Breaking the coding barrier: transition from Level 1 to Level 2 programming

Other

Covering page

Project Title: Breaking the coding barrier: transition from Level 1 to Level 2 programming

Your name: Frances Chetwynd

Report submission date: 10 July 2018

Names of any key staff associated with the project:

Fiona Aiken; Helen Jefferis

Contact point:

f.d.chetwynd@open.ac.uk

Executive Summary

In this research we have investigated the use of a visual-based programming language at HE Stage 1 for beginner programmers. We conducted a survey of Stage 2 programming students to ascertain their retrospective views on their preparedness for text-based programming in year 2 having completed visual-based programming studies in year 1 (quantitative and qualitative data). Using whole cohort datasets from the Open University (UK) – an open entry distance learning HEI – we analysed the assessment scores of students at Stage 1 and Stage 2. Finally, we investigated the efficacy of offering a bridging course to programming students between Stages 1 and 2. We found that a majority of survey respondents did find visual-based programming prepared them well for higher level studies except for the syntax aspects of text-based programming. The assessment data, however, showed that doing badly at year 1 visual programming does not indicate that students will do badly in year 2 using text-based programming. Data analysis of the bridging course attendance showed that students who were most likely to benefit from attending did not do so, and those that did attend did not gain any immediate benefits in the Stage 2 performance. In our efforts to conduct the research for this project we found that OU student assessment data is hard to acquire. Additionally, we found that analysing student results across multiple qualifications, pathways and study rates is resource intensive.

Aims and scope of your project

What were the main aims of the project?

This project aimed to investigate the success of the current approaches to teaching programming to Level 1 undergraduate students at the Open University within the School of Computing and Communications, STEM Faculty. Measures of success used were (a) retention and progression of students to Stage 2 programming modules and (b) student perceptions of their preparedness for Stage 2 study.

Specific aims were to:

1. Investigate the success of using Sense (an in-house variation of Scratch) as an introductory programming language and its role in preparing UG students for Stage 2 programming using text-based languages;
2. Establish if any correlation exists between levels of student engagement and attainment with the programming at Stage 1 and success in programming in Stage 2 modules;
3. Investigate the role the programming skills bridging course (which is offered to all students starting Stage 2 programming modules) in promoting progression to Stage 2 and retention on Stage 2 modules.

What were the more specific goals?

The initial goals were very broad encompassing all Stage 2 programming teaching within the qualification. This was subsequently narrowed to focus on the largest Stage 2 module (M250). Later the project scope was extended to investigate the efficacy of the bridging course which was introduced in summer of 2016 after the project started.

Specific goals were:

1. To better understand the influence of Level 1 programming teaching on students' choice of modules at Level 2;
2. To better understand students' perceptions of their readiness for Stage 2 text-based programming study having primarily studied visual-based programming at Stage 1;
3. To demonstrate any correlation between poor student performance in visual programming at Stage 1 and poor performance in text-based programming at Stage 2;
4. To demonstrate any improvement in retention of students at Stage 2 who took a skills-based programming bridging course prior to their Stage 2 studies
5. To provide evidence to support the Faculty's choice of programming languages in the new Stage 1 modules.

Activities

The project adopted a combined qualitative and quantitative approach using student survey data, assessment data and bridging course participation data.

Qualitative approach:

In this part of the project we surveyed a single cohort of 810 students at the end of their first Stage 2 programming module. The survey questions were designed to elicit students' views on whether their Stage 1 programming studies using Sense had adequately prepared them for higher level studies using a text-based language (here Java) using free-text responses. Questions focused on different aspects of programming such as problem-solving, syntax and variables and asked how well students felt they had been prepared for M250, now that they had finished studying it. Additionally students were asked if their future choice of modules had been influenced by their introductory programming experiences. The following specific activities were undertaken and data gathered:

- Literature review undertaken to establish previous work (both within HE and schools) on visual-based vs text-based programming
- Questions and wiring diagram developed for survey
- SRPP and HREC guidance and authorisation obtained;
- Survey opened 23rd September 2015 to 810 students of 2014J M250 cohort
- Survey closed 21st October 2015 with 203 responses (25%)
- Qualitative analysis of survey responses undertaken

Quantitative approach

In this part of the project we analysed the quantitative data from the student survey, extensive assessment data from 6 student cohorts on TU100 and M250, and limited data from student participation in the bridging course.

