Teaching Methodologies for Third-Level Computer Science Support Centres

Their impact on academic development of students

Olga Murdoch
Olga.murdoch@ucdconnect.ie
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland

Tadhg O'Sullivan
t.osullivan@ucd.ie
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland

Henry Mcloughlin and John Dunnion
Henry.mcloughlin, john.dunnion {@ucd.ie}
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland

Abstract

The Computer Science Support Centre (CSSC), formerly known as the Computer Programming Support Centre (CPSC) was established in 2008 by the UCD School of Computer Science and Informatics. A previous publication (Naughton, 2010) introduced the CPSC and examined its use and effectiveness based on data collected from attendance records and student feedback. The numbers of students attending the CPSC were observed to increase over time and it was illustrated through analysis of attendance records that the main area students sought help in were related to the implementation of specific programming languages.

The number of students attending the CSSC has continued to increase over time meaning there are more students than ever that rely on CSSC support at some point in their studies. Repeat visits by individual students have also increased. It is essential that the teaching methodolgies adopted by CSSC tutors enable students to become independent learners, rather than becoming dependant on the additional support throughout their studies. The existance of attendance records since the establishment of the support centre has provided valuable insight into the effectiveness of teaching methodologies adopted by tutors in the CSSC. This paper extends the work of the previous authors by describing the evolution of these teaching methodologies, based on ongoing analysis of student interactions with the centre. A concept-led teaching methodology for supporting students with programming difficulties is described and its impact is examined through comparative analysis of attendance records before and after this approach was adopted in the CSSC.

Keywords

Third-level education, Support Centre, Computer Science, Teaching methodologies

1. Introduction and Motivation

The Computer Science Support Centre (CSSC – formerly CPSC) is a free academic support centre that operates on a drop-in basis from Monday to Friday and is available to all UCD students taking Computer Science modules. This work extends that of (Naughton, 2010) which introduced the support centre and provided initial insights into it's effectiveness as an academic support through analysis of the attendance records that had been accumulated over the centres first two years of operation. It was found that students who received assistance from the support centre are more likely to pass the modules they received tutorials in. The numbers of visitors and time spent by visitors were shown to increase over time. With the CSSC now considered a staple academic support by many of its visitors it becomes necessary to re-evaluate the services offered with an aim to providing the best and most appropriate support to its students.

The UCD Education Strategy states strategic objectives which include:

- To foster early and lasting student engagement
- To grow and develop graduate education
- To widen participation and support lifeline learning

It is with these objectives in mind that we define the goals of the CSSC and the services provided to achieve such aims. With a wider goal of reducing the drop-out rate of Computer Science students through the provision of one-to-one and group tutorials, the CSSC must aim to enable students to become successful and independent in their academic studies.

It is of upmost importance, to both the student and the university, that the degrees (and assessments that comprise such) awarded to students are an accurate representation of the knowledge and skills learned and developed by a student through their academic studies. It is therefore necessary to consistently review the attendance records with respect to the support strategy and teaching methodologies adopted by the CSSC tutoring team. Such a review took place midway through semester one of the 2011/2012 academic year and its findings and the resulting changes to the support strategy and teaching methodologies are presented in this paper. While the CSSC aims remain unchanged, these changes have caused a shift in the way tutorials are provided and the way students use the support centre. Comparative analysis of attendance records prior to and post the implementation of the new teaching methodologies provide grounds for evaluating the effectivness of concept-led support centre teach as opposed to the previous student led approach.

Section 2 identifies the issues with student-led support strategies for teaching practical subjects such as computer programming as found in the 2011 CSSC review. Section 3 introduces the alternative concept-led teaching methodologies and outlines the changes to the support strategy that aim to overcome the challenges identified in the previous section. A comparative analysis forms the evaluation of the new teaching methodology in Section 4 followed by conclusions and future work.

2. Challenges in Student-Led Computer Science Support Centres

From its conception the CSSC operated using a student-led approach to tutorials. Students mostly attended the support centre having encountered a difficulty in a practical exercise, usually programming. The student receives a tutorial covering the issue and is then able to complete their work having overcome the initial difficulty. While the steadily increasing number of CSSC visitors is an indication strongly in favour of the popularity of the service, it also warrants closer analysis in order to validate the cause of the rise.

Midway through semester one of the 2011/2012 academic year, a review of the supports offered by the CSSC was undertaken. It was found that students regularly utilized the CSSC as a support for completing on-going assignments. By availing of the CSSC with an aim to completing a particular assignment, students were often found to successfully submit assignments without understanding the core concepts underpinning them. The result of such interactions is repeat visits by the student who needs to ‘re-learn’ previous concepts when they appear or are presented in a previously unseen context.

