ATHABASCA UNIVERSITY

TEACHING COMPUTER PROGRAMMING USING EDUCATIONAL ROBOTS

BY

RON PATTERSON

A project submitted in partial fulfillment

Of the requirements for the degree of

MASTER OF SCIENCE in INFORMATION SYSTEMS

Athabasca, Alberta

June, 2011

DEDICATION

I would like to dedicate this essay to my lovely wife Mylène, and our children Samantha and Brandon who have stood faithfully by me throughout this long process, and to my parents Bruce and Shirley who have always been there for me.

ABSTRACT

This paper investigates the use of educational robots to teach introductory computer programming. In preparing this report I studied over 50 research papers and read many articles on both educational robots and their use in computer programming. Of these, I have selected 22 of the most appropriate to form the basis of my research. The main goal of my research was to determine the effect that the use of robots would have on student learning, interest, attraction to programming, and drop-out rate. After compiling the results from these papers, I have found that the robots do have a positive impact on the student's interest in computer programming. The results in the other three areas are more varied. Many of the researchers observed an increase in student learning when using the robots but a few found no difference or even a lower rate of learning while using the robots. According to the researchers, the principal reasons for the unfavorable results were that the students did not get to spend enough time with the robots, the robots were prone to errors or malfunctions, and the turn-around time for the write, test, debug cycle was too long using a robot versus a regular computer and IDE. The researchers put forth several solutions to these problems. These include having one robot per student that they could take home with them, using emulators when the robots are unavailable, and keeping spare robots and parts to replace non-working ones. These along with more teacher preparation and combining the robot lessons with traditional lessons were proposed as solutions that could increase student learning and reduce student attrition.

ACKNOWLEDGEMENTS

This essay is dedicated to my lovely and patient wife Mylène who stood faithfully by me throughout the two years that I worked on my Masters degree. Although the road was long and the workload heavy at times, I always knew I could count on her to pick up the slack around the house when I had to work late on assignments or study for exams. Thank you so much Mylène. I could never have done it without you. I would also like to dedicate this essay to our two children, Samantha and Brandon, and to my parents, Bruce and Shirley, who have always supported me fully in all my endeavors. Finally, I would like to acknowledge my supervisor Dr. Richard Huntrods who inspired and assisted me in the writing of this essay.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Statement of Purpose

The purpose of this research paper is to determine the effect that the use of robots in computer programming courses have on student interest, learning, or retention. Since suitable robots can be purchased for as little as $300 each and can support all of the more popular programming languages, the door has been opened for the adoption of this technology in the classroom. However, many teachers resist adopting them because they have no conclusive research on the subject or they are not sure what benefits they can expect if they adopt them. Furthermore, schools usually require a cost/benefit analysis of new acquisitions before investing in them, and without any supporting research, this task is extremely difficult to complete. Therefore, my goal is to study and summarize the recent research on this subject to help inform teachers of both the benefits and disadvantages of using this technology. As a result, I hope to make it easier for teachers to make the decision as to whether or not they want to adopt robots in their classroom.

Of course, the idea of using robots to teach computer programming is not new. In fact, ever since LEGO introduced their first personal robot in 1998, teachers have been experimenting with them in the classroom. This interest has also resulted in a lot of research being done on the subject, covering a variety of hypotheses. Of these, the more commonly studied are the effects that the robots have on student interest, student learning, student enrollment, and student retention.

During my discussion of this topic I intend to concentrate on three different areas. The first area I want to discuss is the effectiveness of using robots in computer programming courses. In other words, how does the use of robots in the classroom affect the learning of the students? For this part I intend on relying completely on previously documented research as I will not be conducting any experiments of my own for this essay. The second subject that I want to discuss is the different ways in which the robots can be introduced to make it easier for both the student and the teacher to embrace this new technology. Again much information on this subject can be summarized from the research papers in the area. In the majority of cases, the research papers point out items that may have affected the outcome of their research. This information usually outlines problems that the researchers encountered while using the robots and includes suggestions as to how they may be remedied for future work. Thus, assuming that a teacher decides to go ahead with the idea of using robots in the classroom, this information can help them do so more effectively. Finally, I will look at the different types of educational robots that are available and the programming languages that can be used with them. During this discussion I will not only look at the different choices available but also try to suggest which of the robots and languages are more suited to this type of project.

