DOI:
10.1145/3137065.3137085

Document Version
Peer reviewed version

Citation for published version (APA):
Kallia, M., & Sentance, S. (2017). Computing Teachers' Perspectives on Threshold Concepts: Functions and Procedural Abstraction. In *Proceedings of the 12th Workshop in Primary and Secondary Computing Education: WIPSCE '17* https://doi.org/10.1145/3137065.3137085

Computing Teachers’ Perspectives on Threshold Concepts: Functions and Procedural Abstraction

Maria Kallia
King’s College London UK
maria.kallia@kcl.ac.uk

Sue Sentance
King’s College London, UK
sue.sentance@kcl.ac.uk

ABSTRACT

With the introduction of the new computing curriculum in England, teachers are facing many challenges, among them the teaching of computer programming. Literature suggests that the conceptual understanding of this subject contributes to its difficulty and that threshold concepts, as a source of troublesome knowledge, have a significant role in this. This paper explores computing teachers’ perspectives on the Threshold Concept framework and suggests potential threshold concepts in the area of *Functions* and, more generally, in *Procedural Abstraction*. A study was conducted, using the Delphi method, including both computing teachers with experience teaching at upper secondary/high school and computing teachers with experience practicing programming in a professional environment for more than 7 years. The results indicate that the majority of the participants support that the Threshold Concept framework can explain students’ difficulties in programming and agreed on 11 potential threshold concepts in the area of *Functions* and *Procedural Abstraction*. The participants focused more on the troublesome characteristic of threshold concepts and less on the transformative and integrative. Most of the participants also specified that they would change the way they teach a concept if they knew that this is a threshold one. Finally, the paper discusses the findings and how these will shape our future research[1]

CCS CONCEPTS

• **Social and professional topics → CS1**; **K-12 education**;

KEYWORDS

Threshold Concepts, Computer Science Education, Computer Programming

1 INTRODUCTION

Recently there was a reformation in England’s school computing curriculum in primary and secondary education. A new national curriculum was introduced in 2014 which included Computing as a new and autonomous subject consisting of three elements: computer science, information technology and digital literacy [24]. The new curriculum introduced many challenges for teachers including the teaching of computing programming [48].

The study that is presented in this paper is a part of a larger research project which aims to provide guidance to computing teachers about students’ difficulties in programming and to make suggestions on how they can modify their teaching practices to help students overcome the corresponding learning obstacles. The literature suggests that in every discipline there are some concepts that pose extra difficulties to learners. These concepts, as Meyer and Land [36] advocate, are thresholds and their understanding is significant to master a discipline. A large amount of studies has been conducted with the aim to identify these concepts in different disciplines.

In computer programming, research studies focus in undergraduate level and identify threshold concepts in the broader area of computer programming. For example, Boustedt et al. [5] recognise that the thresholds they identified and especially *Object oriented programming* are broad areas inside which other thresholds exist. In light of this, the current study focuses on secondary education programming and in a specific thematic area – that of *Functions* and, more generally, *Procedural Abstraction*. This is an area in programming that research has identified as difficult for students, an area where misconceptions arise [6, 17, 32, 50], and a potential threshold concept itself [2]. Additionally, this is an area that is taught in secondary computing as well as in undergraduate computer programming courses. The study’s findings will be useful both for teachers in secondary computing education and for lecturers/professors at colleges and universities.

Specifically, the study endeavours to provide answers to the following research questions:

- What do computing teachers report about students’ difficulties in computer programming in upper secondary school (grades 9-12)? What are computing teachers’ perspectives on the usefulness of the Threshold Concept framework in secondary computer programming?
- What are the concepts that computing teachers agree as being threshold concepts in the area of *Functions* and *Procedural Abstraction*?

To answer these questions, the Delphi Method was employed. This method is known as a consensus method and is employed when the researcher seeks to achieve consensus among the participants. The panel that took part in this study was made up of 10 computing teachers; 6 of them were also practitioners with experience practicing programming outside school settings for more than 7 years. Three on-line questionnaires were sent individually to all the participants and an iteration process of design, distribution, data collection and analysis was used. The data was qualitatively and quantitatively analysed with the SPSS software.

The study contributes to the teaching framework of computer programming at schools. To the authors’ knowledge this is the first attempt to identify threshold concepts in computer programming at school level as well as the first attempt to identify threshold concepts in this field by employing a consensus method. Goldman

[2] *Procedural Abstraction* as a concept was among the 33 concepts suggested as threshold in the study of Boustedt et al. [5].

et al. [20] also employed the Delphi method but to identify the important and difficult concepts in three subjects, one of which was programming fundamentals.

Identifying threshold concepts could impact greatly on curriculum design, meaning that these concepts could become a central focus in the teaching of computer programming. Based on the study's findings, further research will be conducted to investigate in more depth how teachers understand the transformative and integrative characteristic of these concepts and explore students' difficulties when dealing with problems which incorporate these concepts. Conclusively, this study is the first step towards helping teachers design their courses and their practice around these concepts.

2 BACKGROUND

2.1 Threshold Concepts

In 2003, Eric Meyer and Ray Land introduced the notion of threshold concepts, concepts that present specific features and play a major role in the organisation of the education course. Meyer and Land [36, p. 1] defines threshold concepts as *"akin to a portal, opening up a new and previously inaccessible way of thinking about something. It represents a transformed way of understanding, or interpreting, or viewing something without which the learner cannot progress"*.

The definition of threshold concepts emphasises the new conceptual landscape which the learner crosses where initial notions and ideas, formerly not understood, come into sight. These are concepts whose understanding is necessary for mastering a discipline, but are also concepts that cause students' progress to grind to a halt. Once understood, threshold concepts lead to a qualitatively different way of seeing a discipline and the learning experience [28]. This is the basic characteristic that distinguishes a threshold concept from a core concept. Explicitly, Meyer and Land [36, p. 4] argue that the main difference between a core and a threshold concept is that the former is a *"conceptual building block"*, essential to be understood in order for progress in the course to be achieved, whereas threshold concepts' understanding leads to a qualitatively different way of seeing the subject. This indicates that a threshold concept diverges from other categories of concepts due to its transformative characteristic - an important change in the insights of a discipline - which core concepts lack.

The transformation that the learner experiences as a result of grasping a threshold concept may be abrupt or prolonged over an extensive period during which the learner will experience emotions of troublesomeness. This period is defined as liminality. Liminality is an insecure space where the learners fluctuate between old and new understandings and may find themselves in a state where understanding is mostly based on imitation and lack of authenticity [36]. As Cousin [7] aptly explains, liminality is a space just like the one that a child passes to move to adulthood.

