Teaching Novices Programming Using a Robot Simulator: Case Study Protocol

Louis Major Theocharis Kyriacou Pearl Brereton

School of Computing and Mathematics
Keele University, UK
l.major@keele.ac.uk t.kyriacou@keele.ac.uk o.p.brereton@keele.ac.uk

Keywords: POP-I.B. programmer education, POP-V.B. research methodologies, POP-VI.E. Computer science education research

Abstract

This protocol provides details of a case study design that will investigate the use of simulated robots as introductory programming teaching tools. This research is motivated by the results of a Systematic Literature Review which indicated that such work would be valuable. The protocol will help to ensure that a reliable, transparent and rigorous study is performed. Furthermore, potential problems have been considered and accounted for in advance of its implementation. The protocol may also act as a point of reference for other researchers interested in performing a case study.

1. Introduction

In this paper a protocol for a planned case study is presented. Case studies are empirical strategies for research which involve an investigation of a phenomenon using several sources of evidence. This case study will investigate the use of simulated robots as introductory programming teaching tools. The research has been influenced by the results of a Systematic Literature Review (SLR) which indicated that such work would be valuable. A range of participants will be involved in the case study including novice programmers and trainee high school teachers. Data collected during the study will be used to evaluate the effectiveness of a robot simulator, and associated workshop, which have been developed to support the learning of introductory programming. This research aims to contribute to knowledge by addressing the findings of the SLR. Moreover, this is the first case study to examine the implementation of a robot simulator in such a context.

The remainder of this paper is organised as follows. In Section Two information relating to the background of the research project is presented. Section Three provides information about the design of the case study. Section Four offers an overview of data that will be collected while Section Five provides details of how this will be analysed and interpreted. In Section Six measures which have been taken to ensure the validity of the case study, in addition to a consideration of potential limitations, are outlined. This is followed by a summary in Section Seven.

2. Background

Learning to program a computer is a difficult task for novices (Kelleher and Pausch 2005). Various efforts have been made by educators to overcome such a problem by implementing active learning environments (McGill 2012). This has included using robots as teaching tools (Fagin 2003, Lauwers et al 2009, Martin and Hughes 2011, McWhorter and O'Connor 2009). The work that is presented is motivated by the results of a SLR which investigated the use of robots in such a manner (Major et al 2011a, Major et al 2012). In total, 36 papers were accepted in the SLR. Of these: 25 examined the effectiveness of physical robots, seven the effectiveness of simulated robots and four the use of physical and simulated robots together. 26 papers (75%) report robots to be effective when used to teach introductory programming. However, the potential to further investigate the use of robots remained, particularly in regards to simulated robots. This is because the quality and rigour of the seven papers related to simulated robots was judged to be inadequate as: four offer a 'lessons learned' account, or description of an approach, and provide no empirical data (Becker 2001, Buck and Stucki 2001, Enderle 2008, Ladd and Harcourt 2005); one describes the results derived from interviews as being non-generalisable as only four novices were involved (Borge et al 2004); one specifies the use of a questionnaire but presents no quantitative data (Lemone and Ching 1996); one describes the implementation of pilot lessons but does not undertake detailed analysis (Sartatzemi 2005).

As a result of the SLR, and after reviewing educational software guidelines (Squires and Preece 1999, ANSI Standards 2001, Beale and Sharples 2002), a robot simulator and associated workshop material have been developed. The simulator is modelled after a small real world robot called the Mark III[1]. The robot has two actuators and several input devices. The workshop covers the fundamental programming constructs identified in the ACM/IEEE Computer Science Curriculum Joint Task Force Report (ACM/IEEE 2008). This includes: basic syntax and semantics of a higher-level language; variables, types, expressions and assignment; simple I/O; conditional and iterative control structures; methods and parameter passing; structured decomposition. In addition, the workshop also includes an introduction to arrays. The ACM/IEEE recommend a minimum coverage time of nine hours for the fundamental constructs they identify. With the introduction of arrays the workshop will last around 10 hours in total. Java has been selected as the programming language that will be taught as this is used on Computer Science courses at Keele University as it is currently one of the most widely adopted programming languages (TIOBE 2012). The simulator and workshop were piloted with a number of novice programmers in order to validate the procedures and instruments. Other research has also taken place including the hosting of introductory programming sessions involving 23 pre-service and seven in-service ICT/Computer Science teachers. These sessions were used to evaluate an early version of the robot simulator and to determine participant's attitudes towards the teaching of programming. Some of this research is described in Major et al (Major et al 2011b).

