

Connecting Undergraduate Students as Partners in Computer Science Teaching and Research

Mike Brayshaw¹ & Neil Gordon²

Computer Science, University of Hull, Hull, HU6 7RX, United Kingdom ¹m.brayshaw@hull.ac.uk;
²n.a.gordon@hull.ac.uk

*Corresponding Author: n.a.gordon@hull.ac.uk

Keywords: *Enquiry Based Learning; Computing Education; Research Inspired Education.*

Abstract

Connecting undergraduate students as partners can lead to the enhancement of the undergraduate experience and allow students to see the different sides of the university. Such holistic perspectives may better inform academic career choices and postgraduate study. Furthermore, student involvement in course development has many potential benefits. This paper outlines a framework for connecting research and teaching within Computer Science—though this is applicable across other disciplines. Three case studies are considered to illustrate the approach. The first case study involves students in their honours' stage (level 6, typically 3rd year) project, the second an undergraduate intern between stages 5 and 6, and finally, a MSc (level 7) project. All three case studies have actively involved students in core parts of the University's teaching and research activities, producing usable software systems to support these efforts. We consider this as a continuing engagement process to enhance the undergraduate learning experience within Computer Science.

Introduction

Research is a word that can mean different things on different levels. The common definition of research for many is the collection and assembly of existing knowledge to fulfill the

academic assessment requirements that they are challenged with. Undergraduates often cannot see beyond research as being this searching and foraging for knowledge. The importance of appreciating the research side of a university is to provide an understanding of what research actually is and to develop the capacity for research and development, both within academia and beyond given the needs of the modern knowledge economy. Teaching and Research can be partners – particularly when the goals of the research to be undertaken is of a pedagogic nature. Engaging students in the process of improving course content, student experience and delivery of that experience is one way of delivering this approach.

The engagement of students can be done within the context of a live course and the research may be contemporaneous with the course delivery, where students can be instrumental in the design and implementation of this change. Action research (Carr & Kemmis, 1986) renders students as active agents in the change process; in this approach, a traditional scientist identifies a set of dependant and independant variables, they then devise a series of perturbations to those variables, and then, from the point of view as a detached observer, watch how those variable behave. However, those variables can also be observed from within the process of

change. This approach is action research. A particular application of this work in education is by Freire (1938) who tried to break down the classic teacher-student barriers and reform the power relationships between the two. New students should not be seen as empty vessels to be filled with knowledge by the teacher, but the student and the teacher are both co-learners. His approach led to participatory action research which aims to involve all participants as co-researchers – an essentially fluid process where input and prior knowledge and experience is equally valued by all those taking part. The learner is an agent of change and able to influence their future world.

The following sections explore specific pedagogic approaches that were utilised within the case studies, with enquiry based learning in the first and third case studies and a flipped approach supported by a MOOC in the second case study.

Enquiry Based Learning and Teaching Computer Science

Enquiry Based Learning (Vygotsky, 1934; Bruner, 1961; CILASS, 2016) is where learning is undertaken by the student in a self-directed manner – not in a directed chalk and talk style. The nature of the task or problem may be open, and there is not a strict or single way of solving it. Through the efforts of their enquiry the students must learn the nature of the problem and what methods of exposition and solution are suitable. Central to this approach is self-analysis and critical thinking. The teacher acts as a research supervisor, not providing solutions but acting as a facilitator to their students' progress. Enquiry and research can be done on many levels – the vocabulary and what research means will be very different for a first year undergraduate from that of final year or a post-graduate. Enquiry Based Learning allows students to be partners and fellow explorers in the learning space. As argued elsewhere, Computer Science lends itself to this topic - indeed this is often what neophyte Computer Scientists do naturally of their own accord (Gordon & Brayshaw 2008).