2.2 Characteristics of Threshold Concepts

Not all concepts in a discipline can be categorised as threshold concepts. To be able to categorise a concept as a threshold, Meyer and Land [36] presented seven key features that may characterise a threshold concept:

- Transformative: once these concepts are understood, they evoke an important change in students' behaviour and in how students perceive their disciplines. Meyer and Land [36] also believe that once the concepts are understood a potential effect is to occur a transformation of learners' personal identity which involves a change in values and beliefs, emotions or attitudes.
- Integrative: threshold concepts not only lead to a transformation of one's perception and understanding in a discipline but also they can influence and change how other disciplines are viewed by integrating with the already existing knowledge [46]. In other words, they uncover the *"formerly unseen interrelatedness of something"* [36, p. 5].
- Bounded: this feature indicates that threshold concepts have borders that, when traversed, can lead to other conceptual developments [28]. Thus, these boundaries function as distinction points between subject areas, to define, as Meyer and Land [36] aptly comment, academic regions.
- Troublesome: threshold concepts can be puzzling and difficult to understand and consequently can be troublesome for students who engaged with these concepts.
- Irreversible: the shift on learners' view of the subject and the modification of their perspectives, that occurred by understanding a threshold concept, are implausible to be forgotten.
- Discursive: the change the learner undergoes crossing the threshold concept leads him/her to express himself/herself in more discipline-like ways [22]. In other words, crossing a threshold concept will improve the way the learner uses the discipline's language.
- Reconstitutive: this characteristic refers to the change of the learner's identity and subjectivity once the threshold concept is understood. It involves a reconfiguration of the learners previously formed schema (constructivism centred) which will cause an ontological and an epistemic shift [29].

2.3 Identifying threshold concepts

The Threshold Concept framework provides a fruitful area for developing teaching principles and guidelines to improve students' understandings and achievements in a discipline. The conceptual challenge that students face when confronted with demanding concepts in a subject, makes education researchers eager to explore what students need to understand within a subject in order to progress. Therefore, a considerable amount of research has been conducted involving threshold concepts in a variety of subjects and in computer programming [5, 15, 16, 26, 42, 44, 45, 49].

The review of the literature has demonstrated a rich repertoire of methods employed for identifying threshold concepts. A review conducted by Sanders et al. [45] provides a summary of the threshold concepts identified so far in Computing and the methods employed to identify them. Nevertheless, the identification process is difficult and requires time, reflection and debate. Barradel [2] suggests that it seems that in the first stages, when a preliminary list of threshold concepts is constructed, consensus is significant. For this reason, she recommends the use of the Nominal Group Technique (NGT) and the Delphi Technique which are both examples of consensus approaches and appropriate for exploring both individuals' and a group's points of view.

Despite the fact that substantial amount of research studies have been conducted to identify threshold concepts in various disciplines, the question yet remains on how researchers should approach the identification problem.

3 METHODOLOGY

Barradell [2] argues that a consensus technique is vital to the process of identifying threshold concepts. Endorsing this argument, the method that was chosen for collecting the data was the Delphi Method (both qualitative and quantitative form) and the participants were individuals currently employed as teachers.

3.1 Delphi Method - An Overview

The Delphi method is usually employed when the existing evidence on a topic of interest is limited or where the collective opinion of experts would be beneficial [25], especially in research where the topic under investigation is full of uncertainty [55]. The Delphi technique is mainly qualitative in nature but it can also provide quantitative evidence depending on how it will be applied [4].

The basic design of a Delphi technique includes bringing together groups of experts, not restricted in a geographical area, who will participate in a number of rounds answering to a specific question through e-mail [30]. After each round, the participants obtain feedback of the groups' responses to the previous round and the round procedure is repeated until consensus is reached [30]. According to Vernon [53] consensus usually ranges from 55% to 100% with the most frequent occurrence being 70%.

3.2 Delphi Designs

There are two main Delphi designs, the conventional Delphi and the modified Delphi [1]. In the conventional Delphi, the list of arguments, suggestions and generally the list of alternatives, are the product of the panel's initiatives. In the modified Delphi, the list of arguments or suggestions is provided by the researcher [1] having conducted a literature review upon the specific research subject. This study adapts the conventional Delphi design for two reasons: firstly, we wanted our participants to suggest the concepts that from their experience are potential threshold concepts, and secondly, there are currently no studies that suggest threshold concepts in the thematic area of functions and procedural abstraction that could have been used to generate the alternatives.

Specifically, in conventional designs, consensus is reached through repeated rounds in which the participants will have to answer open-ended questions [4, 27]. The first round begins with an open-ended question which is regarded as the foundation of soliciting the panel's information on a specific area [27]. The initial question is significant as it sets the underpinning for the whole process [10].

3.3 Characteristics of Delphi Method

There are three characteristics that are extremely important in the Delphi method: *anonymity, iteration* and *feedback* [1].

Anonymity is one of the main characteristics of this method. Some research studies advocate that collecting opinions separately is a more accurate procedure than collecting data from groups of people working together like focus groups [9]. In effect, Dalkey and Helmer [9, p. 459] argue that this method seems *"to be more conducive to independent thought"*. This is because, working individually, gives the opportunity to participants to express themselves freely without the influence of others people opinions [4, 27] and also assures confidentiality between the participants and the researcher. This is why in this technique all the participants should communicate individually with the researcher and vice versa.

The second characteristic is the *iteration process*. In a Delphi process, there is not a fixed number of rounds, although most studies usually apply a three-round process. The first is usually exploratory in which the members are asked to provide suggestions or other responses to a question [54]. The following rounds usually include quantitative ratings and/or qualitative data. The researcher should establish a criterion for when the iteration process should finish and usually this is the consensus among the members of the panel [54].

The third characteristic which is critical in this method is *feedback*. In each round, the researcher collects and analyses the data and return them to the panel. The members are then asked to review the results of the previous round and designate if they agree or not with explaining why they propose changes. Hsu and Sandford [27] point out that feedback gives the opportunity for reassessing initial stand points and altering them if required. This process is repeated until consensus is achieved.