This case study will form part of a PhD project. The case study methodology is being used as it is highly flexible and suitable for complicated studies involving multiple human participants. Case studies are strategies for research which involve an empirical investigation of a phenomenon using several sources of evidence (Robson 2002). One strength of case studies is that they are able to provide a deeper understanding than controlled experiments (Runeson et al 2012) whilst remaining capable of achieving scientific objectives (Lee 1989). The development of a protocol helps to ensure reliable, transparent and targeted research which considers potential problems in advance (Yin 2009).

3. Design

In this section information relevant to the case study design is outlined including the aim of the study, propositions, workshop structure, participants, data sources, cases, procedures and roles.

3.1 Aim

This is an exploratory case study as it aims to seek new insights (Runeson et al 2012). Moreover, it is also considered to be a positivist case study as past evidence has been examined (as detailed in the SLR), a range of variables will be measured, propositions will be tested and inferences will be drawn from samples to stated populations (Klein and Myers 1999). This protocol is based on one described by Brereton (Brereton et al 2008). The aim of this study is to determine whether a robot simulator is effective for supporting the learning of introductory programming by using such a tool in a specially designed workshop. The following research question will be asked:

Is a robot simulator an effective tool for supporting the learning of introductory programming?

3.2 Propositions

Four propositions have been developed as a result of the case study aim:

P1 A robot simulator is an effective tool for supporting the learning of introductory programming

P2 A robot simulator improves novice's perceptions of programming

P3 A robot simulator offers a more effective introduction to basic programming concepts when compared to other teaching methods

P4 A robot simulator improves trainee ICT/Computer Science teacher's confidence in their ability to teach introductory programming

[1] http://www.junun.org/MarkIII/

3.3 Data Sources

Several data sources will be used during the case study. The research question and propositions will be addressed as follows:

- By using **questionnaires** to determine participants programming knowledge, in addition to their attitudes towards programming, both before and after the workshop.
- By administering (and later scoring) **programming tests** both during and after the workshop, which have been constructively aligned with the learning objectives of the workshop, in order to determine programming progress.
- By maintaining a **log of events** that occur during the workshop session.
- By **interviewing** three current teachers, who have been involved in the planning of the workshop sessions, in order to determine their thoughts on the effectiveness of the simulator.

3.4 Workshop Structure, Participants and Cases

Two separate workshop sessions have been scheduled which will involve students with limited or no programming experience aged between 16 and 18 years old. Each workshop will last two days (5 hours per day) and will involve 10 and 11 students respectively. The two sets of students are studying at different Further Education (FE) institutions. Two programming tests will be completed by students in order to gauge progress and a pre- and post-workshop questionnaire will be administered. At the end of the workshop students will also complete four programming tasks which are designed to draw together the concepts that they have encountered.

22 trainee ICT/Computer Science teachers will also take part in a workshop that will replicate the FE students workshop discussed above. Two separate two-day workshops will again be held and this cohort of trainees will be split. Whereas the students have had limited exposure to programming these trainee teachers have all encountered programming in some capacity before. Therefore, in addition to undertaking the assessment tasks designed for student participants, trainee's confidence in their ability to teach programming (in addition to their attitude towards the subject) will be investigated. This will be done by administering a pre- and post-workshop questionnaire. The trainees will also be asked to compare their previous programming learning experience to the one using the robot simulator.

In addition, a further introductory programming workshop has been scheduled. This will involve a cohort of 22 Year 9 High School pupils (aged around 14 years old) who are about to embark on a Computing GCSE qualification and have little or no programming experience. As this group of participants are still enrolled at High School it is not possible to run the 10 hour version of the workshop previously discussed (due to pupils having other time commitments). Instead, a shortened version of the full workshop (lasting five hours) will be delivered. During this modified version of the workshop pre- and post-workshop questionnaires will be used to collect data.