In the survey, students responded to questions about how their Sense studies prepared them for Stage 2 using scales such as "very well; fairly well; poorly; not at all" which data we then analysed. The following specific activities were undertaken and data gathered:

- Survey set-up activities as above
- Quantitative analysis of survey responses undertaken

For the analysis of student performance in assessment the cleansing, sorting and analysis of the data proved extremely complicated. Extensive records were obtained of TMA scores for the specific programming questions on TU100, which were mixed in across all TMAs. As well as the usual difficulties in working with OU student data such as incomplete records, withdrawn and deferred students, and inconsistent data formats, a particular issue here was that the assessment scheme on TU100 changed frequently and the placement of individual programming questions moved about within TMAs. Further complications arose when trying to track students from TU100 to M250 across various years and cohorts. After some initial pilot analysis it was decided to restrict the data used to a single cohort of TU100. SPSS Statistics Package V22 was used to analyse the data, and the following specific activities were undertaken and data analysed:

- Students who had studied M250 before TU100 were excluded
- Scatter graphs were plotted to compare the average Sense scores with the M250 OCAS
- These were also split into cohorts to see if the length of time between studying level 1 and level 2 had any effect.
- The strength of correlation between the level 1 and level 2 scores were calculated for a variety of scenarios.
- A more detailed look was taken at the students that did badly at level 1 and those that id badly at level 2 (scoring <50%)

The bridging course data proved particularly elusive partly because of its non-standard use of VLE features. The summer of 2016 data was eventually obtained. It became quickly obvious that levels of participation had varied widely and therefore some pseudo measures were devised, allowing us to conduct analysis using groupings such as “no participation; limited participation; full participation”. The following specific activities were undertaken and data gathered:

- Literature review undertaken to establish previous work on bridging courses, identify both the approaches taken by other HEIs to bridging courses and the approaches to tackling skills gaps in STEM subjects
- Scatter graphs were plotted for each of the 3 groupings to compare the students’ level 1 and level 2 results to see if there was any relationship between the level of participation and their results
- The relative performance for students that took a full part in the bootcamp was carefully compared with that of students that took no part.

Dissemination activities

The following dissemination activities have been undertaken:

2016 **5th eSTEeM 2016 annual conference, Open University** – short presentation and poster presentation

(Chetwynd et al, 2016a)

OU Learn about fair stand, Open University – poster presentation and software demonstration

Horizons in STEM Higher Education 2016, Leicester University – short presentation and poster presentation

(Chetwynd et al, 2016b)

2017 **1st Computing Education Practice conference, Durham University** – short presentation

(Chetwynd et al, 2017a)

Annual HEA STEM Conference 2017, Manchester – two short presentations and poster presentation

(Chetwynd et al, 2017b); (Chetwynd et al, 2017c)

6th eSTEeM 2017 annual conference, Open University – workshop

(Chetwynd et al, 2017d)

Horizons in STEM Higher Education 2017, Edinburgh University – short presentation

(Aiken et al, 2017e)

STEM bridging/transition projects workshop, Open University – short presentation

2018 **7th eSTEeM 2018 annual conference, Open University** – structured discussion

(Chetwynd et al, 2018a)

FINDINGS

1. Visual programming provides Stage 1 HE distance learning students with a moderate level of confidence in introductory programming techniques when they start their Stage 2 studies, using text-based languages.

The quantitative analysis of the survey results revealed that:

- 67% of those responding were very well or fairly well prepared for the programming problem-solving requirements at Stage 2
- 75% of those responding were very well or fairly well prepared in the use of control structures required at Stage 2.

Koulouri et al (2014) showed that the choice of programming language has an impact on the success with which novice programmers develop appropriate programming skills. Furthermore, Armoni et al (2015) in their work with secondary school children found that the "programming knowledge and experience of students who had learned Scratch greatly facilitated learning the more advanced material in secondary school: less time was needed to learn new topics, there were fewer learning difficulties, and they achieved higher cognitive levels of understanding of most [programming] concepts". Our findings support their conclusions.