3.4 The role of the researcher

In the Delphi method, the researcher acts as a coordinator and not a contributor and has a double role: the planner and the facilitator [1]. In the planning phase, the researcher is responsible for identifying the members that will take part in the study, the number that will be sufficient and the specialty of each of the participants. The issue of who qualifies as an expert to contribute in the research is of critical significance. Avella [1] suggests that the researcher should ask himself/herself of which groups have an interest in reaching the study's purpose. He further argues that the criteria for defining the experts should include those determining characteristics that each group would recognize as those crucial to expertise. Having decided on the groups that will participate in the panel, the second decision relates to the size of the panel. There is not a standard number or any recommendations about the size of the Delphi's panel. Bourgeois et al. [4] argue that the panel size should preferably be of around 10 to 18 individuals.

As a facilitator, the researcher's job concentrates more on him/her taking control of the arguments presented in the panel. The researcher should ensure that the debate and arguments presented in each round are independent of the participants' reputation and that all opinions and arguments receive equivalent weight.

3.5 Strengths and Weaknesses

The most important aspect of the Delphi method is that of consensus, especially when the situation under investigations is ambiguous [40]. At the same time, this design is easier to execute than other consensus techniques like the Nominal Group Technique. In the latter, the researcher invites the participants in a common place,

Table 1: Participants' Characteristics

	Less than a year or none	1-3 years	4-6 years	More than 7 years
Years of teaching experience	0	10%	0	90%
Years of teaching experience at key stage 4 and/or 5	0	10%	10%	80%
Professional experience in computer programming	10%	10%	20%	60%

on an agreed time that is convenient for all. In Delphi, there are no such limitations as each participant responds to the questionnaire at his/her own pace and place. Additionally, due to the anonymity, the participants feel free to express their opinions without thinking of potential criticism [12] and so this design supports the freedom of expression. However, there are some problems with anonymity which are described in the following paragraph. Finally, the very nature of this technique assures an ease of communication.

One among the first potential flaws of this research design is the researcher's bias. To this effect, Linstone and Turoff [30] argue that the researcher may even accidentally influence the process due to its authority duties. For example, issues like who is to participate in the panel group and how the research questions are framed are sources that the researcher may put his/her own perspectives and stances. That is why, Avella [1] supports that the best way to prevent this situation is to have an external reviewer to investigate the questions in order to ensure that the questions as formulated do not lead to specific responses. The second author of this paper along with an external reviewer ensured that the above problems where not an issue for our study. Additionally, a potential flaw of this design is the researcher's preconceptions which might influence the way that the researcher interprets the participants' responses and how he/she transmits them in the participants for the next round [12]. For this reason, we were very careful in our individual communication with each of the participants as we realised soon that this was a potential peril for our study. As such, we were vigilant not to expose our perspectives nor constrained the production of the alternatives positions, but to trust the expertise of the members.

The anonymity factor could tempt the participants to pay less attention and not to be fully motivated to the purpose of the research [35]. This, could potentially lead to less valid and rigorous contributions. In our study, there were cases where the participants provided an insufficient explanation of their suggestions. This is a weakness of the Delhpi process and one of the reasons we have decided to further interview our teachers to get a deeper understanding of this issue. Sackman [43] also contends that anonymity is interpreted as lack of accountability and, thus, the participants can say anything. Additional problems stem from the participants' right to drop out at any state of the design. Hsu and Sandford [27] argue that due to its iteration process, the Delphi methods is vulnerable to low response rates. That is why it is important to accurately select participants that have also a personal motivation to take part in the study [31]. Indeed, during the iteration process, it was clear to us that some of the participants lost their interest in contributing to the study. One of the reasons is that it takes time to collect all the questionnaires back, analyse the responses, design the new questionnaire and distribute it again for the new round.

4 DATA COLLECTION

4.1 The Participants

The selection of the participants is an important aspect of the Delphi method. Habibi et al. [21] explain that the validity of the results is linked to the participants' knowledge and capability. Additionally, the participants should have a personal interest for the study [31] because of Delphi's high drop out rates between the rounds.

For this reason, the basic characteristic that describes the participants is that they are all computing teachers or previously employed as computing teachers in secondary education and have experience teaching at key stage 4 and/or 5[3]. Another important characteristic is that 60% of the participants have or had experience practicing programming in a professional level. Including individuals with such an experience would provide the study with a variety of opinions. Table 1 depicts in detail the participants' teaching and professional background information.

We have also based our decision of exploring teachers' opinions on the criticism and suggestions of other researchers in the field. Male and Ballie [33, p. 252] argue that to identify threshold concepts the most appropriate source is to collect data either directly from students or from *"people whose experiences give them awareness of students' experiences"*. They further argue that teachers can identify not only concepts that are troublesome but concepts that are transformative for students which is of particular importance to this research. Additionally, Shinners-Kennedy and Fincher [49] advocate interviewing teachers to identify threshold concepts. They note that *"there, after all, is where the reality of student learning is lodged, in the day-to-day classroom experience"* [49, p. 14]. They also state that the identification of threshold concepts needs both pedagogical and content knowledge on behalf of the interviewees. For this reason, Zwaneveld et al. [56] contend that this can be found in secondary teachers rather than university teachers. They further advocate an approach that employs both teachers and students for this process.

Taking these points into consideration, we first started our investigation by asking experienced secondary computing teachers to suggest potential threshold concepts. However, because we cannot see this exploration without the involvement of students, our future research plans involve both teachers' and students' semi-structured interviews as a complementary approach to this one.

[3]Key stage 4 is equivalent to grades 9 & 10 and Key stage 5 to grades 11 & 12

In total, 13 individuals initially volunteered to participate in the study. None of them had previously made known to the authors his/her thoughts and perspectives upon the topic under research and none of the participants had any kind of relationship with the authors. This is an important prerequisite of Delphi studies. Specifically, an issue that the researcher should consider for the selection of the participants is the potential bias, that is, the positions held by the participants may be acknowledged by the researcher. Hasson et al. [23] argue that impartiality must be a priority in the Delphi method and for this reason, Murphy et al. [38] suggest that participants with previous relationship with the researcher should be excluded.

4.2 The study

The present study was conducted from 14th March 2017 to 1st of May 2017 and included the following phases:

4.2.1 Before the Delphi rounds. Three days before the start of Round 1, the participants were asked to read a short literature review on threshold concepts created by the authors and were encouraged to further investigate the topic by reading some papers on threshold concepts such as Meyer's and Land's articles (e.g. [36]). The information sheet included information about the study and the time-line for each upcoming round of Delphi. The participants were asked to complete the consent form and return it to the authors along with their Round 1 responses.