In order to address the research question this study will be a multiple-case case study:

- *Case 1* will be the novice student programmers who are aged 16-18 years old and are currently in FE. The experiences of 21 students will be considered as part of this case. Case 1 participants will take part in the full two-day workshop.
- *Case 2* will be the trainee ICT/Computer Science High School teachers who all have some programming experience. The experiences of 22 trainees will be considered as part of this case. Case 2 participants will take part in the full two-day workshop.
- *Case 3* will be the novice student programmers who are aged around 14 years old and are currently in High School. The experiences of 22 students will be considered as part of this case. Case 3 participants will participate in a shortened one-day version of the main workshop.

3.5 Case Study Procedure and Roles

LM will deliver the introductory programming workshops and will be the case study leader. PB and TK reviewed an early version of the case study protocol and will continue to be consulted as the study progresses providing research support and advice. The case study protocol has also been evaluated by an independent expert (Barbara Kitchenham of Keele University).

3.6 Ethical Considerations and Participant Code Numbers

Data collected from participants will be stored securely in accordance with the Data Protection Act 1998. Real names, raw, analysed and demographic data will not be associated with any participant. Any identifying features arising from the interviews will be removed during transcription. All data will be exclusive to members of the research team. All participants will receive an information leaflet and informed and written consent forms will be received from all participants. All participants will be given a unique code number. This will be written by participants on all data collection instruments. Keele University's Research Ethics Panel has approved the use of the robot simulator, and workshop, for research purposes.

4. Data Collection

Details of data that will be collected during the case study is provided in this section.

4.1 Case 1: Novice Programmers (FE Students)

Pre-Workshop Questionnaire

A paper based questionnaire will be completed by novices before the workshop in order to determine their past programming experience and attitude towards the subject. An overview of the content of this questionnaire is presented in Table 1.

Novice's Past Programming Experience	Novice's Attitude to Programming	Misc.
Have novices previously programmed	Should programming be taught in schools	Gender
What languages have novices used (if any)	Would novices consider learning to program	
Why novices previously programmed	Problems while learning to program	
Was previous experience challenging	Stereotypes associated with programming	
Enjoyment of previous experience		

Table 1 – Pre-Workshop Questionnaire Overview (Case 1 Participants - Novices)

In-Workshop Programming Exercises

Two paper based programming exercises will be completed during the workshop in order to monitor programming progress. These will draw on concepts that students have encountered.

In-Workshop Researcher Log

The lead researcher (LM) will keep a personal log of incidents or issues that occur during the workshop session according to pre-determined criteria.

Post-Workshop Questionnaire

A second paper based questionnaire will be completed by novices after the workshop in order to gauge their thoughts on the workshop experience. In addition, novices will again be asked about their attitude towards programming. Table 2 provides an overview of this questionnaire.

Novice's Workshop Programming Experience	Novice's Attitude to Programming	Misc.
Enjoyment of session	Programming plans going forward	Gender
Difficulty of session	Has simulator changed perceptions	
Thoughts on effectiveness of simulator	Has simulator dispelled any stereotypes	
Most/least liked aspects of simulator (up to three)	Would novices consider learning to program	
Thoughts on effectiveness of workshop	Should programming be taught in schools	
Comparison of previous programming learning experience (if any) against the workshop		

Table 2 – Post-Workshop Questionnaire Overview (Case 1 Participants - Novices)

Post-Workshop Programming Exercises

Novices will complete four programming challenges that have been constructively aligned with the learning objectives of the workshop session and draw on the concepts encountered. At least 30 minutes will be required for this. These exercises will determine whether *deep learning* has taken place. Case (Case 2008) describes deep learning as when students aim towards understanding whereas *surface learning* is where students aim to simply reproduce material in a test or exam without actually understanding it. Code will be collected and graded according to a three point scoring system:

A) Participant's code shows evidence of deep learning as knowledge gained during the workshop has been used to critically solve a new problem. At least 80% of code is correct.