Despite the efforts of a CSSC tutor to ensure a student understands the concepts involved in a tutorial, students frequently require repeat tutorials covering the same or similar topics. The differing goals of each participant involved needs to be considered in order to determine the ideal solution. The student has likely become frustrated with a programming problem and is attending the CSSC in order to find a solution. Some students visit the CSSC a number of times during their completion of a single practical assignment. The student sees the tutor as a means to solve a programming problem. The tutor is aiming to ensure the student understands how the problem arose and how it can be solved in future instances. Based on feedback from both students and tutors it is evident that in many cases the student does not learn transferable skills from the tutorial as, their primary goal, completing the assignment has been achieved.

3. A Concept-Led Teaching Methodology for Support Centres

With an aim to addressing challenges described in Section 2, the way the CSSC operates was amended. From November 2011 tutors in the CSSC were no longer permitted to help students directly with any work that is to contribute to their grade for a Computer Science module. Students were no longer permitted to work on assignments that are yet to be graded under the supervision of a CSSC tutor. This change meant that students were required to think more about the problems they are facing before attending the CSSC than they had been previously.

Enforcing new rules to within a pre-existing support service could potentially alienate students who may become unsure as to whether they are welcome or are suited to use a particular service. To ensure inclusivity of all students a set of CSSC usage guidelines and problem comprehension and decomposition advice were issued to students prior to the change being implemented by tutors. Details of the academic support structure within CSI were explained and the secondary nature of the CSSC was reinforced and students were also assured that they should still attend the CSSC when struggling with assignments, but that the more general support they received would have long term benefits in terms of their academic development.

A core challenge in this change to operation is the ability of the tutor to work with a student in order to identify what the student's pain point actually is. The tutor must manage their interactions with the student in a manner that will help both tutor and student reach a conclusion as to what the student actually needs support with. With this is mind, tutors are advised to follow a conversational deduction process. For example, a student presenting themselves to the CSSC may open a conversation with the tutor by saying *"my assignment program doesn't work and I don't know why"*. The tutor may now converse with the student by asking the following questions:

Question 1: *"What is the assignment asking you to do?"*
Question 2: *"How have you broken down the problem?"*
Question 3: *"What programming concepts are required to solve each sub-problem?"*
Question 4: *"Can you explain these programming concepts?"*
Question 5: *"Do you understand the different issues that may arise when implementing these concepts?"*

By using a conversational process such as this, the tutor is able to identify the point in the assignment completion process at which the student becomes stuck, as per the problem types defined in (Naughton, 2010). Questions 1 and 2 relate to **assignment understanding**. Does the student fully understand what is expected of them and how to break a larger task into smaller problems that are easier to solve (problem decomposition). Questions 3, 4 and 5 are asking if the student understands the **general concepts** required to solve a problem. In a programming task for example, this means understanding what programming constructs are available and what problems each construct can be used solve. Question 5 in particular requires a thorough understanding of a concept that enables the student to apply a programming construct (for example) to solve a problem in a manner that does not leave their solution open to issues. In programming in particular this is a hugely important question when it comes to supporting students and, typically, the answer is not specific to any programming language. In the case that the student can answer all of the above questions without too much difficulty, the tutor can now deduce that the student may in fact be having difficulty with **implementation-syntax** specific to the programming language they are using. The tutor can then help the student to understand the syntactical requirements of the specific language and may also provide advise in how to effectively debug (find syntactic problems) their program.