4.2.2 Delphi - Round 1. The first questionnaire was distributed separately to each of the participants by email and a week was given to complete and return it. This questionnaire consisted of three parts: the first part included questions about the participants' education and professional background, the second part included questions about the participants' perspectives on the Threshold Concept framework and on students' difficulties in programming and the third part included a task which asked participants to consider and propose threshold concepts in the area of *Functions* and *Procedural Abstraction*, providing a short description for each concept. The teachers were not asked to reflect on their own experience of when they learnt how to program but on their practice as teachers and what they believe their students experience as thresholds. The Round 1 questionnaire was sent to 13 participants who had initially agreed to participate in the study but 10 of them returned the questionnaire.

As soon as the panelists returned the questionnaires, the authors started the analysis of the participants' responses. Responses on part one and two were quantitatively analysed with the SPSS software while responses on the third part were qualitatively analysed. The aim of the latter analysis was the generation of a list with all the potential threshold concepts proposed by the panel.

4.2.3 Delphi - Round 2. The second questionnaire included all the potential threshold concepts suggested in the previous round along with the descriptions provided by the participants. The order of the concepts was organised alphabetically. The questionnaire also included a task that asked participants to indicate their level of agreement of whether a concept is a potential threshold concept. A five point Likert scale was used with items ranging from strongly disagree to strongly agree. The second questionnaire was again distributed by email individually to each participant who were given one week to complete and return it. All participants returned the questionnaire but more time was needed; so this phase lasted almost three weeks.

The analysis of the panelists' responses to the second questionnaire was a quantitative one. The purpose of this round was to investigate if consensus (whether a concept is or not a potential threshold concept) has been reached for some concepts. As soon as the quantitative analysis was concluded, two lists were created: the first list included the concepts for which consensus has been reached and the second one the remaining concepts. Each list also included all the statistical information calculated in order to provide the participants with the corresponding feedback.

4.2.4 Delphi - Round 3. The third questionnaire administered included the aforementioned two lists and the statistical information generated and asked the participants to review the information provided. In this stage the participants were also asked if they wanted to change their previous responses based on the information provided for each concept. For this purpose, the questionnaire included information on the participants' previous responses for each concept and asked the participants to add a new level of agreement if they wanted to change their response and explain the reason of the change.

All the participants returned the questionnaires and the analysis of the new responses was also quantitative, including the same statistical information as the previous round. However, according to Dajani et al. [8], to decide if the Delphi process can be terminated, stability must be reached for each concept. If stability is not achieved, then another round is needed. In our study, stability had been achieved in this round so the Delphi was terminated. As a final step, all the statistical information were disseminated to the participants.

5 DATA ANALYSIS AND RESULTS

This section describes the participants' responses in each round of the Delphi process. In total, three rounds were conducted until stability and consensus were achieved.

5.1 Delphi - Round 1

The first questionnaire distributed included three sections: the participants' education and professional information depicted in Table 1, the participants' perceptions of the Threshold Concept framework and suggestions of potential threshold concepts.

5.1.1 Perceptions of the threshold concept framework. The second section of the first questionnaire included questions about the panelists' perspectives on the Threshold Concept framework and on students' difficulties in programming. The results of each question are depicted in Table 2.

Questions 1 to 3 refer directly to threshold concepts and it seems that most of the participants were familiar with this framework. Most of them also indicated that threshold concepts can explain students' difficulties in programming and that they would change the way they approach and teach a concept if they knew that it is a threshold one. Questions 5 to 7 refer to students' difficulties and whether these are accumulated on the theoretical/conceptual

Table 2: Teachers' perceptions on Threshold Concepts Framework

Question	Yes	No	I am not sure
1. Are you familiar with the framework of threshold concepts?	90%	10%	0%
2. Do you believe that the framework of threshold concepts can explain some of the students' difficulties in computer programming?	80%	0%	20%
3. Would you change the way you approach a concept or a construct if you knew that it was a threshold concept?	70%	10%	20%
4. Do you think more research is needed in how core concepts or constructs should be approached in teaching computer programming at KS4 and KS5?	100%	0%	0%
5. Do you think that some students can effectively write code even though they haven't really grasped the theory behind the concepts they employ?	40%	50%	10%
6. Do you think that some students have difficulties in applying their understanding in programming tasks even though they have a theoretical understanding of the corresponding concepts and constructs?	100%	0%	0%
7. Do you think that students encounter more difficulties in applying their knowledge in programming tasks or in understanding concepts and how constructs work?	(Applying)30%	(Understanding)20%	(Both)50%

understanding of programming or/and to the application of theory in practical problems. In particular, question 5 concentrates the most divergent views. 40% of the teachers support that students can write workable code even though they have not grasped the theory while a 50% of teachers support the opposite. However, there was unanimous agreement that the understanding of the theoretical framework of programming does not indicate that students' won't experience difficulties when they encounter practical problems. Finally, question 4 is concerned with whether teachers feel that more research is needed on the didactics of programming. Teachers unanimously agreed that more research is needed on how to approach and teach core concepts and constructs in computer programming.

5.1.2 Potential threshold concepts. The third section of the first questionnaire referred to threshold concepts and asked the participants the following question: *"Based on your experience, can you suggest three or more potential threshold concepts in the broad area of functions and procedural abstraction? Please provide a short description for each of the threshold concept you suggest."*

Round 1 resulted in 27 suggestions which were qualitatively analysed. The aim of the analysis was the generation of a list with the potential threshold concepts proposed by the panel. For each suggested concept, a suitable entry was made in the list. However, there were some cases that were more challenging than others. For example, some participants provided only descriptions of the concept rather than single words. These created two problems: Firstly, many of these descriptions matched another concept already proposed by another participant. Thus, we had to put together concepts that were articulated slightly differently but covering the same theme. For example, a participant used the following sentence to describe a potential threshold concept: *"students find difficult to understand that the code jump around and not executed sequentially"*. This description was matched to the concept of *Control Flow* which was suggested by another participant. We only grouped together the participants' responses referring exactly to the same computing concept. Similar concepts like arguments and parameters were not grouped together as they have a slightly different definition in the curriculum. Secondly, some participants included more than one potential threshold concept in one description. For example, the following phrase was used by one participant which led us to create two potential threshold concepts:

"The use of argument passing and return values to reduce the need for global variables. Students initially struggle to understand the benefit of local variables over global, particularly as they may perceive the use of argument passing and return values more time consuming and challenging than simply using global variables."

The list was distributed to the participants to initiate Round 2. Using the description provided for each of these concepts, the authors pieced together a list of 19 potential thresholds.