B) Participant's code shows some evidence of deep learning as the new problem has been attempted and successfully solved in part. Between 50% and 80% of code is correct.

C) Participant's code shows no or little evidence of deep learning as no or little attempt has been made to solve the problem. The participant may have not differentiated between general principles and examples. The participant may have simply tried to repeat information from memory or has merely copied previous code without trying to adapt it to solve the new problem. Less than 50% of the code is correct.

4.2 Case 2: Trainee ICT/Computer Science Teachers

For Case 2 participants (the trainee ICT/Computer Science teachers) different pre- and post-workshop questionnaires will be used to collect data. The procedures for the in-workshop programming exercises, in-workshop researcher log and post-workshop programming exercises, however, remain the same as those described for Case 1 novice programmers.

Pre-Workshop Questionnaire

A paper based questionnaire will be completed by the trainees before the workshop in order to determine their past programming experience and attitude towards the subject. An overview of the content of this questionnaire is presented in Table 3.

Trainee's Past Programming Experience	Trainee's Attitude to Programming	Misc.
Have trainees previously programmed	Should programming be taught in schools	Gender
What languages have trainees used	Confidence teaching programming in school	
Why trainees previously programmed	Perceived difficulty teaching programming	
Enjoyment of previous experience/Identification of concepts previously used		
Was previous experience challenging		

Table 3 – Pre-Workshop Questionnaire Overview (Case 2 Participants – Trainee Teachers)

Post-Workshop Questionnaire

A second questionnaire will be completed by trainees after the workshop in order to gauge participant's thoughts on their workshop experience. In addition, trainees will also be asked about their attitudes towards programming. Table 4 provides an overview of this questionnaire.

Trainee's Workshop Programming Experience	**Trainee's Attitude to Programming**	**Misc.**
Enjoyment of session	Consider using simulator in own lessons	Gender
Difficulty of session	Confidence teaching programming in school	
Thoughts on effectiveness of simulator	Perceived difficulty teaching programming	
Most/least liked aspects of simulator (up to three)		
Thoughts on effectiveness of workshop		
Comparison of previous programming learning experience against the workshop		

Table 4 – Post-Workshop Questionnaire Overview (Case 2 Participants – Trainee Teachers)

4.3 Case 3: Novice Programmers (High School Students)

For Case 3 participants, the same pre- and post-workshop questionnaires will be used as for Case 1. The in-workshop researcher log will also be completed. Due to this workshop being shorter in time, however, the programming tests will not be implemented.

4.4 Additional Data Source: Teacher Interviews

Teacher's thoughts on the robot simulator as a means of introducing programming concepts to novices will also be collected. Themes will include the suitability and effectiveness of the robot simulator as an introductory programming teaching resource. Semi-structured interviews will be used during this process and will be recorded with the consent of the interviewee for later transcription. By the time of the interviews all three teachers will have seen the robot simulator, the workshop sessions and will have discussed with their students about the workshop experience.

5. Analysis

As outlined in Section 4, several sources of data are to be collected during the case study. This will enable the triangulation of collected data which will strengthen the findings of the case study due to it allowing for converging lines of enquiry and corroboration. Triangulation involves taking multiple measures of a studied object and is relevant for qualitative, quantitative and mixed method studies (Runeson et al 2012). Triangulation also helps to address the potential problem of construct validity (discussed in Section 6). An electronic case study database will be used to organise and document collected data. This will be made available to secondary investigators and will help to ensure the transparency of the case study process. A chain of evidence will also be established. Details of how collected data will be analysed during the case study is presented as follows:

- Table 5 provides details of the analysis strategy for data collected from Case 1 (FE Novice Programmers) participants
- Table 6 provides details of the analysis strategy for data collected from Case 2 (Trainee Teacher) participants
- Table 7 provides details of the analysis strategy for data collected from Case 3 (High School Novice Programmers) participants

In regards to the teacher interviews each interview will be transcribed before being thematically analysed for commonalities.