Flipped Learning and New Partnerships

A relatively recent model of learning the breaks down the barriers of classical classroom based teaching is Flipped Learning where students study new material by themselves, e.g. at home, and then work on projects and group tasks in the classroom. The Flipped Learning Network (FLN) define it at

“Flipped Learning is a pedagogical approach in which direct instruction moves from the group learning space to the individual learning space, and the resulting group space is transformed into a dynamic, interactive learning environment where the educator guides students as they apply concepts and engage creatively in the subject matter”

(FLN, 2016)

In the context of connecting undergraduates as partners in teaching it clearly breaks down the type of unhelpful issues Freire (ibid) raises. The dynamics interactions can be used as the starting point of collaboration. However if it is to be successful then the partnership has to be of a certain type. The teacher has to act more as a guide and facilitator, providing feedback and flexible assessment the FLN argue – just leaving it to the dynamics of the classroom without the professional input of a teacher will not bring success by itself.

Developing New Partnerships

The notion of Connection is one of engagement throughout the student journey. This engagement can start from the initial point of contact – the initiation of the contractual bond between student and university within the admissions process – right through to graduation and beyond. Enquiry based and flipped learning can provide new mechanisms by which to connect and engage in new partnerships with students, developing collaboration at various points of the journey, thereby developing the students' capacity to engage with research. As outlined above, approaches to develop this can be developed throughout the student journey. Some examples are identified within a simply

framework for connecting research to teaching, along with specific case studies illustrating this framework:

- Enquiry Based Learning Sessions from year one onwards;
- Student Internships
 - Year Long – e.g. embedded within a research group where they can be attached for a longer period and take on larger projects;
 - Short Summer Internships: where they can undertake small projects with the research groups [case study 2 below].

- Project Based Sessions (UG and MSc) [case study 1 and 3 below];

Three Illustrative Case Studies

The following case studies utilise the above framework (figure 1) to develop the skills in students through enquiry based activities from level 4 (first year) upwards. Early activities involve team projects and peer assessment, to provide a scalable solution to manage medium to large (300) cohorts of students. Having developed these skills across level 4 and 5, the internship (level 5 to 6 transition), individual or team based honours projects (level 6), and individual masters project (level 7) show how students have been effective at applying their skills to develop novel solutions.