5.2 Delphi - Round 2

With the list of 19 concepts, the participants were asked to indicate the level of agreement on a five-point Likert scale of whether each concept is a potential threshold one. For each concept, the following statistical information was calculated: mean, median, mode, SD, IQR, and the percentage of *"agree"* and *"strongly agree"* responses. To reach consensus the following criteria were considered for each concept:

- The percentage of the participants stated that *"agree"* or *"strongly agree"* that a concept is a threshold must be more than 75%
- The standard deviation (SD) should be less than 1.5
- The Interquartile Range (IQR) should be less than or equal to 1

These criteria were adapted from the study by Giannarou and Zervas [19]. It should be noted that 75% agreement level was chosen to increase the confidence in the study's results and is one of the strictest in the literature[4].

The statistical analysis followed resulted in 10 concepts reaching consensus of being threshold concepts. Table 3 depicts these concepts.

[4]For example, Giannarou and Zervas [19] adopted a 51% of agreement level.

Table 3: Concepts that reached consensus on Round 2

Concept	Mean	Median	Mode	% (agree and strongly agree)	SD	IQR
Arguments	4.20	4	4	90%	0.919	1
Calling a function	4.30	4.50	4 and 5	90%	0.949	1
Control flow	4.20	4	4 and 5	80%	0.789	1
Parameters	4.20	4	4	90%	0.919	1
Parameters passing	4.20	4	4	90%	0.632	1
Procedural Decomposition and Design	3.90	4	4	90%	1.101	0
Recursion	4.40	4.50	5	90%	0.699	1
Return values	4.30	4	4	100%	0.483	1
Variable	4	4	4 and 5	80%	1.247	1
Variable scope	4.40	4	4	100%	0.516	1

Table 4: Concepts that reached consensus on Round 3

Concept	Mean	Median	Mode	% (agree and strongly agree)	SD	IQR	Wilcoxon Signed-Ranks Test: p value
Arguments	4.40	4	4	100%	0.516	1	.317
Calling a function	4.30	4.50	4 and 5	90%	0.949	1	1
Control flow	4.20	4	4 and 5	80%	0.789	1	1
Parameters	4.20	4	4	90%	0.919	1	1
Parameters passing	4.20	4	4	90%	0.632	1	1
Procedural Decomposition and Design	3.90	4	4	90%	1.101	0	1
Recursion	4.40	4.50	5	90%	0.699	1	1
Return values	4.30	4	4	100%	0.483	1	1
Variable	4	4	4 and 5	80%	1.247	1	1
Variable scope	4.40	4	4	100%	0.516	1	1
Abstraction	4	4	4	80%	0.667	1	.317

5.3 Delphi - Round 3

In this round, the participants were again asked to review the results of the previous round and indicate a new level of agreement if they wanted to change their previous one. Three participants in total made changes to their previous level of agreement. These changes led to another concept reaching consensus of being a threshold one, that of *Abstraction*, making the total number of threshold concepts 11.

Dajani et al. [8] argue that to terminate the Delphi process, stability must be reached for each argument. Thus, to measure stability for each concept the Wilcoxon signed-ranks test was employed as suggested by the literature ([47], [11]). By employing this test, researchers can determine whether a difference between the data of two Delphi rounds has statistical significance thereby testing for stability of the data.

All concepts reached stability (the significance level was set at .05) and thus the Delphi was terminated with three rounds. Table 4 depicts the concepts that have reached consensus in this round. The p value of the Wilcoxon signed-ranks test is also provided for each concept. Changes in some values in comparison with Table 3 indicate a change in the participants' opinion.

6 DISCUSSION

Studies of computer programming and threshold concepts have been conducted only at undergraduate level and, thus, this study is the first attempt to identify threshold concepts in secondary computer programming. This section discusses the study's findings under the theoretical framework of threshold concepts. The first part refers to our first research question and discusses teachers' perspectives on this framework and any pedagogical implications and future research directions that arise from these. The second part refers to the second research question and discusses the 11 concepts suggested as potential threshold concepts from the participants; the discussion is based on the participants' responses as well as relevant previous research studies in this thematic area.

6.1 Teachers' Perspectives on the framework of Threshold Concepts

Examining the teachers' responses depicted in Table 2, it can be argued that the framework of Threshold Concepts influences and can further inspire secondary teachers' practices: Most of the participants indicated that students' difficulties in programming can be examined and explained under the view of threshold concepts. Most importantly, a high percentage of them reported that knowing that a concept is a threshold one would influence their approach in teaching this concept. This was quite a surprising result considering that there are no practical guidelines to date on how threshold concepts should be approached in teaching. However, it would be interesting to further investigate teachers' views on this matter and how they would change their teaching methods to teach a threshold concept.

Searching for the source of computer programming difficulty is not a new endeavour. Since the mid '80s many researchers centred their attention on identifying students' misconceptions. For example, Perkin and Martin [39] identified four types of fragile knowledge relevant to computer programming: *Partial Knowledge, Inert Knowledge, Misplaced Knowledge, and Conglomerated knowledge.* Du Boulay [14] also identified specific areas that students find difficult and grouped these areas into categories: *Orientation, Notional Machine, Notation, Structures, and Pragmatics.* In this study, we focused on two broader categories of knowledge: theoretical and practical knowledge which on the threshold framework can be seen as a distinction between threshold concepts and threshold skills as proposed by Thomas et al. [52] and Sanders et al. [44]. Teachers' views on the relationship between the theoretical and practical understanding do not converge. Their responses surely indicate that the theoretical understanding of programming is not enough to ensure that students will not experience difficulties while writing programs. On the other hand, their perspectives of whether the partial or lack of theoretical understanding obstructs students' practical knowledge are almost dichotomous, highlighting the difference between the theoretical knowledge students need to understand and the skills they need to practice. From the one hand, it stands to reason that the theoretical understanding must precede students' engagement with practice. However, there are research studies that indicate that students can write workable programs even though the misconceptions they hold have not been resolved. An illustration of such a study is the one conducted by Madison and Gifford [32].

Concluding this section, it was not a surprise to see that the participants unanimously agreed that more research is needed on the didactics of programming. Computer programming is a relatively new area in England's secondary computing curriculum and teachers feel that more support is needed in teaching this subject. This is in line with the research conducted by Sentance and Czizmadia [48] in the UK and with initiatives that support computing teachers and provide training opportunities such as Computing at School (UK) and CS For All (US).