The survey responses further revealed that, as might be expected, learning introductory visual programming does not prepare most students for the rigours of writing programs with more complex syntax (e.g. java or python).

- 35% of those responding were very well or fairly well prepared for using programming languages that allow errors in syntax at Stage 2

More surprisingly, despite the significant differences between the structure of programs in Sense compared to Java, nearly half of those responding felt that visual programming had prepared them for working with more complex program structures.

- 45% of those responding were very well or fairly well prepared

2. For some groups of students there is a correlation between their programming scores at Stage 1 and Stage 2, but not for those students who do badly at Stage 1.

Figure 1 below, plots the average programming assessment score achieved at Stage 1 (on TU100, using Sense an in-house version of Scratch) compared to the average assessment score achieved on at Stage 2 (on M250, using Java). There was a moderate significant correlation between achievement in assessment at the two stages ($n = 803$ $r=0.499$, $p <.000$).

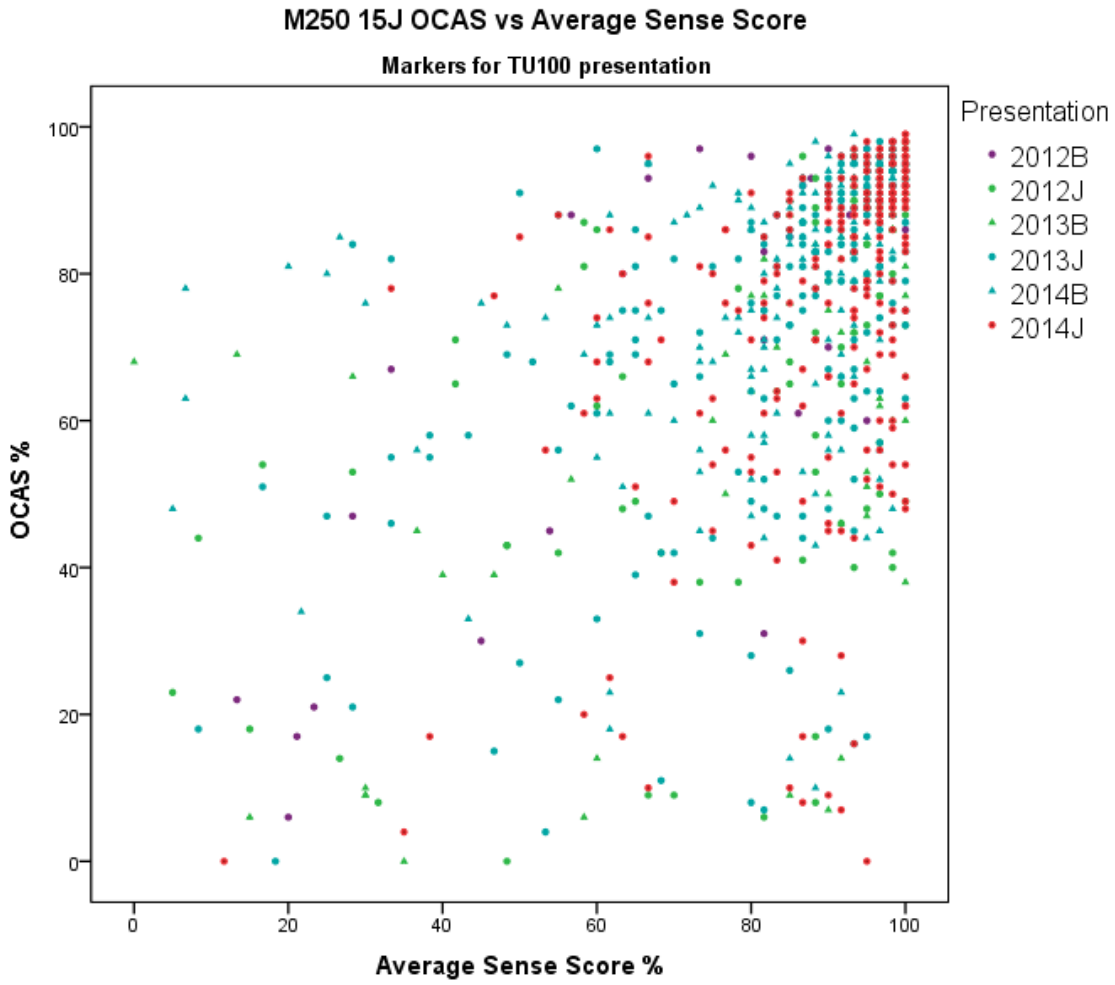

Fig 1: Average stage 1 Sense score (TU100) vs average assessment score at Stage 2 (OCAS on M250)