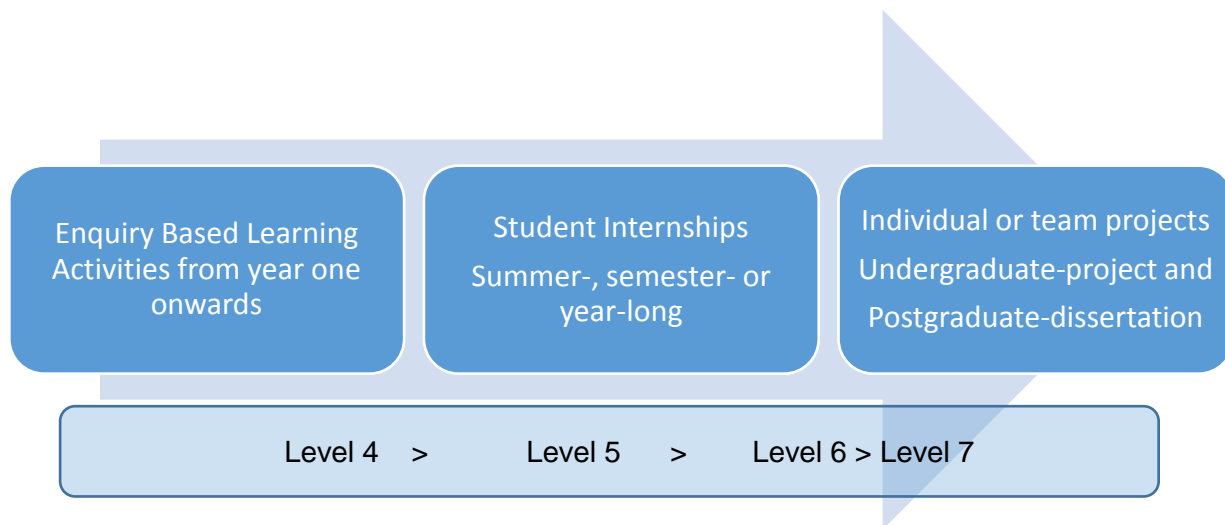


Figure 1 The framework for connecting research to teaching

Case Study 1: A Pedagogically Motivated Inquiry Based Tutor for C#

Overview

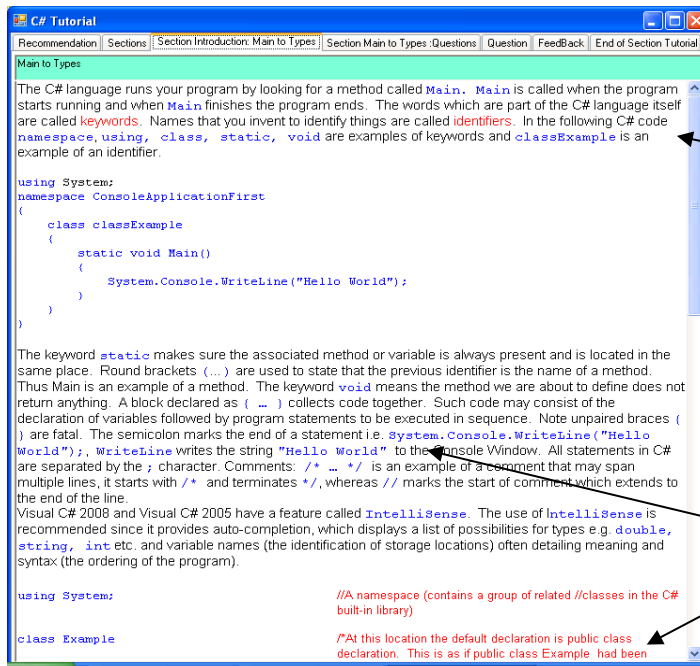
This first case study illustrates the undergraduate project approach. Within the context of a level 6 (honours stage/final year) project, a student worked with their academic supervisor to carry out a literature survey and then develop a research approach to produce a tutoring system for the C# programming language. This was followed up with a formal evaluation and documentation that led to a publication.

The project

Learning by being told is a classic method of instruction, utilised in many tutoring systems for learning programming languages e.g. The Lisp Tutor (Anderson et al, 1984). This model illustrates the classical roles of teacher and student. Traditional Socratic dialogues can be augmented with classical models of misconceptions (e.g. Anderson & Jeffrey, 1985), to teach and debug the students' learning. This approach assumes that the teacher's model of the world is a superset of the tutee's model, which is not necessarily true (Moyse & Elsom-

Cook, 1990). An alternative method making use of enquiry is to let the user explore via discovery (Papert, 1968); this work aimed to let students explore a rich world and to discover things about it through exploration. This work should be viewed in context with contemporary chalk and talk approaches or approaches that emphasize strong learning by association, e.g. programmed

Learning (Skinner, 1965). As a pragmatic compromise, Elsom-Cook (1990) proposed Guided-Discovery Learning where the student is guided into making the appropriate discoveries; whilst learning in their own context, mental model and vocabulary, the learning could be ordered, structured, and timely.



The colour coded banner to represent a selected section

The Section Introduction text has all the programming code written in the colour blue and in a different font to make it clear to read and stand out to the Student. Important information is put in red to make it stand out to the student.

Figure 2 Material from the C# tutor. All the material and the presentation was designed by the student.

In the above context, and synthesizing ideas from Enquiry Based Learning, the student worked with their supervisor to produce a C# Tutoring System for the first 20 hours of an Introductory Programming Course (Butterworth & Brayshaw, 2014). The student created the whole course themselves, drawing on the experience of others who had designed first programming courses. The very act of writing your own first programming course for the first year naturally raised many design and structuring issues. A guided discovery rationale for the delivery of the material was adopted. Figure 2 shows the typical tutorial screen from the system.

delivery soon became apparent. To this end a Student Designed Management and Analytics was implemented (figure 3).