6.2 Teachers' suggestions on potential threshold concepts within Functions and Procedural Abstraction

Many years of research have revealed that students experience many challenges and misconceptions when they start using functions and/or procedures. That is why most of the concepts identified in this study as threshold concepts do not come as a surprise to a reader familiar with relevant research.

Fleury [17] and Madison and Gifford [32] reported that the concepts of *arguments, parameters, and parameter passing* are concepts that cause trouble in students' understanding. In her paper, Fleury [17] discusses constructed rules (false assumptions) that lead to students' misconceptions around this area. Among them, she referred to *parameters values* and *variable scope.* Madison and Gifford [32] with their study on *parameters passing* indicate that students can write workable programs despite their existing misconceptions. They also refer to *parameter passing* as the most challenging concept in the introductory programming. More recent studies also highlight the problems students' encounter with actual and formal parameters and the order of parameter evaluation and a function call [6, 50] while Miller et al. [37] suggest that *parameter passing* is a potential threshold concept. Under the lens of our theoretical framework, this study categorises these concepts as potential threshold concepts. The participants, in their descriptions, focused mostly on the conceptual difficulty that these concepts impose on students, emphasising their troublesome characteristic. However, there were cases where the participants grouped together these concepts highlighting the connection between them and, thus, providing evidence of their integrative characteristic. For instance, one participant referred to *parameter passing* as: *"passing values into the parameters of the procedures and functions is often hard for students to get to grips with, particularly when passing variables as arguments".* Other participants linked the concept of *parameter passing* with *variable scope* and *return values*: *"Students initially struggle to understand the benefit of local variables over global, particularly as they may perceive the use of argument passing and return values more time consuming and challenging than simply using global variables"* and *"students find difficult to understand the use of parameter passing instead of global variables".* These examples are evidence of the relationship of these concepts and the possible transformation that a student exhibits once these concepts are understood. A participant's description aptly summarises this: *"These concepts are the cornerstone of understanding how functions work".*

The order with which functions or procedures are executed has also been reported in the literature as a source of difficulty for students. For instance, Sleeman et al. [51] reported that students regard that the execution of functions begins when the program starts. Ragonis and Ben-Ari [41] also reported that students believe that the order of the methods in the program determines the order they are called. Sirkia [50] found that students misunderstood how a function call works reporting on several problems students experience. In this study, the participants perceived *calling a function* and *control flow* as threshold concepts. Most of the participants again focused on the troublesome characteristic. For example, one participant stated that *"The first Threshold concept for students to understand is that the function will not do anything until it is called. Most students struggle to understand why it will not run when you have defined it. They assume that once they have defined a function and compiled it, that the code will execute".* Another participant reported that *control flow* and *calling a function* are interrelated concepts and once one of them is understood the other becomes clearer, highlighting in this way their integrative characteristic.

Another area in which the literature has shown a variety of students' misconceptions is the notion of *variables* in programming. Du Boulay [14] was among the first to highlight that the analogies made to help students understand an aspect of programming are sources of misconceptions. It is undoubtedly a very important concept in computer programming and understanding is imperative to progress in the discipline. To the authors' knowledge this is the first study that proposes *variables* as a threshold concept. The troublesome characteristic was evident in the participants' descriptions: *"students can struggle with the fact that variables can store values that can be referenced in other parts of the program and that the value stored can change"* while another participant highlighted its significance: *"I think this is a very important concept that goes beyond procedural abstraction"*. These examples emphasise the importance of understanding *variables* and indicate the impact that the misunderstanding may have on students' progress in the discipline. This concept was also mentioned many times in the participants' descriptions of other concepts, especially, the concepts of *variable scope* and *return values*. This is probably because the concept of *variable* is a fundamental concept that it is connected with others across programming and conceptually involves as students move forward in the discipline.

Gal-Ezer and Harel [18] advocate *recursion* as one of the most challenging concepts for teaching. Using a phenomenographic approach, Booth [3] recognised three different ways that students' experience recursion: as a programming construct, as a means for repetition, and as a self-reference. Many studies put recursion in their focus and a good summary of them is described in the study of McCauley et al [34]. In their paper, they summarise years of findings about students' challenges, mental models developed when practising recursion, and best practices for teaching this concept. Due to its compound difficulty, some studies have already added this concept to the threshold concept repository [5, 26, 42]. Along the same lines, *Abstraction* as a threshold concept has already been proposed by some researchers [5, 16]. It is indeed a core concept and as Detienne [13] contends, a necessary ability for object oriented programming.

Finally, *Procedural Decomposition and Design* is a concept that has not been reported in the literature as either a source of misconceptions or as a threshold concept. However, in the study of Goldman et al. [20], procedural design and functional decomposition was among the 10 concepts with the highest rankings of being important and difficult in programming. Undoubtedly, students face difficulties in understanding how to break a problem into smaller parts and then create the corresponding procedures or functions. A recent study conducted by Thomas et al [52] suggests software design as a threshold skill. In the same line, it seems more accurate to regard *Procedural Decomposition and Design* as a threshold skill than a threshold concept. This argument was also suggested by one of the participants that had initially suggested this concept: *"I think that this is more a threshold skill than a threshold concept"* as well as indirectly by another participant, noting his/her disagreement about *Procedural Decomposition and Design* being a threshold concept: *"I was at odds with the group on Decomposition and Design. For me, this is a complex activity that relies on a range of knowledge, experience and skills."*. Further research will be conducted to explore *Procedural Decomposition and Design* under the lens of threshold skills.

Comparing the results of our study with the studies conducted in higher education, our findings suggest some new potential threshold concepts in the area of functions while also corroborate the findings of other researchers regarding the concepts of Parameter Passing [37], Abstraction [5, 16], and Recursion [5, 26, 42].

7 CONCLUSION AND FUTURE WORK

This paper discusses some of the difficulties students' face in secondary computing education based on teachers' perspectives. Our analysis is grounded on the framework of Threshold Concepts; both teachers' views on its suitability in secondary computing and teachers' suggestions on potential threshold concepts in the area of *Functions* and *Procedural Abstraction* are explored. In this respect, teachers support that students' difficulties in programming can be explained under the spectrum of threshold concepts and further indicate that knowing that a concept is a threshold could have an impact on their teaching.

To identify threshold concepts, a three-round Delphi approach was employed until both consensus and stability were achieved for each proposed concept. To the authors' knowledge, this is the first study that employs a consensus technique to identify threshold concepts in programming and in secondary education. The concepts that reached consensus of being potential thresholds are the following: *Arguments, Calling a function, Control Flow, Parameters, Parameter passing, Procedural Decomposition and Design, Recursion, Return Values, Variable, Variable Scope, and Abstraction*.