4. Evaluation

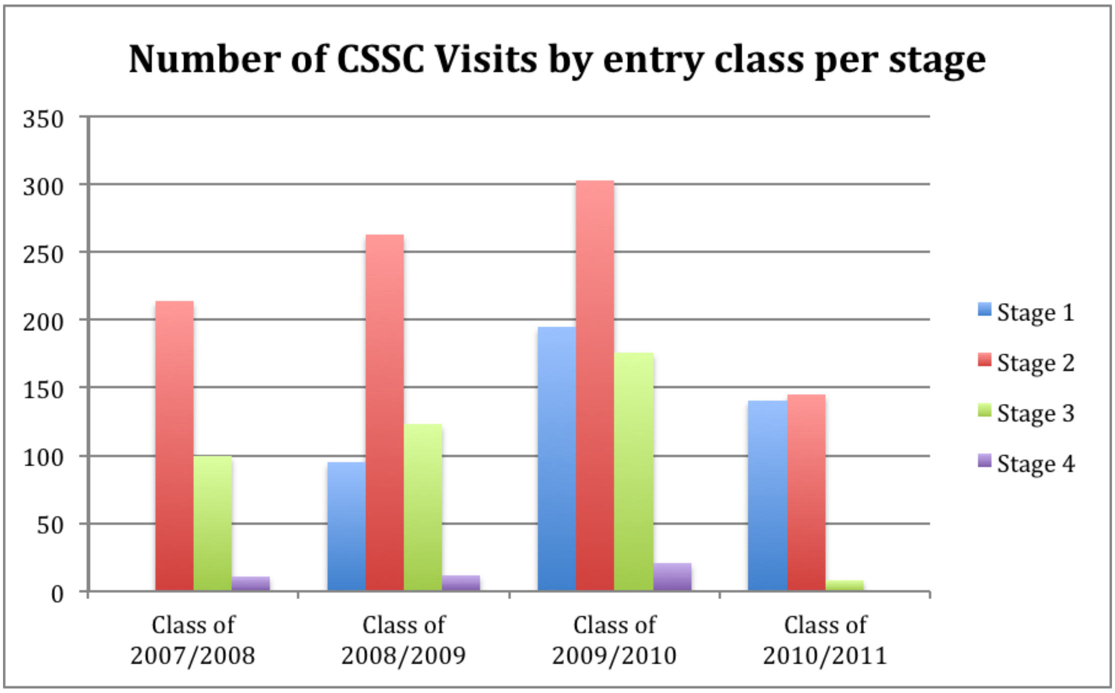

Figure 1 – Number of CSSC visits by yearly entry class per stage of study for each class. (Note that the CSSC first opened in the 2008/2009 academic year. This means that the class of 2007/2008 were in Stage 2 when they could first avail of the service. Similarly, the class of 2010/2011 have not yet entered Stage 4 and as such there is no information available. Stage 4 for the class of 2009/2010 and Stage 3 for the class of 2010/2011 represents partial information for the first semester of the 2012/2013 academic year.)

Previous analysis of attendance records identified an increase in the number of visits each academic year since the CSSC opened. While this trend has continued, a more interesting trend lies in the CSSC use on a year's entry class basis. Figure 1 illustrates the number of CSSC visits for the Classes of 2007/2008, 2008/2009, 2009/2010 and 2010/2011. Visualising the information in this manner makes it clear to see the spike in CSSC visits as each group enters stage 2. However the class of 2010/2011 only saw a minor increase in CSSC visits during stage 2. The reduction in visits aligns directly with the new teaching methodology. While numbers for stage 4 Class of 2009/2010 and stage 3 Class of 2010/2011 are partial, in that they represent only part of a semester, it is evident that the former group utilised the CSSC more often and more consistently in their later years than the group who were earlier in their academic career when the CSSC changes were enforced.

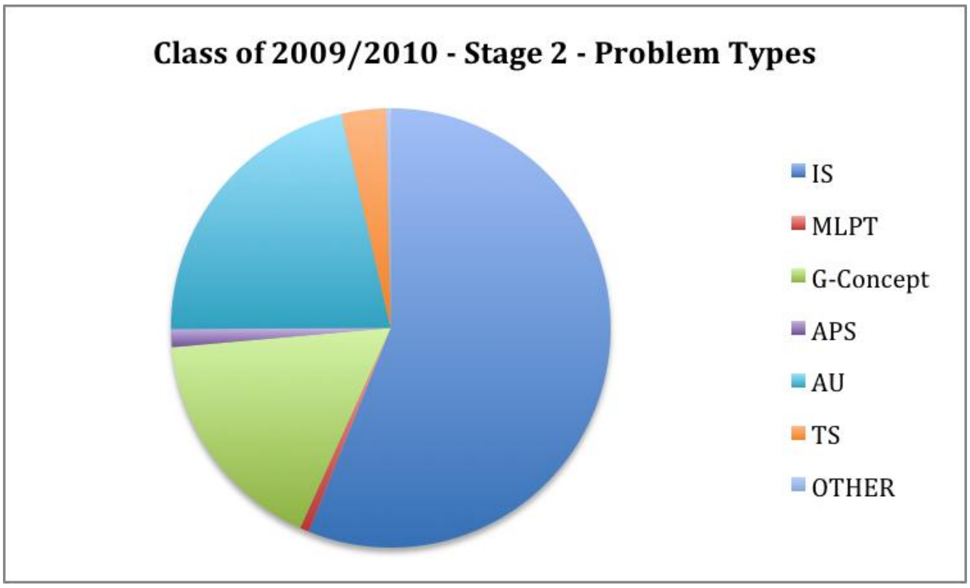

Figure 2 – Pie chart illustrating the distribution of problem types presented in the CSSC by the class of 2009/2010 in Stage 2 of their degree (2010/2011 academic year)