Looking in more detail at the left hand side of the graph, for students with Stage 1 Sense scores of 50% or less (n = 62), there was no correlation between their Stage 1 Sense scores and their Stage 2 OCAS (Fig 2). The mean OCAS score for students in this group was, however, only 42.73% compared with a cohort mean of 73.3%. So, doing badly at Stage 1 is no predictor of success or failure at Stage 2.

And looking in more detail at the lower half of the graph, for students with Stage 2 OCAS of less than 50% (n=125), there is a significant but weak correlation (r=0.268 p=.002) (Fig 3).

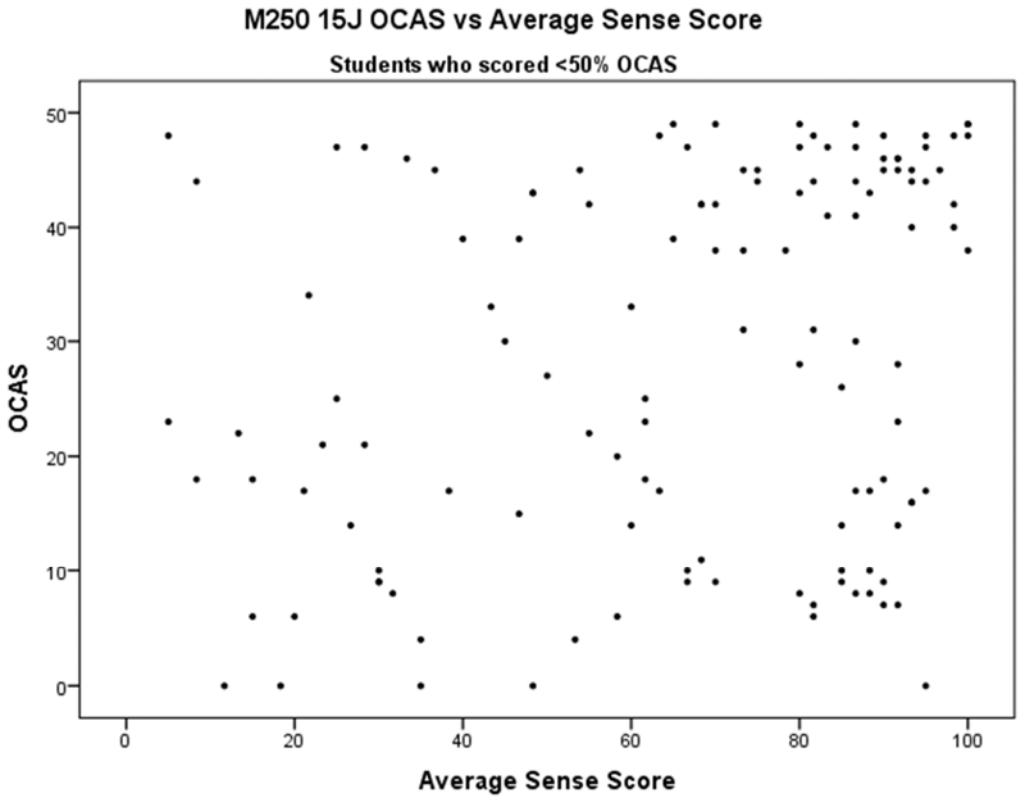

Fig 2: Stage 1 Sense score (TU100) of less than 50% vs Stage 2 OCAS (M250)