This study found that most participants based their suggestions on the troublesome characteristic and less on the transformative and integrative ones. We also acknowledged that it is unlikely that all the concepts proposed are threshold concepts. For this reason, further research will be conducted with teachers to investigate in more depth how the concepts suggested incorporate the other characteristics of threshold concepts. Finally, we intend to investigate further if *Procedural Decomposition and Design* is regarded by computing teachers as a threshold skill rather than a threshold concept.

8 ACKNOWLEDGEMENTS

The first author acknowledges the support of Google which has enabled this work.

REFERENCES

[1] J.R Avella. 2016. Delphi panels: Research design, procedures, advantages, and challenges. *International Journal of Doctoral Studies* 11, 1 (2016), 305–321.
[2] S. Barradell. 2013. The identification of threshold concepts: A review of theoretical complexities and methodological challenges. *International Journal of Doctoral Studies* 65, 2 (2013), 256–276.
[3] S. Booth. 1993. The experience of learning to program. Example: Recursion. In *Fifth Annual Psychology of Programming Interest Group workshop (ICSE '16)*. Paris:INRIA, 122–145.
[4] J. Bourgeois, L. Pugmire, K. Stevenson, N. Swanson, and B. Swanson. 2009. The Delphi method: A qualitative means to a better future. (2009). http://www.freequality.org/documents/knowledge/delphimethod.pdf Retrieved March 15, 2017.
[5] J. Boustedt, A. Eckerdal, R. McCartney, J.E. Moström, M. Ratcliffe, K. Sanders, and C. Zander. 2007. Threshold Concepts in Computer Science: Do They Exist and Are They Useful? *(SIGCSE '07)*. ACM, New York, NY, USA, 504–508. https://doi.org/10.1145/1227310.1227482
[6] C.L. Chen, S.Y. Cheng, and J.M.C. Lin. 2010. A study of misconceptions and missing conceptions of novice Java programmers. In *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS' 12)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 84–89.

[7] G. Cousin. 2010. Neither teacher-centred nor student-centred: threshold concepts and research partnerships. *Journal of Learning Development in Higher Education* 2 (2010), 1–9.

[8] J.S. Dajani, M.Z. Sincoff, and W.K. Talley. 1979. The Delphi stability and agreement criteria. *Technological Forecasting and Social Change* 13, 1 (1979), 81–90.

[9] N.C. Dalkey and O. Helmer. 1963. An experimental application of the Delphi Method to the use of experts. *Management Science* 9, 3 (1963), 458–467.

[10] P.L. Davidson. 2013. The Delphi technique in doctoral research: considerations and rationale. *Review of Higher Education and Self-Learning* 6, 22 (2013), 53 – 65.

[11] E. De Vet, J. Brug, J. De Nooijer, A. Dijkstra, and N.K De Vries. 2005. Determinants of forward stage transitions: a Delphi study. *Health Education Research* 20, 2 (2005), 195 – 205.

[12] M. de Villiers, P. de Villiers, and A. Kent. 2005. The Delphi technique in health sciences education research. *Medical Teacher* 27, 7 (2005), 639–643.

[13] F. Detienne. 1997. Assessing the cognitive consequenses of the object-oriented approach: a survey of empirical research on object-oriented design by individuals and teams. *Interacting with Computer* 9, 1 (1997), 47–72.

[14] B. Du Boulay. 1986. Some Difficulties of Learning to Program. *Journal of Educational Computing Research* 2, 1 (1986), 57–73.

[15] A. Eckerdal. 2009. *Novice Programming Students' Learning of Concepts and Practice.* Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology, Uppsala: Acta Universitatis. Ph.D. Dissertation.

[16] A. Eckerdal, R. McCartney, J.E. Moström, M. Ratcliffe, K. Sanders, and C. Zander. 2006. Putting Threshold Concepts into Context in Computer Science Education. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITICSE '06).* ACM, Bologna, Italy, 103–107. https://doi.org/10.1145/1140124.1140154

[17] A. Fleury. 1991. Parameter passing: the rules the students construct. *ACM SIGCSE Bulletin* 23, 1 (1991), 283–286.

[18] J. Gal-Ezer and D. Harel. 1998. What (else) Should CS Educators Know? *"Communications of the ACM"* 41, 9 (sep 1998), 77–84. https://doi.org/10.1145/285070.285085

[19] L. Giannarou and E. Zervas. 2014. Using Delphi technique to build consensus in practice. *International Journal of Business Science and Applied Management* 9, 2 (2014), 65–82.

[20] K. Goldman, P. Gross, C. Heeren, G. Herman, L. Kaczmarczyk, M.C. Loui, and C. Zilles. 2008. 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 (SIGCSE '08).* ACM, Portland, OR, USA, 256–260. https://doi.org/10.1145/1352135.1352226

[21] A. Habibi, A. Sarafrazi, and S. Izadyar. 2014. Delphi technique theoretical framework in qualitative research. *The International Journal of Engineering and Science* 3, 4 (2014), 8–13.

[22] B. Harrison, P.H. Clayton, and G.A. Tilley-Lubbs. 2014. Troublesome Knowledge, Troubling Experience: An Inquiry into Faculty Learning in Service-Learning. *Michigan Journal of Community Service Learning* 20, 2 (2014), 5–18.

[23] F. Hasson, S. Keeney, and H. McKenna. 2000. Research guidelines for the Delphi survey technique. *Journal of Advanced Nursing* 32, 4 (2000), 1008–1015.

[24] F. Heintz, L. Mannila, and T. Färnqvist. 2016. A review of models for introducing computational thinking, computer science and computing in K-12 education. In *Frontiers in Education Conference (FIE '16).* IEEE, Eire, PA, USA, 1–9. https://doi.org/10.1109/FIE.2016.7757410

[25] G. Hejblum, V. Ioos, J. Vibert, P. Böelle, C. Chalumeau-Lemoine, C. Chouaid, A. Valleron, and B. Guidet. 2008. A web-based Delphi study on the indications of chest radiographs for patients in ICUs. *Chest Journal* 133, 5 (2008), 1107–1112.

[26] M. Holloway, E. Alpay, and A. Bull. 2010. A quantitative approach to identifying threshold concepts in engineering education. In *Engineering Education 2010 (EE2010) Inspiring the next generation of engineers.* Higher Education Academy Engineering Subject Centre, Loughborough, UK.