Data Source	Description
Pre-Workshop Questionnaire	Qualitative and quantitative analysis
In-Workshop Programming Exercises	Examination/comparison of participants programming knowledge during the workshop
In-Workshop Researcher Log	Notable events discussed. Common trends identified
Post-Workshop Questionnaire	Comparison with pre-workshop questionnaire results in addition to further analysis
Post-Workshop Programming Exercise	Analysis of participant’s programming progress, and evidence of deep learning, by grading participant’s code according to a pre-determined three point scale

Table 5 – Analysis Strategy for Case 1 Data (FE Novice Programmers)

Data Source	Description
Pre-Workshop Questionnaire	Qualitative and quantitative analysis. Comparison with previously collected data reported in Major et al 2011b
In-Workshop Programming Exercises	For those without substantial Java programming experience analysis of how knowledge progressed during the workshop by examining, comparing and scoring responses. Separate analysis of data collected from participants with considerable Java experience
In-Workshop Researcher Log	Notable events discussed. Common trends identified
Post-Workshop Questionnaire	Comparison with pre-workshop questionnaire results in addition to further analysis. Comparison with collected data reported in Major et al 2011b
Post-Workshop Programming Exercise	For those without substantial Java programming experience analysis of programming progress by grading code according to a pre-determined three point scale. Separate analysis of data collected from participants with considerable Java experience

Table 6 – Analysis Strategy for Case 2 Data (Trainee Teachers)

Data Source	Description
Pre-Workshop Questionnaire	Qualitative and quantitative analysis
In-Workshop Researcher Log	Notable events discussed. Common trends identified
Post-Workshop Questionnaire	Comparison with pre-workshop questionnaire results in addition to further analysis

Table 7 – Analysis Strategy for Case 3 Data (High School Novice Programmers)

5.2 Rival Explanations

It is also intended that a further analytical strategy, examining rival explanations, will be adopted and embedded in the data collection and data analysis stages. Examining rival explanations involves engaging in a systematic search for alternative themes, divergent patterns and rival explanations (Patton 2001). Reporting that a case study sought out, considered and did not find evidence to support a number of plausible rival explanations enhances the credibility of a case study and helps to counter the suggestion that the results are shaped by any predispositions or biases. Yin (Yin 2009) lists many types of potential rivals while Rosnow and Rosenthal (Rosnow and Rosenthal 1997) discuss factors that can impact upon the result of experiments involving human subjects. Several of these issues are considered as potentially relevant to the case study and will be addressed as follows:

Rival Explanation	**Description**	**How to Address**
Null Hypothesis	Observations are the result of chance circumstance only	Workshops to be replicated. Multiple sources of evidence to be used to support findings
Novelty of the Simulator	The novelty of the simulator encourages participants to say they have learnt more than they actually have (i.e. participants confuse interest in the learning mechanism with actual learning)	Is addressed by the scoring process which distinguishes between deep and surface learning on the post-workshop programming exercises
Experimenter Expectation Effect	The scoring of the programming tests is influenced by the experimenter's expectation that the simulator is an effective learning mechanism	Will be addressed by adhering to the marking schedule documented in Section 4.1. A random sample of this data (programming exercises completed by 12 participants) will also be marked by a second member of the research team (TK) according to this schedule. Disagreements in the scoring of these tests will be resolved by consulting the other research team member (PB) and by grading the exercises collected from all participants collectively
"Good Subject Effect"	When participants mark their subjective opinions strongly in favour of the simulator in order to aid the research project and not because it helped them to learn programming	Is addressed in several ways: 1) By the scoring process used to grade participants programming progress 2) By asking participants to identify up to three things they like/dislike about the simulator – participants are likely to be more truthful when identifying positive and negatives than simply answering a question on whether the simulator helped them to learn 3) If far more positives than negatives are reported this would corroborate positive answers to the questionnaire questions related to the effectiveness of the workshop and robot simulator
Implementation Rival	The implementation process (e.g. the nature of the workshop sessions), not the robot simulator, accounts for the results	Is addressed by asking participants to rate (on a five point scale) the effectiveness of the simulator in addition to the effectiveness of the workshop in general. If substantially more participants rate the workshop as effective, and the simulator as ineffective, then the nature of the workshop itself may account for the results of the study