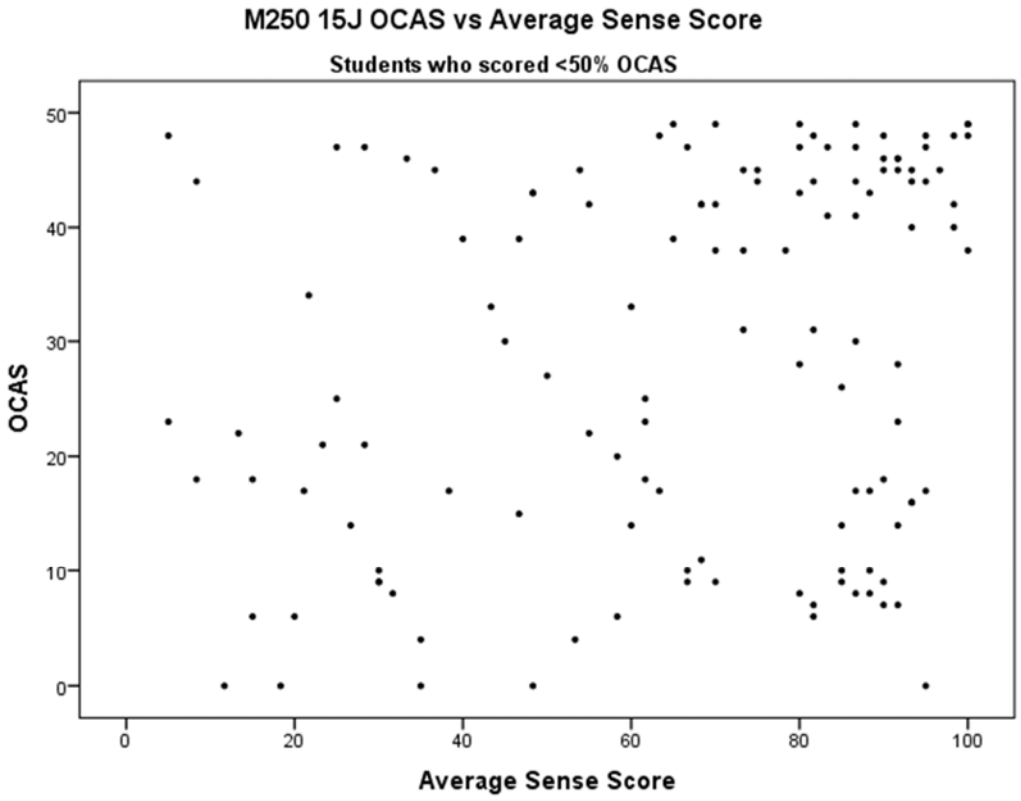

Fig 3: Stage 2 OCAS (M250) of less than 50% vs Stage 1 Sense score (TU100)

3. The students that took part in the bridging course were not necessarily the ones that would have benefitted most from the additional practice.

Computer Sciences has a sector-wide continuation rate of 91% which is one of the lowest for any discipline within UK HE (Woodfield, 2014) and, as usually found at the OU, the continuation rate within our Computing and IT qualifications is significantly lower than that. In their paper "Learning Computer Science Concepts with Scratch", Meerbaum-Salant et al (2013) found (with school children) that to effectively learn CS concepts using Scratch students need 'close and effective mentoring' rather than being left alone to experiment, when they will learn very little. Over the nine month TU100 module the students followed a set of developed activities, with self-assessment checks, and formal assessment questions. Meerbaum-Salant's work suggests that those students who did badly will not improve by simply experimenting with Sense between Stages 1 and 2. Therefore, to have any chance of improving the likelihood of poorly-performing CS students at the OU successfully continuing on to Stage 2, some form of directed study is needed, such as the Bootcamp.

Of the 803 students that completed M250, 140 of them had fully engaged in the bootcamp. Of these only 6 of them had scored less than 60% on programming at level 1. So these students had mostly done well at Stage 1 and therefore, although not guaranteed to be equally successful at Stage 2, certainly had a strong likelihood of doing so without the benefit of the bridging course. 364 of the completing M250 students didn't engage with the bridging course at all, and of these 55 (15%) scored 60% or less on programming at Stage 1. Given that the average score on the programming at Stage 1, of the 803 M250 students, was approx. 73.3%, these students were well below average.