Research Problem

The problem addressed in this research paper is whether or not there is an advantage to using autonomous educational robots in the teaching of computer programming. My research will use five different criteria to answer this question. They are:

1. Do the students learn more or less when using robots versus the conventional teaching methods?
2. Are students more interested in programming with robots than without?
3. Does using robots attract more students to the courses than when using conventional programming alone?
4. Does using robots attract a different type of clientele than in traditionally taught courses?
5. Is there a difference in retention rate in courses where robots are used versus those where only conventional teaching methods are used?

I anticipate the following positive outcomes from my research essay:

- A description of the different educational robots available.
- A summary of the programming languages that can be learned using educational robots.
- A summary of the research that either supports or contradicts the use of educational robots as a teaching tool for computer programming.
- A description of the different methodologies and environments that can be used to teach computer programming using robots.

The following are the possible negative outcomes from my research essay:

- The research on the subject may be inconclusive making it difficult for me to make a recommendation either in support or against the teaching of computer programming using robots.
- There may be areas of the topic that have not been researched sufficiently thus making it difficult for me to draw conclusions in that area.
- Since I am not performing any experiments in the area myself, there may be areas of research that I would like to discuss that I may not be able to.

Definition of Terms

Conventional teaching method – when applied to computer programming this means that students create their programs individually on a computer in an IDE built expressly for this purpose. The students are able to quickly compile, run, and view their results. These programs are usually small in nature and print simple text or graphics on the screen.

Teaching method using robots - when applied to computer programming this means that students create and compile their programs on their computer, and then they must download the program into the firmware of the robot to see the results of their program. The turnaround time is therefore much longer between consecutive program runs. Students usually work in teams and must do more debugging without the benefit of running the programs since they usually have limited access to the robots.

Firmware - Computer instructions that are downloaded into the robot to make it perform some action. These instructions are written and compiled on a computer and then downloaded to the robot once compiled.

Autonomous educational robot – A small, inexpensive robot that is meant to be programmed and controlled through software written on a personal computer. The LEGO Mindstorms NXT is one of the most popular educational robots on the market today.

LEGO Mindstorms - An educational robot sold by LEGO that the student must build and then program using a personal computer and one of several computer languages. Programs are written and compiled on a personal computer and then downloaded to the robots firmware. The robot then runs the program contained in its firmware and performs the actions indicated by the computer program.

Computer language - A language that is used to give instructions to a computer or robot. Computer programmers write instructions to the computer in a computer language, after which the program is compiled into machine readable format that can be run on the computer or robot.

Organization of Remaining Chapters

This research paper has 4 chapters following this one.

Chapter 2 discusses the previous research done in the area of educational robots in computer programming courses. In this chapter I take a look into the history of the LEGO Mindstorms robot, which is the principal robot used in most research into this field. Following this I look at the different theories that have been tested by the previous research, the methodologies they used to test these theories, and the results they obtained. I also discuss both the advantages and disadvantages that the previous research has mentioned to the use of robots in introductory programming courses.

Chapter 3 discusses the methodology that I used to perform the research for this paper. The purpose of this chapter is to discuss my research methodology and the way that I reached my conclusions so that another researcher, using the same method, could verify and confirm the results of my research.

Chapter 4 explains the results of my research by listing each of the research papers that I have studied, and discussing the conclusions of each. Since the results of the research on this topic are mixed, there are both negative and positive research papers in my references. To make it easy for the reader to follow I have divided the discussions into these two camps. First I list and discuss the research papers that have a negative bias on the use of robots in programming courses. Following this I discuss the papers that have a more positive attitude towards the subject. After reading this chapter, the reader will come away with a very good idea of both the positives and negatives of using robots in the classroom.

Chapter 5 discusses my final conclusions on this research topic, the recommendations I have, and the areas that I believe should be researched further. My final conclusions are based on the conclusions drawn by all 25 of the research papers that I reference as well as many others that I have researched while writing this paper. Likewise the recommendations have been extracted from similar recommendations made by authors of the research papers on the subject. The areas of further research indicate areas that I believe are lacking in the current available research, and that could help made the use of robots in the classroom more successful if it was available.