Table 8 – Consideration of Potential Rival Explanations

5.3 Interpretation

During the analysis stage data will be used to address the four propositions as follows:

***P1**. A robot simulator is an effective tool for supporting the learning of introductory programming*

This proposition will be supported if:

- 75% of novices are awarded either an A or B on the post-workshop programming exercise
- The average score of novices on the in-workshop programming tests is greater than 50%
- 50% of novices rate the robot simulator as an effective introductory programming learning tool on the post-workshop questionnaire
- All of the teachers interviewed believe the robot simulator is an effective tool for supporting the learning of introductory programming

***P2**. A robot simulator improves novice's perceptions of programming*

This proposition will be supported if:

- A comparison between novice's pre- and post-workshop questionnaire data shows a positive improvement in regards to participant's perceptions of programming
- All of the teachers interviewed believe the robot simulator helps to improve novice's perceptions of programming

***P3**. A robot simulator offers a more effective introduction to basic programming concepts when compared to other teaching methods*

This proposition will be supported if:

- 50% of trainees who have previously been taught programming believe their previous introductory programming learning experience to be less effective than the one using the robot simulator
- 50% of novices who have previously been taught programming believe their previous introductory programming learning experience to be less effective than the one using the robot simulator
- All of the teachers interviewed believe the robot simulator offers a more effective introduction to basic programming concepts when compared to other teaching methods

***P4**. A robot simulator improves trainee ICT/Computer Science teacher's confidence in their ability to teach introductory programming*

This proposition will be supported if:

- A comparison between trainee's pre- and post-workshop questionnaire data shows a positive improvement in trainee's confidence in their ability to teach introductory programming

The in-workshop researcher logs will be used to ensure that any significant incidents or issues which occur during the study, and could impact upon its findings, are documented according to pre-determined criteria. The logs will not be used to directly address any of the case study propositions.

6. Plan Validity and Study Limitations

In this section measures which have been taken to ensure the validity of the case study, in addition to a consideration of potential limitations of the study, are presented.

6.1 Plan Validity

In order to ensure the rigour and reliability of the case study several measures have been taken. Firstly, as documented in Appendix A, this protocol has been designed after considering Per Runeson and Martin Höst's case study design checklist (Runneson and Höst 2008).

Secondly, as suggested by Yin (Yin 2009) in order to ensure construct validity, multiple sources of evidence (pre- and post-workshop questionnaires, in and post-workshop programming tests, in-workshop researcher log and teacher interviews) and the establishment of a chain of evidence (plans to make available a database for secondary researchers, the final report to refer heavily to collected evidence and the protocol procedures to be followed and deviations documented) are to be used.

In regards to internal validity, a pre-identification of potential rival explanations (see Section 5.2) coupled with the adoption of a data collection and data analysis strategy which actively investigates these rivals helps to ensure internal validity is established.

As Case 1 and Case 3 participants are aged between 14 and 18 years old, and do not all come from the same educational institution, it is believed that the results of the study will be generalisable to a similar demographic of novice programmers. Moreover, despite all Case 2 participants being enrolled on a Teacher Training Course (PGCE) at Keele University the programming backgrounds of participants are significantly varied. As such it is considered that the results of the study will be generalisable to a similar demographic of trainee ICT/Computer Science teachers. As the case study protocol has undergone expert review, in addition to peer review, the risk of unidentified threats to the validity of the study are considered to have been minimised.

6.2 Study Limitations

Aside from the programming exercises, other instruments that will be used during the case study will collect data that is self-reported (i.e. cannot be independently verified and what participants say in interviews and questionnaires has to be taken at face value). This may lead to sources of bias such as selective memory and exaggeration. It is intended that the use of open and closed questions (to avoiding ‘leading’ participants) and reinforcing the anonymous nature of the study will help to reduce the potential impact of self-reported bias.

Another possible limitation of the case study is that the interviewees and student participants will be self-selected. Indeed, in Major (Major 2012) it is described how some potential learner participants chose not to be involved in a study after they were approached. There is a risk that a similar occurrence during the case study could result in some data being excluded from the final report. By inviting a broad selection of participants to take part in the research (which in total will number over 65), however, it is predicted that this risk has been minimised.