4. For students who attended the online one week bridging course (AKA bootcamp), for Computing and IT students progressing to Stage 2 programming, attendance had no effect on the grade outcome of their studies of M250.

At 60%, Computer Science is at the bottom end of the table for percentage of students achieving a first or upper second (Woodfield, 2014), and the Open University has one of the lowest rate of awarding firsts in the UK HE sector. It is important, therefore, that the OU provides opportunities for its Computing students to improve their chances of getting a top grade in their qualification. HE students have reported that the step up to second year leaves them feeling unprepared for aspects such as the workload and more demanding academic content and assessments. Does second year have an identity problem? "First year is for settling in; final year is packed with preparation for exams and employment. Second year is the middle child in a three year undergrad curriculum" (Milsom, 2015). In Computing and IT, for stage 2 students, new and more complex programming languages can be a barrier to progress.

Offering a bridging course to all students embarking on their Stage 2 computing modules does not immediately appear to be a cost effective means of boosting performance because this attracts the 'wrong' students, as shown in Finding 3. However, Douglas and Attewell in a 2014 paper named "The Bridge and the Troll Underneath: Summer Bridge Programs and Degree Completion", concluded that "at community colleges and less selective 4-year colleges, students who attend bridge programs are 10 percentage points more likely to finish within 6 years." This suggests that the impact of the bridging course may be wider than its intended aim, and further investigation is needed.

5. Whole cohort module datasets of assessment scores at the OU are highly complex, particularly where students can take a variety of pathways and qualifications at different study rates.

Although we had data for a number of cohorts across several years for two modules at Stage 1 (TU100 and TM129) and three modules at Stage 2 (M250, M269 and TT284), it was quickly evident that tracking individual students through a multitude of possible routes would not be cost-effective in terms of analyst time. However, given the cohort sizes at the OU, it was also evident that focussing on one module for one year would produce a dataset of significant size. For example, using the end data for one year of M250 results in over 800 students to consider. In the end therefore we picked the 2015 M250 cohort and tracked the students backwards to the different Stage 1 TU100 cohorts that they came from. The project used assessment data down to individual question level. Whilst this is difficult to obtain it does offer the advantage of being a compulsory element of a module, and therefore the most widespread measure of student performance in a cohort.

Further work is needed to establish which students, if any, did badly at Stage 1 AND took the bridging course AND did not submit the first assessment for M250.

SUCCESSES, UNANTICIPATED OUTCOMES AND UNEXPECTED OPPORTUNITIES

The most impactful aspect to date of this large project has been the work on the Computing and IT Level 1 to Level 2 bridging course (Bootcamp). Other researchers have concluded, through slicing the data differently, that attending the bridging course at best had no impact on the likelihood of a student's success on M250, and at worst may have reduced their chances of success. Our work has shown that those students most likely to benefit from attending the bridging course actually did not engage with it at all. As a result, rather than cancelling the Bootcamp altogether for remaining TU100 students, the Faculty is taking steps (including considering ways to involve ALs in selecting and encouraging students to join) to improve the attendance rate of those most likely to benefit. Our research is also feeding into discussions about future bridging courses in Computing and IT and the wider Faculty.

There are two key unanticipated outcomes of the project.

The first, is the (in print) publication of a journal article entitled 'Reflections on the 2017 HEA STEM conference: Graduate employability challenges and solutions' in a special edition of the AdvanceHE journal *Higher Education Pedagogies* published by Taylor and Francis (Chetwynd et al, 2018b). This came about as a result of attending the 2017 HEA STEM Annual conference where contributions for the special edition were requested in the opening remarks. Although this project did not have employability as a primary theme, the skills focus of the work made it suitable to be included as part of the wider discussion in the journal paper. The employability work has been further disseminated in a workshop at the 7th annual eSTEeM conference, and is planned to be presented at several cross-faculty AL Staff Development events around the UK and Ireland.

The second, is the publication of a short video describing the methodology and key findings of the project on both the eSTEeM website and YouTube. This video is not only an external-facing illustration of the work of the Open University and the STEM Faculty, it is also for use internally by the nascent scholarship centres being created in the other three faculties.