CHAPTER 2

PREVIOUS RESEARCH

Introduction

Since the introduction of the educational robots in the late 1990's, much research has been performed on their effectiveness as a teaching tool for computer programming courses. In this report I have cited 19 different research papers on the subject. Of these, five reported a negative or mixed view on the use of robots in the classroom. Barry S. Fagin and Laurence Merkle reported in [1] that students in robotics section of their courses scored lower than those in the regular section during a one year study of 938 programming students. Daniel C. Cliburn reported in [4] that only one out of five courses that he taught using robots could be considered successful. He did point out however, that part of the reason for the failure in the other courses was due to mechanical problems or lack of access to the robots. In [10] and [13] the authors report mixed results in their research using robots, with the consensus leaning towards the negative. Finally William Isaac McWhorter reports in his doctoral thesis that there was no significant difference between the robotics group and the regular group in his study. Based on his hypothesis that the robotics students should show better results than the control group, he considers this result to be a negative one.

Of the 19 research papers that I have cited, 14 of them have reported positive results. In [2], Chuck Leska summarized the results of a programming course that was taught at Randolph-Macon College. Faced with falling enrollment, the college implemented a programming course using robots for the non-major students. According to the author, the results were very positive, with a very high enrollment and a very positive reaction from the students. Both David A. Gustafson and Michael W. Lew [3][11] reported on the use of robots in more advanced programming courses with similar success. In both cases the students were enthusiastic and learned at the expected pace. Debra Burhans was the author or coauthor of two of the papers I cited [5][6]. In both studies the authors surveyed the students with questionnaires after the course was done. Overall the students responded very positively to the robots and to the programming course. Pauline Mosley and Rickard Kline also used student questionnaires to gage the effectiveness of their robot programming course in [14] with similar positive results.

Positive results were reported by Eric Wang in [7] and the authors of [8] who reported that the students were both motivated and successful in the courses using the robots. The authors in [9] took a slightly different approach by adopting the Scribbler robot rather than the Lego Mindstorms. In addition to this they supplied each student with their own robot. The idea behind this was to ensure that the students had ample access to the robots in order to do their work. They report that this approach worked very well with the students very enthusiastic about the course and the robots. The authors in [12] did not actually do any research themselves but instead summarized past research in the field. Their overall consensus however, was that based on the past research that they studied, robots did have a very useful place in the classroom.

In [15] the authors were more cautious in their conclusions about the robots. They did give a positive report but provided a set of conditions that should be met in order to obtain similar results in your course. Many of these ideas I have incorporated in the recommendations section of this paper. Two ideas keep recurring in the research is that of robot challenges and teamwork to inspire and teach the students. The authors in [16] used both of these methods to teach their students. All the students were divided into teams at the start of the course. They then had 6 lessons on robot programming followed by a robot challenge. The author reports positive results from this experience mentioning that the students were more apt to relate programming to real world experiences when using this method.

Finally, in [18] and [19] the authors taught robot programming to their students for a short period. In [18] the students had two weeks of instruction while in [19] they had only 6 hours each. In both cases however, the results were positive with the students reporting that they liked using the robots and found them very instructional.

Background

Most of the recent research on the use of educational robots for computer programming has focused on the LEGO Mindstorm robot. This robot was invented in the late 1990's in partnership with MIT to encourage people, both old and young to learn more about technology and computer programming. To this end it has been greatly successful. Almost immediately, hobbyists and schools all over the world jumped on the bandwagon and began to buy, build, and program the robots. The earliest LEGO robot was called the RCX which came with a programmable brick, 2 servo motors and several different sensors. In the same vein as all LEGO products, the robot came in pieces and had to be built by the user. This allowed the user to create many different varieties of robots with different features and abilities. At the heart of every robot however was the LEGO brick. This is the part of the robot that contains the firmware that allows the user to download programs to the robot and control it. When delivered the LEGO came with its own graphical programming language. This language was very easy to use for beginners and non-programmers, but many users found this language too simplistic to do any real computer programming. It quickly became apparent that if the LEGO Mindstorm was to gain any real mainstream support, it would have to support the more traditional high level languages that were used in industry by professional programmers. As a result, developers soon developed firmware that could support many of the popular programming high level languages such as Ada and C.

In 2006 LEGO introduced the Mindstorm NXT followed in 2009 by the Mindstorm NXT 2.0. This most recent robot product features 3 servo motors, sensors for touch, light, sound, and distance, and of course the programmable brick at its center. It also provides the user with instructions to create 4 different varieties of robots using these pieces. LEGO also is delivered with a graphical programmable language that is both simple and fun to use. As with the RCX, developers have developed firmware for the NXT 2.0 that supports many of the high level languages. Of these, C and Java are two of the most popular languages used to program the NXT 2.0 robots today.