Evaluation and Outcomes

This work was valued using Heuristic evaluation. The Experts used in the evaluation were lecturers who had experience of delivering computer programming courses at Undergraduate Level. On the basis of these evaluations a new revised version of the C# Tutor was implemented. The heuristic evaluation was reported along with two other PhD evaluations in an academic publication (Brayshaw et al, 2014).

In addition to the Student Designed Curriculum, the role analytics can take in course design and

Questions	NotAnswered	Correct	InCorrect	Answered
34	27	6	1	7

Section	Question	Answer	QuestionTitle
1	1	Correct	First Example Main, Part 1
1	2	Correct	First Example Main, Part 2
1	3	Correct	First Example Class
1	4	Correct	using System
1	5	Correct	Console
1	6	NotAnswered	Statements
1	7	NotAnswered	comments
1	8	NotAnswered	identifiers
2	1	Correct	verbatim
2	2	InCorrect	escape sequences
2	3	NotAnswered	storing and accessing variables
2	4	NotAnswered	int
2	5	NotAnswered	float, double
2	6	NotAnswered	decimal
2	7	NotAnswered	char
2	8	NotAnswered	bool
2	9	NotAnswered	Cast, Expressions
2	10	NotAnswered	if statement

Figure 3 Analytics to show the students' progress with the material and how this view could then provide teacher's with insights into progress

Case Study 2: Learning and Gaming in Media Enriched MOOC

Overview

This second case study illustrates the intern project approach. Taking place between level 5 and level 6, this is based on a summer internship model – the student receives a small stipend to support them over the project period - where the student works with an academic member of staff. In this example, the intern carried out a small scale research project on the relevant pedagogic approaches and the supporting technologies to enable the development of a MOOC. This was followed up with a formal evaluation and write up.

Context

For the second Case Study we developed a MOOC that we then used with a level 5 (diploma stage/second year) Undergraduate AI course and subsequently evaluated. A key feature was to exploit the potential of a MOOC in terms of media delivery of the course and to include elements of gamification into the course. Gamification is the inclusion of elements that have game dynamic properties, such as scoring, puzzles and/or fun activities. The system was implemented in EdX. The basic screen is shown in figure 4. The main topics in the course are shown down the left (the course itself was already written) and thus presented a navigation bar. Here we see a snapshot mid-way through an exercise where the student is required to click on the stopping condition of the recursion.