[27] C.C. Hsu and B.A. Sandford. 2006. The Delphi Technique: Making Sense Of Consensus. *Practical Assessment, Research and Evaluation* 12, 10 (2006), 1–8.

[28] M. Kiley and G. Wisker. 2009. Threshold concepts in research education and evidence of Threshold crossing. *Higher Education Research and Development* 28, 4 (2009), 431–441.

[29] R. Land, J. Rattray, and P. Vivian. 2014. Learning in the liminal space: a semiotic approach to threshold concepts. *Higher Education* 67, 2 (2014), 199–217.

[30] H.A Linstone and M. Turoff. 1975. *The Delphi method: Techniques and applications.* Vol. 29. Addison-Wesley Reading, MA.

[31] B.G. Ludwig. 1994. *Internationalizing Extension: An Exploration of the Characteristics Evident in a State University Extension System that Achieves Internationalization.* Ohio State University.

[32] S. Madison and J. Gifford. 1997. *Parameter passing: The conceptions novices construct.* Technical Report. https://eric.ed.gov/?id=ED406211

[33] S.A Male and C.A Baillie. 2011. Engineering Threshold Concepts. In *Proceedings of SEFI Annual Conference.* Lisbon, Portugal, 251–257. http://www.sefi.be/wp-content/papers2011/T7/24.pdf

[34] R. McCauley, S. Grissom, S. Fitzgerald, and L. Murphy. 2015. Teaching and learning recursive programming: a review of the research literature. *Computer Science Education* 25, 1 (2015), 37–66.

[35] H.P. McKenna. 1994. The Delphi technique: a worthwhile approach to nursing. *Journal of Advanced Nursing* 19, 6 (1994), 1221–5.

[36] J.H.F Meyer and R. Land. 2003. Threshold concepts and troublesome knowledge: linkages to ways of thinking and practicing. In *C. Rust (Ed.), Improving Student Learning-Ten Years On.* Oxford, OCSLD.

[37] A. Miller, Settle and J. Lalor. 2015. *Learning object-oriented programming in python: Toward an inventory of diculties and testing pitfalls.* Technical Report. School of Computing, DePaul University.

[38] M. Murphy, N. Black, D. Lamping, C. McKee, C. Sanderson, J. Askham, and T. Marteau. 1998. Consensus development methods, and their use in clinical guideline development. *Health Technology Assessment* 2, 3 (1998), 1–88.

[39] D. N. Perkins and Fay Martin. 1986. Fragile Knowledge and Neglected Strategies in Novice Programmers. In *Papers Presented at the First Workshop on Empirical Studies of Programmers on Empirical Studies of Programmers.* Ablex Publishing Corp., Norwood, NJ, USA, 213–229. http://dl.acm.org/citation.cfm?id=21842.28896

[40] C. Powell. 2003. The Delphi technique: Myths and realities. *Journal of Advanced Nursing* 41, 4 (2003), 376–382.

[41] N. Ragonis and M. Ben-Ari. 2005. A long-term investigation of the comprehension of OOP concepts by novices. *Computer Science Education* 15, 3 (2005), 203–221.

[42] J. Rountree and N. Rountree. 2009. Issues Regarding Threshold Concepts in Computer Science. In *Proceedings of the Eleventh Australasian Conference on Computing Education - Volume 95 (ACE '09).* Australian Computer Society, Inc., Wellington, New Zealand, 139–146. http://dl.acm.org/citation.cfm?id=1862712.1862733

[43] Harold Sackman. 1974. *Delphi assessment: Expert opinion, forecasting, and group process.* Technical Report. DTIC Document.

[44] K. Sanders, J. Boustedt, A. Eckerdal, R. McCartney, J.E Moström, L. Thomas, and C. Zander. 2012. Threshold Concepts and Threshold Skills in Computing. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research (ICER '12).* ACM, Auckland, New Zealand, 23–30. https://doi.org/10.1145/2361276.2361283

[45] K. Sanders and R. McCartney. 2016. Threshold Concepts in Computing: Past, Present, and Future. In *Proceedings of the 16th Koli Calling International Conference on Computing Education Research (Koli Calling '16).* ACM, Koli, Finland, 91–100. https://doi.org/10.1145/2999541.2999546

[46] O.J. Sandri. 2013. Threshold concepts, systems and learning for sustainability. *Environmental Education Research* 19, 6 (2013), 810–822.

[47] E. Seagle and M. Iverson. 2002. Characteristics of the turfgrass industry in 2020: a Delphi study with implications for agricultural education programs. *Journal of Southern Agricultural Research* 52, 1 (2002), 1–13.

[48] S. Sentance and A. Csizmadia. 2016. Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies* (2016), 1–27.

[49] D. Shinners-Kennedy and S.A. Fincher. 2013. 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 '13).* ACM, San Diego, San California, USA, 9–18. https://doi.org/10.1145/2493394.2493396

[50] T Sirkiä. 2012. *Recognizing Programming Misconceptions–An Analysis of the Data Collected from the UUhistle Program Simulation Tool.* Department of Computer Science and Engineering, Aalto University. Master Thesis.

[51] D. Sleeman, R.T. Putnam, J. Baxter, and L. Kuspa. 1988. An introductory Pascal class: A case study of students' errors. In *R. Mayer(Ed) Teaching and Learning Computer Programming: Multiple Research Perspectives.* Norwood, NJ, Lawrence Erlbaum Asociates, 237–257.

[52] L.A. Thomas, J. Boustedt, A. Eckerdal, R. McCartney, J.E. Moström, K. Sanders, and C. Zander. 2017. In the liminal space: software design as a threshold skill. *Practice and Evidence of the Scholarship of Teaching and Learning in Higher Education* 12, 2 (2017), 333–351.

[53] W. Vernon. 2009. A Delphi technique: A review. *International Journal of Therapy and Rehabilitation* 16, 2 (2009), 69–76.

[54] F. Woudenberg. 1991. An evaluation of Delphi. *Technological forecasting and social change* 40, 2 (1991), 131–150.

[55] X. Yang, L. Zeng, and R. Zhang. 2012. Cloud Delphi method. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 20, 1 (2012), 77–97.

[56] B. Zwaneveld, J. Perrenet, and R. Bloo. 2016. Discussion of methods for threshold research and an application in computer science. In *R. Land, J.H.F. Meyer, and M.T. Flanagan (Eds) Threshold Concepts in Practice.* Springer, 269–284.