To gain further insight into the differences between the CSSC usage of the class of 2009/2010 and the class of 2010/2011, we take a closer look at the types of problems brought to the CSSC while each group were in stage 2 of their degrees. Figure 2 illustrates the distribution of problem types for the 2009/2010 class. Over 56% of the visits from this group were classified as Implementation and Syntactic problems which is typically an indicator that the student has had help writing/debugging code directly. Such an approach is suitable for programming tutorials but when a student requires help understanding the concept rather than the implementation of the concept, this approach is of little benefit to the students' academic development. Another 38% of the visits have been classified as either assignment understanding or a general concept problem which indicates that the tutorial involved more high level explanations of concepts in a manner that aims to enable the student to implement a solution independently.

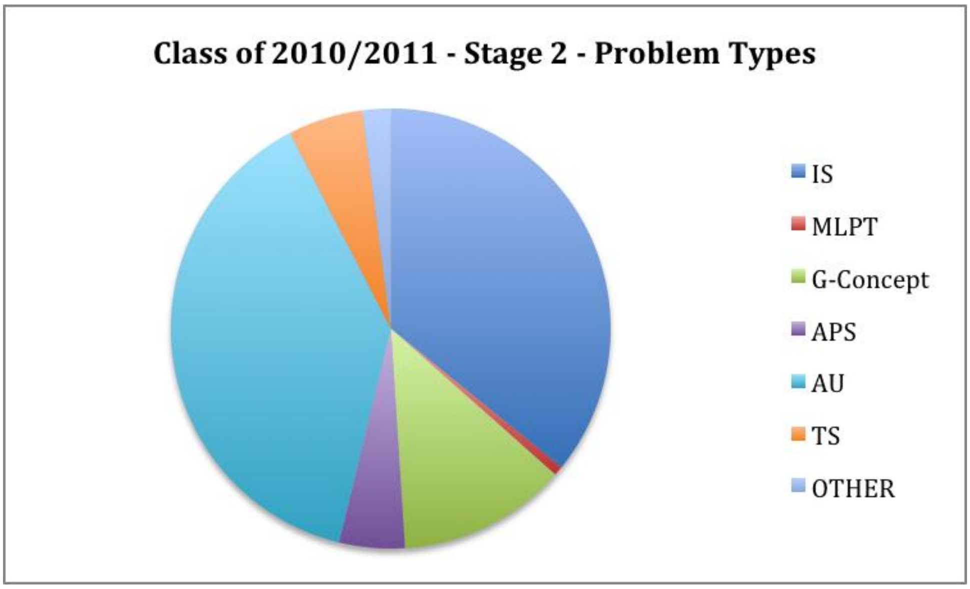

Figure 3 – Pie chart illustrating the distribution of problem types presented in the CSSC by the class of 2010/2011 in Stage 2 of their degree (2011/2012 academic year)

Figure 3 illustrates the same data for the class of 2010/2011 in stage 2 of their undergraduate degree. This data corresponds to tutorials given during and after the changes to the CSSC support strategy were introduced. Implementation and Syntactic problems have been reduced to 36%, while the combination of Assignment Understanding and General Concept problems has risen to over 50%. This shift in focus to less hands on tutorials indicates that students are being supported in a way that enables them to implement programming solutions independently instead of relying on the handholding of CSSC tutors. Such a progression is further illustrated through the reduction in CSSC visits by students in 3rd and subsequent stages. The pattern of reduced visits appears to increase following the adoption of a concept-led teaching methodology which places onus on the student to perform hands work independently.

5. Conclusions and Future Work

As defined in previous work, hands-on learning is the main methodology through which Computer Science (CS) students will understand concepts such as programming. However, students who face difficulties with programming often get frustrated, misunderstand the problem and seek the easiest and quickest solution. A Computer Science Support Centre can easily become the source of solutions for students who struggle with programming and as such it is imperative that the CSSC adopts appropriate teaching methodologies that will support a student while aiding their development into successful and independent academics. Analysis of attendance records of the CSSC while a student-led and then a concept-led teaching methodology was in use provides initial insight into the impact of such teaching methodologies on the progression of CS students. It has been illustrated that a concept-led approach that encourages independent hands-on learning by the student may result in more long term improvements to a student's independent academic progress. In future work additional metrics relating to exam pass rates, student dropout rates should be included in a longitudinal study that evaluates the impact of teaching methodologies on undergraduate students. The support strategy and teaching methodologies utilised in the CSSC will be consistently reviewed and adapted according to the changing needs of undergraduate students.

6. References

Naughton, M., N. Curley, J. Carthy, and T. Kechadi (2010). "Third-Level Specialised Support Centres Their impact on student success."