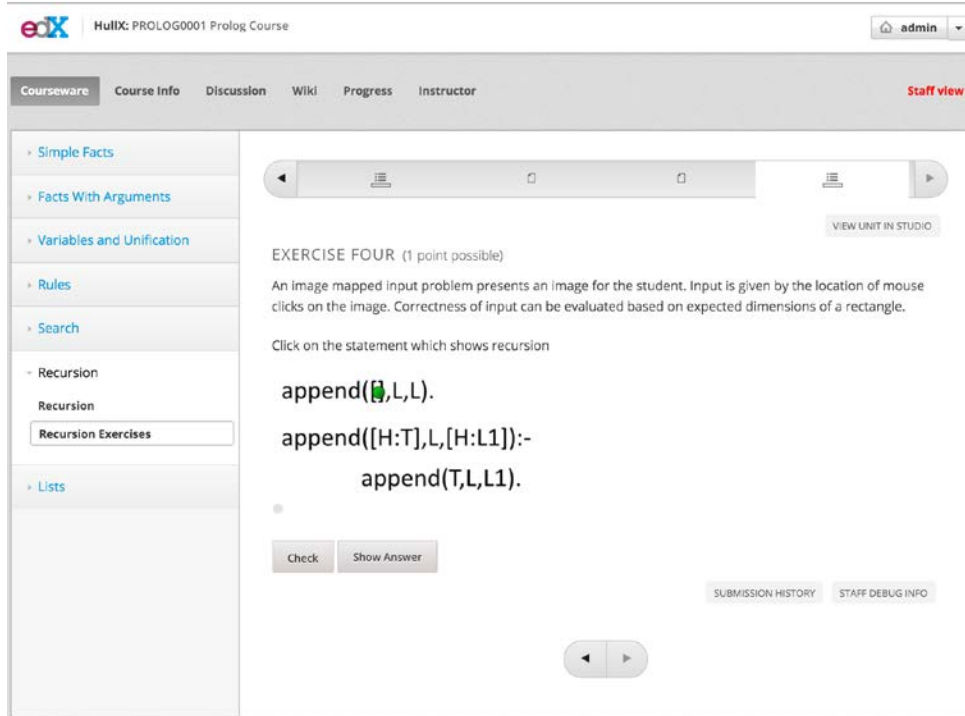


Figure 4 Brayshaw and Balaghan showing the navigation bar on the right to element of the Introduction to Prolog course and a sample exercise taking place on the right hand side.

Media Enriched Using YouTube Animations

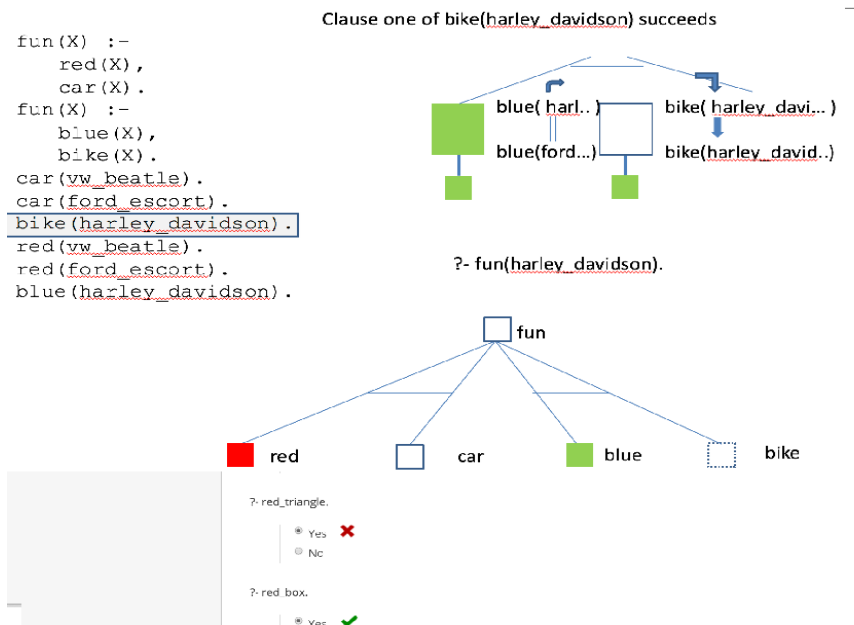


Figure 5 Showing the program code on the left. The line of code currently being executed is shown on the left. The goal tree is shown towards the bottom of the screen and at the top is a Fine-grained view showing individual clause head unifications.

To further add to the user experience Software Visualisation of the Prolog Virtual Machine was used. This was done by using TPM (Brayshaw & Eisenstadt, 1991) a visual model of the behavior of Prolog. This was tightly coupled to the program itself in the animations produced by the first author and delivered via YouTube (figure 5).

