepts: From Dead End to a New Direction," in *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research (ICER)*. New York, NY, USA: ACM, 2013, pp. 9–18. [Online]. Available: http://doi.acm.org/10.1145/2493394.2493396

[9] K. Sanders and R. McCartney, "Threshold Concepts in Computing: Past, Present, and Future," in *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '16. New York, NY, USA: ACM, 2016, pp. 91–100. [Online]. Available: http://doi.acm.org/10.1145/2999541.2999546

[10] C. M. Goldrick and M. Huggard, "Peer learning with Lego Mindstorms," in *34th Annual Frontiers in Education (FIE)*, Oct 2004, pp. S2F–24–9 Vol. 3.

[11] E. Mazur, "Peer instruction: getting students to think in class," in *AIP Conference Proceedings*, vol. 399, no. 1. AIP, 1997, pp. 981–988.

[12] M. Huggard, F. Boland, and C. Mc Goldrick, "Using cooperative learning to enhance critical reflection," in *44th Annual Frontiers in Education Conference (FIE)*, Oct 2014, pp. 1–8.

[13] F. T. Lyman, "The responsive classroom discussion: The inclusion of all students," *Mainstreaming digest*, vol. 109, p. 113, 1981.

[14] R. S. Lindell, E. Peak, T. M. Foster, L. McCullough, L. Hsu, and P. Heron, "Are they all created equal? a comparison of different concept inventory development methodologies," in *AIP Conference Proceedings*, vol. 883, no. 1. AIP, 2007, pp. 14–17.

[15] J. Michael, W. Cliff, J. McFarland, H. Modell, and A. Wright, *The Core Concepts of Physiology: A New Paradigm for Teaching Physiology*. New York, NY: Springer New York, 2017.

[16] S. Krause, J. Birk, R. Bauer, B. Jenkins, and M. J. Pavelich, "Development, testing, and application of a chemistry concept inventory," in *34th Annual Frontiers In Education Conference (FIE)*. IEEE, 2004, pp. T1G–1.

[17] A. Stone, K. Allen, T. R. Rhoads, T. J. Murphy, R. L. Shehab, and C. Saha, "The statistics concept inventory: A pilot study," in *33rd Annual Frontiers In Education Conference (FIE)*, vol. 1. IEEE, 2003, pp. T3D–1.

[18] R. Caceffo, S. Wolfman, K. S. Booth, and R. Azevedo, "Developing a Computer Science Concept Inventory for Introductory Programming," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. New York, NY, USA: ACM, 2016, pp. 364–369. [Online]. Available: http://doi.acm.org/10.1145/2839509.2844559

[19] A. E. Tew, "Assessing fundamental introductory computing concept knowledge in a language independent manner," Ph.D. dissertation, Georgia Institute of Technology, 2010.

[20] A. E. Tew and M. Guzdial, "The FCS1: a language independent assessment of CS1 knowledge," in *Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM, 2011, pp. 111–116.

[21] K. Karpierz and S. A. Wolfman, "Misconceptions and Concept Inventory Questions for Binary Search Trees and Hash Tables," in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*. New York, NY, USA: ACM, 2014, pp. 109–114. [Online]. Available: http://doi.acm.org/10.1145/2538862.2538902

[22] H. Danielsiek, W. Paul, and J. Vahrenhold, "Detecting and Understanding Students' Misconceptions Related to Algorithms and Data Structures," in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*. New York, NY, USA: ACM, 2012, pp. 21–26. [Online]. Available: http://doi.acm.org/10.1145/2157136.2157148

[23] J. Krone, J. E. Hollingsworth, M. Sitaraman, and J. O. Hallstrom, "A reasoning concept inventory for computer science," 2010.

[24] V. L. Almstrum, P. B. Henderson, V. Harvey, C. Heeren, W. Marion, C. Riedesel, L.-K. Soh, and A. E. Tew, "Concept inventories in computer science for the topic discrete mathematics," in *ACM SIGCSE Bulletin*, vol. 38, no. 4. ACM, 2006, pp. 132–145.

[25] L. Porter, S. Garcia, H.-W. Tseng, and D. Zingaro, "Evaluating student understanding of core concepts in computer architecture," in *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*. ACM, 2013, pp. 279–284.

[26] G. L. Herman, "The development of a digital logic concept inventory," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2011.

[27] G. Herman, M. Loui, and C. Zilles, "Administering a digital logic concept inventory at multiple institutions," in *Proceedings of the 2011 American Society for Engineering Education annual conference and exposition*, 2011, pp. 26–29.

[28] M. F. Farghally, K. H. Koh, J. V. Ernst, and C. A. Shaffer, "Towards a Concept Inventory for Algorithm Analysis Topics," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. New York, NY, USA: ACM, 2017, pp. 207–212. [Online]. Available: http://doi.acm.org/10.1145/3017680.3017756

[29] M. Hertz, "What Do "CS1" and "CS2" Mean?: Investigating Differences in the Early Courses," in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*. New York, NY, USA: ACM, 2010, pp. 199–203. [Online]. Available: http://doi.acm.org/10.1145/1734263.1734335

[30] Reed-Rhoads, T., and Imbrie, P. K., "Concept inventories in Engineering Education," in *National Research Council Promising Practices in Undergraduate STEM Education Workshop 2*, Washington, DC., 13 – 14 October 2008.

[31] M. Huggard, "Programming trauma: can it be avoided," *Proceedings of the BCS Grand Challenges in Computing: Education*, pp. 50–51, 2004.

[32] C. Taylor, D. Zingaro, L. Porter, K. Webb, C. Lee, and M. Clancy, "Computer science concept inventories: past and future," *Computer Science Education*, vol. 24, no. 4, pp. 253–276, 2014. [Online]. Available: http://dx.doi.org/10.1080/08993408.2014.970779

[33] "K-12 computer science framework," 2016. [Online]. Available: https://k12cs.org/

[34] M. Huggard and C. McGoldrick, "Incentivising students to pursue computer science programmes," in *36th Annual Frontiers In Education Conference (FIE)*, Oct 2006, pp. 3–8.

[35] M. Huggard and C. Mc Goldrick, "Computer experience–enhancing engineering education," in *Proceedings of the International Conference in Engineering Education*, July 2006, pp. T4C–21 – T4C–25.

[36] R. A. Green, "The Delphi Technique in Educational Research," *SAGE Open*, vol. 4, no. 2, p. 2158244014529773, 2014. [Online]. Available: http://dx.doi.org/10.1177/2158244014529773

[37] N. Dalkey and O. Helmer, "An experimental application of the Delphi method to the use of experts," *Management science*, vol. 9, no. 3, pp. 458–467, 1963.

[38] M. J. Clayton, "Delphi: a technique to harness expert opinion for critical decisionmaking tasks in education," *Educational Psychology*, vol. 17, no. 4, pp. 373–386, 1997. [Online]. Available: http://dx.doi.org/10.1080/0144341970170401

[39] J. Pill, "The Delphi method: Substance, context, a critique and an annotated bibliography," *Socio-Economic Planning Sciences*, vol. 5, no. 1, pp. 57–71, 1971. [Online]. Available: http://EconPapers.repec.org/RePEc:eee:soceps:v:5:y:1971:i:1:p:57-71

[40] K. Goldman, P. Gross, C. Heeren, G. Herman, L. Kaczmarczyk, M. C. Loui, and C. Zilles, "Identifying Important and Difficult Concepts in Introductory Computing Courses Using a Delphi Process," in *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*. New York, NY, USA: ACM, 2008, pp. 256–260. [Online]. Available: http://doi.acm.org/10.1145/1352135.1352226