The main unexpected opportunity that arose was to disseminate the work to the Student Experience and TEI workstream senior managers in the Students First Transformation programme. Although the SFT work is not progressing currently, cross-faculty dissemination is not always easy to achieve, and the briefing was a valuable opportunity.

<u>Impact</u>

a) Student experience, Teaching and Learning design

During the course of the project work the C & C School replaced the subject Stage 1 60 credit module, TU100, with two 30 credit modules, TM111 and TM112. The first of these – TM111 – uses OUBuild for programming which is a version of Scratch and very similar to Sense as used on TU100. The second of these – TM112 – uses Python a text-based programming language frequently used in education. Whilst the programming teaching on TM111 is of a similar level to TU100, and generally the students do very well on this, the programming on TM112 is of a higher level and moves the

students from using visual programming to text-based programming. The first two findings of our project strongly support the decisions made by the School in replacing TU100.

The second two findings of this project contribute significant knowledge to the efficacy of running bridging courses for distance learning students. The findings demonstrate that students who are most likely to benefit from attending a bridging course will need to be contacted personally and encouraged to sign up for it and attend.

Overall, the project has demonstrated that OU cohort data, when considering multiple modules across a number of years, is extremely complex. Whilst it is relatively easy to track an individual student, following multiple cohorts of students across several possible pathways and qualifications results in a spider's web of interconnectivity. Given the current work on CSR and a new data strategy for the OU, consideration should be given to providing this type of data in a readily accessible and usable format.

List of deliverables

Digital object – video presentation of project approach and findings

(Chetwynd et al, 2017e)

Journal paper – in print paper in Advance HE journal 'Higher Education Pedagogies'

(Chetwynd et al, 2018b)

Conference presentations and abstracts – as listed in dissemination activities and referenced below

Figures and tables

Fig 1: Average stage 1 Sense score (TU100) vs average assessment score at Stage 2 (OCAS on M250)

Fig 2: Stage 1 Sense score (TU100) of less than 50% vs Stage 2 OCAS (M250)

Fig 3: Stage 2 OCAS (M250) of less than 50% vs Stage 1 Sense score (TU100)

References

Aiken, F.J. Chetwynd., F.D., Jefferis, H., (2017) 'Getting a leg over the coding barrier: an evaluation of the success of pre-start bridging courses for stage 2 computer programming students.' *Horizons in STEM Higher Education Conference*, Edinburgh, 29-30 June 2017. p33.

Armoni , M., Meerbaum-Salant, O., Ben-Ari, M., (2015) 'From Scratch to "Real" Programming', *ACM Transactions on Computing Education,* vol. 14, no. 4, Article 25, pp 25:1 -25:15. https://dl-acm-org.libezproxy.open.ac.uk/ft_gateway.cfm?id=2677087&ftid=1546361&dwn=1&CFID=49398910&CFTOKEN=c23be133d2c9bb82-5F870F14-C49F-FBFE-A1C6809ECAF464AE

Chetwynd, F.D., Jefferis, H., Aiken, F.J.(2016a) 'Breaking the coding barrier: Transition from Level 1 to Level 2 programming', *The 5th annual eSTEeM conference*, The Open University, Milton Keynes, 14-15 April 2016, pp 26, 72-73. http://www.open.ac.uk/about/teaching-and-learning/esteem/sites/www.open.ac.uk.about.teaching-and-learning.esteem/files/files/ecms/web-content/2016-04-14-Chetwynd-Jefferis-Aiken.pptx

Chetwynd, F., Aiken, F. and Jefferis, H. (2016e) 'Breaking the coding barrier: transition from Stage 1 to Stage2 programming' *Horizons in STEM Higher Education* 30 Jun – 1 Jul 2016, Leicester University, UK

Chetwynd, F., Aiken, F. and Jefferis, H. (2017a) 'How high is the coding barrier? A quantitative analysis of student transitions from stage 1 to stage 2 programming' *Computing Education Practice (CEP) Conference* 11 Jan 2017, Durham University, UK http://community.dur.ac.uk/cep.conference/2017/programme.php#abstract13