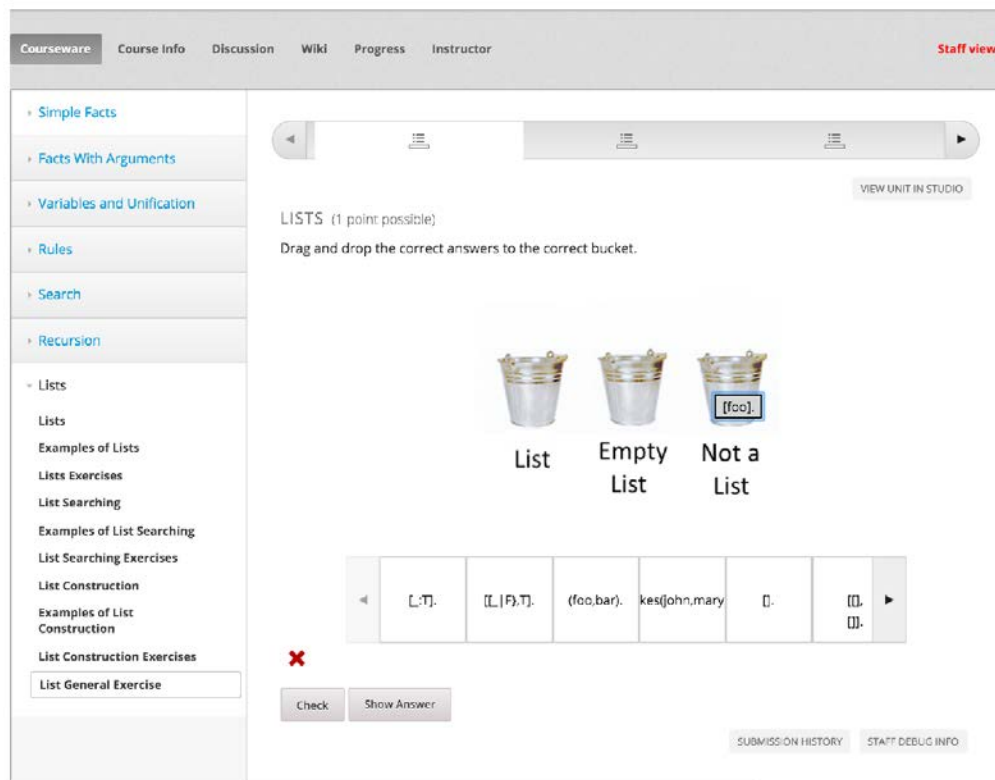
As we noted above, an element we wanted to include was gamification. Figure 6 illustrates one sort of interaction. Along the bottom are a series of symbols, some are in lists, others are not. The game is to correctly place these items in one of

three bins. One bin is marked List, the next Empty List (a list containing nothing), and the last Not a List. The goal is thus to correctly put the right things in the correct bins.

Evaluation & Outcomes

The system was successfully used in 2014/15 for a two Semester Course on AI to teach the Prolog part of the course. An evaluation was undertaken with broadly positive results. The whole project was then written up and reported in Brayshaw and Balaghan, (submitted). The student involved went on to do postgraduate work.

MOOC and Motivation



Timeliness, Novelty and Game Elements

Figure 6 A simple sorting game.

Case Study 3: Social Computing and Personalisation

Overview

This final case study illustrates the master's level (level 7) project approach. In this example, the

project involved developing a technology to utilise APIs to link to Twitter and provide a personalised feed of data. Again, a key part of the project was the formal review, analysis and writing up of the research undertaken by the student.

Context

Navigation and seeing the wood for the trees is a common problem as we have to deal with large information spaces online. This is also the case in the world of Social Media. This project was

undertaken with an MSc student and aimed to take an existing Adaptive Information Retrieval technique to produce Smart Twitter Portal (Mounota & Brayshaw, 2015)