7. Summary

In this paper a protocol which provides details of a case study that will investigate the use of a robot simulator as an introductory programming teaching tool has been presented. Such research is being undertaken as a Systematic Literature Review indicated that this work would be valuable. The development of a case study protocol in advance of the main study will help to ensure that reliable, transparent, targeted and rigorous work is performed. Furthermore, potential problems which may affect the study have been considered and accounted for in advance of its implementation. This protocol provides background information, details of the planned study design, information about the strategies for data collection and data analysis in addition to a consideration of factors which could affect the validity of the study. This protocol may also act as a point of reference for other researchers interested in performing a case study.

References

ACM/IEEE Interim Review Task Force, 2008. Computer Science Curriculum. ACM/IEEE.

ANSI Standards Committee on Dental Informatics, 2001. Working Group Educational Software Systems Guidelines for the Design of Educational Software.

Beale, R. and Sharples, M., 2002. Design Guide for Developers of Educational Software. British Educational Communications and Technology Agency, 1. pp. 1-29.

Becker, B.W., 2001. Teaching CS1 With Karel the Robot in Java. In SIGCSE '01: Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education, pp. 50–54. ACM.

Borge, R., Fjuk, A. and Groven, A.K., 2004. Using Karel J Collaboratively to Facilitate Object-Oriented Learning. In ICALT 2004, pp. 580–584.

Brereton P., Kitchenham B., Budgen D. and Li Z., 2008. Using a Protocol Template for Case Study Planning. Proceedings of EASE 2008, BCS-eWIC.

Buck, D. and Stucki, D.J., 2001. JKarelRobot: A Case Study in Supporting Levels of Cognitive Development in the Computer Science Curriculum. In Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education, New York, NY, USA, pp. 16–20. ACM.

Case, J., 2008. Education theories on learning: an informal guide for the engineering education scholar. Higher Education Academy Engineering Subject Centre, Loughborough University. Accessed Online: http://hdl.handle.net/2134/9730 (2nd May, 2012).

Enderle, S., 2008. Grape: Graphical Robot Programming for Beginners. In Research and Education in Robotics EUROBOT, 33, pp. 180–192. Springer Berlin Heidelberg.

Fagin, B., 2003. Measuring the Effectiveness of Robots in Teaching Computer Science. In 34th SIGCSE Technical Symposium on Science Education, pp. 307–311. ACM.

Kelleher, C. and Pausch, R., 2005. Lowering the Barriers to Programming: A Taxonomy of Programming Environments and Languages for Novice Programmers. ACM Computer Survey, 37 (2), pp. 83–137.

Klein, H. K. and Myers, M. D., 1999. A set of principles for conducting and evaluating interpretive field studies in information systems. MIS Quarterly, 23(1):67.

Ladd, B. and Harcourt, E., 2005. Student Competitions and Bots in an Introductory Programming Course. J. Comput. Small Coll., 20 (5), pp. 274-284.

Lauwers, T., Nourbakhsh, I. and Hamner, E., 2009. CSbots: Design and Deployment of a Robot Designed For the CS1 Classroom. In Proceedings of the 40th ACM Technical Symposium on Computer Science Education, New York, NY, USA, pp. 428-432. ACM.

Lee, A. S., 1989. A scientific methodology for MIS case studies. MIS Quarterly, 13(1):33.

Lemone, K.A. and Ching, W., 1996. Easing into C++: Experiences with RoBOTL. SIGCSE Bull., 28 (4), pp. 45-49.

Major, L., 2012. An evaluation of the Advanced Diploma from the Perspective of Staff and Learners. Research in Post-Compulsory Education. Volume 17, Issue 1, March 2012.

Major, L., Kyriacou, T. and Brereton, O.P., 2011a. Systematic Literature Review: Teaching Novices Programming Using Robots. In 15th International Conference on Evaluation and Assessment in Software Engineering (EASE 2011), Durham University, UK, 11 - 12 April 2011. IET. pp. 21-30.