Chetwynd, F., Aiken, F. and Jefferis, H. (2017b) 'How high is the coding barrier? A quantitative analysis of student transitions from stage 1 to stage 2 programming' *HEA STEM Annual conference* 1 – 2 Feb 2017, Manchester, UK
https://www.heacademy.ac.uk/system/files/downloads/chetwyndheastemmanchesterfeb2017coding_barrier1final.pdf

Chetwynd, F., Aiken, F. and Jefferis, H. (2017c) 'Getting a leg up over the coding barrier: pre-start programming bootcamps for Stage 2 students' *HEA STEM Annual conference* 1 – 2 Feb 2017, Manchester, UK
https://www.heacademy.ac.uk/system/files/downloads/chetwyndheastemmanchesterfeb2017coding_barrier2final.pdf

Chetwynd, F., Aiken, F. and Jefferis, H. (2017d) 'Bridge Over Troubled Waters: Would your students benefit from a bridging course to help them transition to second year?' *6th Annual eSTEeM conference 2017.* 25 – 26 Apr 2017, The Open University, Milton Keynes, UK

Chetwynd, F., Jefferis, H., and Aiken, F. (2017e) 'Breaking the coding barrier: transition from Stage 1 to Stage2 programming', Open University centre for STEM pedagogy, video uploaded 12 Jul 2017, http://www.open.ac.uk/about/teaching-and-learning/esteem/resources/videospodcasts/scholarship-shorts-series or https://www.youtube.com/watch?v=B05YMQz7Avs

Chetwynd, F.D., Jefferis, H. and Aiken, F.J. (2018a) Employability skills: Myth, Monster or Misunderstanding? The 8th eSTEeM conference, The Open University, Milton Keynes, 25-26 April 2018.
http://www.open.ac.uk/about/teaching-and-learning/esteem/sites/www.open.ac.uk.about.teaching-and-learning.esteem/files/files/ecms/web-content/2018-04-25-26-Frances-Chetwynd-presentation.pptx

Chetwynd, F., Aiken, F. and Jefferis, H. (2018b) 'Reflections on the 2017 HEA STEM conference: Graduate employability challenges and solutions', *Higher Education Pedagogies*, in print

Douglas, D. and Attewell, P., (2014) *The Bridge and the Troll Underneath: Summer Bridge Programs and Degree Completion*, American Journal of Education 121, no. 1 (November 2014): 87-109.

Koulouri,T., Lauria,S., Macredie, R.D., (2014) ' Teaching Introductory Programming: A Quantitative Evaluation of Different Approaches', *ACM Transactions on Computing Education,* vol. 14,no. 4, Article 26, pp 26:1-26:28.
https://dl-acm-org.libezproxy.open.ac.uk/ft_gateway.cfm?id=2662412&ftid=1524093&dwn=1&CFID=49398910&CFTOKEN=c23be133d2c9bb82-5F870F14-C49F-FBFE-A1C6809ECAF464AE

Meerbaum-Salant, O., Armani M., Ben-Ari, M., (2013) 'Learning Computer Science Concepts with Scratch*', Computer Science Education,* vol .23, no.3, pp239-264.
https://www-tandfonline-com.libezproxy.open.ac.uk/doi/abs/10.1080/08993408.2013.832022

Milsom, C. (2015) 'Disengaged and overwhelmed: why do second year students underperform?' *The Guardian*, 16 Feb 2015 [Online]. Available at https://www.theguardian.com/higher-education-

network/2015/feb/16/disengaged-and-overwhelmed-why-do-second-year-students-underperform (Accessed 8 Feb 2017).

Wolz, U., Leitner, H.H., Malan, D.J., Maloney, J., (2009) 'Starting with Scratch in CS1', Proc. of the 40th SIGCSE Technical Symposium on Computer Science Education, Chattanooga, Tennessee, USA, March 3-7, 2009, pp2-3.

Woodfield, R. (2014) Undergraduate Retention and Attainment across the Disciplines [online]. York: Higher Education Academy. www.heacademy.ac.uk/node/10293

Statement of ethical review

An ethical review was obtained according to the Open University's code of practice and procedures before embarking on this project. Reference number HREC/2015/2095/Chetwynd.

Appendices

None