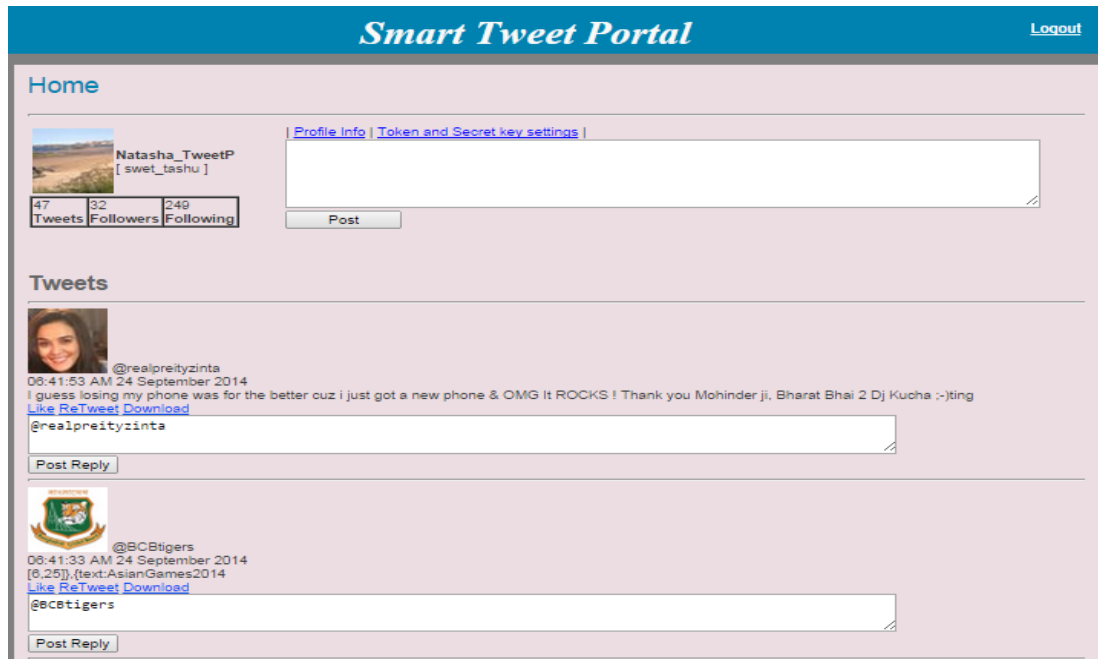


Figure 7 Screen Snapshot of the Smart Twitter Portal (Mounata & Brayshaw, 2015)

Through research into a variety of personalisation algorithms, along with investigation of different social media platforms and their associated API (Application Programming Interfaces). Developing the personalisation algorithm and implementing the relevant calls to the Twitter API, the students were able to develop a personalised portal. The portal allows the user to tailor their social media feeds and personalise it to their own interests. It uses their own behavior with the material they look at to prioritise subsequent views of incoming Tweets.

Evaluation & Outcomes

The system was evaluated with 12 users, using a formal approach from Human Computer Interaction design (Brayshaw et al, 2014). The expert feedback was positive, with an overall score of 91.7% for the relevance of tweets it displayed. This work was also written up and published, and the student later went on into an academic career.

Conclusions and Further Work

Utilising the framework (figure 1), the three cases studies illustrate, at different locations in the student journey, ways to connect teaching and research. The emphasis in case studies one and two has been on the inclusion of students in core university activities from teaching and research. The approach can also be important for recruitment by showing prospective students the difference in activities that go on at University compared with schools and how, for example by a suitable intern program, they can become active participants in a fuller range of activities. This in turn can lead to helping with engaging and motivating using partnership to promote and improve the undergraduate user experience. Wagner (2016) argues this involvement may help confidence in female undergraduates, which in turn might redress the gender gap in Computer Science. Increasing engagement and participation should help performance, satisfaction, and retention. Indeed partnerships

may be widened to include others from beyond the confines of the university. Preece (2016) discusses using collaboration and citizen science in the context of improving Human Computer Interaction. Such a hybrid approach could be used in the future for expanding teaching and research partnerships to include players from beyond the confines of the campus, potentially including suitable industrial locations.

Acknowledgements.

The work reported here was made possible by our work with the following students: Phininder Balaghan, Adele Butterfield, and Natasha Mounota.

References

Anderson, J.R., Farrell, R. & Sauers, R. (1984) *Learning to Program in LISP*, *Cognitive Science*, 8, pp 87-129.