Major, L., Kyriacou, T. and Brereton, O.P., 2011b. Experiences of Prospective High School Teachers Using a Programming Teaching Tool. In Proceedings of the 11th Koli Calling International Conference on Computing Education Research (Koli Calling '11), Koli National Park, Finland, 17-20 November 2011. ACM, New York, NY, USA. pp. 126-131.

Major, L., Kyriacou, T. and Brereton, O.P., 2012. Systematic Literature Review: Teaching Novices Programming Using Robots. In IET Software (To Appear).

Martin, C. and Hughes, J., 2011. Robot dance: edutainment of engaging learning. Proceedings of the 23rd Psychology of Programming Interest Group, PPIG (2011), York, UK, 6-8 September.

McGill, M. M., 2012. Learning to program with personal robots: Influences on student motivation. ACM Trans. Comput. Educ. 12, 1, Article 4 (March 2012).

McWhorter, W.I. and O'Connor, B.C., 2009. Do LEGO Mindstorms Motivate Students in CS1? In SIGCSE '09, pp. 438-442. ACM.

Patton, M., 2001. Qualitative Research and Evaluation Methods (2nd Edition). Thousand Oaks, CA: Sage Publications. p. 553.

Robson, C., 2002. Real World Research (2nd Edition). Blackwell.

Rosnow, R. and Rosenthal., R., 1997. People studying people - Artefacts and Ethics in Behavioural Research. W.H. Freeman and Co, New York.

Runeson, P. and Höst, M., 2009. Guidelines for conducting and reporting case study research in software engineering. Empirical Softw. Engg. 14 (2), pp. 131-164.

Runeson, P., Höst, M., Rainer, A. and Regnell, B., 2012. Case Study Research in Software Engineering: Guidelines and Examples (1st Edition). Wiley.

Sartatzemi, M., Xinogalos, S. and Dagdilelis, V., 2003. An Environment for Teaching Object-Oriented Programming: objectKarel. In Proceedings of ICALT 2003, pp. 342-343.

Squires, D. and Preece, J., 1999. Predicting quality in educational software: Evaluating for learning, usability and the synergy between them. Interacting with Computers, 11 (5), pp. 467-483.

TIOBE Programming Community Index for April 2012. Accessed Online: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html (24th August, 2012).

Yin, R.K., 2000. Rival explanations as an alternative to reforms as experiments. In L. Bickman (Ed.) Validity and social experimentation: Donald Campbell's legacy. Thousand Oaks, CA : Sage.

Yin, R. K., 2009. Case Study Research: Design and Methods (4th Edition). Sage.

Appendix A: Case Study Design Checklist

Item	Checklist Question	Comments
1	What is the case and its units of analysis?	See Section 3 'Design'
2	Are clear objectives, preliminary research questions, hypotheses defined in advance?	One main research question (see Section 3.1) and several propositions (see Section 3.2) have been outlined
3	Is the theoretical basis - relation to existing literature or other cases - defined?	Results of a previously completed SLR (Major et al 2011a, Major et al 2012) provide the basis for this study
4	Are the authors' intentions with the research made clear?	The purpose of the study is to determine whether a robot simulator is an effective tool for supporting the learning of introductory programming (see Section 3.1 'Aim')
5	Is the case adequately defined (size, domain, process, subjects…)?	See Section 3 'Design'
6	Is a cause–effect relation under study? Is it possible to distinguish the cause from other factors using the proposed design?	See Section 5.2 'Rival Explanations' and Section 6.1 'Plan Validity'
7	Does the design involve data from multiple sources (data triangulation), using multiple methods (method triangulation)?	The case study design involves collecting multiple forms of data using multiple data collection methods (as detailed in Section 4 'Data Collection'). Collected data will be triangulated as outlined in Section 5 'Analysis'
8	Is there a rationale behind the selection of subjects, roles, artefacts, viewpoints, etc.?	Yes. This is described throughout the protocol document
9	Is the specified case relevant to validly address the research questions	Expert and peer review of the protocol, use of multiple sources of evidence and the establishment of a chain of evidence help to overcome potential issues with construct validity
10	Is the integrity of individuals/organisations taken into account?	This factor is recognised in Section 3.6 'Ethical Considerations and Participant Code Numbers'