Anderson, J.R. & Jeffries, R. (1985) *Novice LISP Errors: Undetected Losses of Information from Working Memory*, *Human Computer Interaction*, 1, pp 107-131

Brayshaw, M. & Balaghan, P. (nd), *Learning and Gaming in a Media Enriched Prolog MOOC within a traditional delivered course* (submitted)

Brayshaw, M. & Eisenstadt, M.A. (1991) *Practical Graphical Prolog Tracer*. *International Journal of Man-Machine Studies*. **35**, pp. 597-631, ISSN 0020-7373.

Brayshaw, M., Gordon, N., Ngjani, J.T., Wen, L. & Butterfield, A.M. (2014) *Investigating Heuristic Evaluation as a Methodology for Evaluating Pedagogical Software: An Analysis Employing Three Case Studies.*, *Learning and Collaboration Technologies: 16th International Conference on Human Computer Integration*, Greece, *Lecture Notes in Computer Science*, Springer.

Bruner, J.S. (1961) *The act of discovery*. *Harvard Educational Review* 31 (1): 21–3
978-3-319-07481-8

Butterfield, A.M. & Brayshaw, M. (2014) *Pedagogically Motivated Guided Discovery Tutoring System for C#*, *Proceedings of the HEA STEM (Computing) Learning Technologies 2014 Workshop*, University of Hull; ISBN: 978-1-907207-48-8

Carr, W. & Kemmis, S. (1986) *Becoming Critical: Education, Knowledge and Action Research*, Deakin University Press.

CILASS (2016) *Centre for Inquiry-Based Learning in the Arts and Social Sciences (CiLASS)*, *University of Sheffield*, Available <https://www.shef.ac.uk/ibl/cilass> (Accessed 5th October, 2016)

Elsom-Cook, M. (1990) *Guided Discovery Tutoring* in M Elsom-Cook (Ed) *Guided Discovery Tutoring: A Framework for ICAI Research*, London: Paul Chapman, ISBN 0442308353

Eisenstadt, M. & Brayshaw, M. (1988) *The Transparent Prolog Machine (TPM): an execution model and graphical debugger for logic programming*. *Journal of Logic Programming*, **5**(4), pp. 277-342, ISSN 0743-1066.

FLN (2016) Available at <https://flippedlearning.org/> [Accessed 2nd March, 2017].

Freire, P. (1970) *Pedagogy of the Oppressed*. New York, Continuum. ISBN 0816491321
Goldschmid B and Goldschmid ML, 1975, *Peer Teaching in Higher Education: A Review*, *Higher Education*, **5**(9), DOI:10.1007/BF01677204.

Gordon, N.A. & Brayshaw, M. (2008) *Inquiry based Learning in Computer Science teaching in Higher Education*, *Innovations in Teaching And Learning in Information and Computer Sciences*, *ITALICS*, **7** (1), pp22-33, ISSN 1473-7507.

Griffiths, S., Housten, K. & Laenbatt, A. (1995) *Enhancing student learning through peer tutoring in Higher Education*, *University of Ulster*.

Mounota, N. & Brayshaw, M. (2015) *Personalizing your social computing world: A case study using Twitter*, Science and Information Conference 2015, pp263-268.

DOI: 10.1109/SAI.2015.7237153

Moyse, R. & Elsom-Cook, M. (Eds) (1992) *Knowledge Navigation*, Academic Press, September, ISBN-10: 0125093780

Papert, S. (1980) *Mindstorms: Children, Computers and Powerful Ideas*, Brighton: Harvester Press.

Phillips, E.M. & Pugh, D.S. (2010) *How to Get a PhD: A handbook for students and their supervisors*. Open University Press, ISBN-10:0-33-524202-2.

Preece, J. (2016) *Citizen Science: New Research Challenges for Human-Computer Interaction*, International Journal of Human-Computer Interaction, **32**(8).

DOI 10447318.2016.1194153.

Vygotsky, L.S. (1934) *Thought and Language*, Alex Kozulin (Ed), MIT Press, 1986, 0-262-72101-8

Wagner, I. (2016) *Gender and Performance in Computer Science*, ACM Transactions on Computing Education, **16**(3).

DOI: 10.1145/2920173.