Of the research papers that I have included in my report, all but one use the LEGO Mindstorm robot. Of these, approximately half used the Mindstorm RCX and half used the Mindstorm NXT. Of course, the robot that was used depended primarily on when the research was done. Research done between 1998 and 2006 used the RCX since that was the only robot that existed at the time. Robots after that date tended to use the NXT since that was the newest and best robot that existed at that time. In both cases the research did not differ significantly in terms of methods or findings. That is to say, it did not matter which version of robot that was used, the findings were essentially the same. Therefore, in my report I will not differentiate between the different versions of the robots but instead will look at the results as if all were programmed using the same robot.

Hypotheses

The principle hypotheses that the previous research proposed and tested were that the use of robots to teach computer programming when compared to traditional teaching methods should:

1. Increase student learning and comprehension of computer programming
2. Increase student interest in computer science and computer programming
3. Increase student enrolment in post-secondary computer science programs
4. Reduce student dropout rates of computer science programs
5. Attract more females to computer science programs
6. Give students a better understanding of real world computer science situations

Each of the papers in my study had their own hypotheses that they tested, covering at most 2 or 3 of those above. However, by looking at the research as a whole, it is possible to extract information about each individual hypothesis and to draw conclusions based on this. Therefore the main purpose of my research study is to study each of these hypotheses in relation to all of the research that I have included in my reference section.

Methodologies

The methodologies of the research can be divided into two different areas. The first area was the teaching methodology, or the way in which the researchers attempted to teach the students the programming language. The second methodology was the way in which the researchers tested to the students to determine how well they had learned the material and how well they liked programming the robots versus more traditional programming tasks.

All of the research projects provided students with robots, a programming environment, and a series of programming tasks to solve using the robots. These tasks were often done in teams due to the limited number of robots available, and they were usually performed during laboratory hours since it was inconvenient to have the students remove the robots from the premises. This provided the primary teaching method for the research projects. In the cases where the researchers tested the robot group against a control group who learned using traditional programming there was a second laboratory section that did not use robots.

Student testing was done in several different ways. First the researchers observed how well the students were able to program the robots. This involved observing the robots during the assigned programming tasks and in some cases it involved the evaluating the performance of the students in robot competitions with other student teams. They also tested the students to see how well they learned the programming languages and constructs using standard tests and exams. Finally they interviewed the students or gave them questionnaires to find out the students' impressions on using the robots to learn computer programming.

Discussion

The reason I decided to write this research paper was to determine if the use of educational robots in the classroom would be a recommended teaching methodology. After performing the research and compiling the results I have found they these results are positive, with 14 of the research papers I cited favorable towards the use of robots and 5 reporting unfavorable results. However, out of the 5 that reported unfavorable results, both Daniel Cliburn in [5] and William McWhorter in [20] had some positive results within their research. In [5] Daniel Cliburn used robots in 5 different courses. Of those one was a success with very positive results. He has reported negative results for the other four courses, however, technical difficulties with the robots and problems with group work methods were pointed out as two of the big difficulties during these courses. Both of these items are related to course organization more than the method of instruction and thus could easily be resolved and may in fact lead to a more positive experience in future research.

In [20] William McWhorter tested one control group in the fall using the traditional teach methodology and one experimental group in the winter session using the robot methodology. He gave identical tests and questionnaires to both groups and used the comparison of one group to the other for the basis of his results. The results showed no real statistical difference between the two groups in terms of student motivation, learning, or learning methods. This is a negative result if we expect the robot groups to perform better than the traditional methods, but it is a neutral result in the sense that the author could not prove either method superior. This means that if a teacher does go ahead and teach a programming course using robots, then according to this research, they should obtain similar results to groups using traditional methods.

Of the 14 papers I cite with positive results, all reported that the students were very enthusiastic to use the robots. In fact, this seems to be one of the big attractions of the robots, as all of the positive papers and even three of the negative papers report this observation. Four of the papers report that the students learned as well or better using the robots than without [2][3][7][9]. The use of robots can also be used to increase enrollment in computer programming courses. Both Chuck Leska and Eric Wang reported that the use of robots increased enrollment in their courses [2][7]. In the case of Eric Wang, he reported that enrollment had more than doubled in only three years. He also reported that the use of robots in his course was excellent medium for teaching design, creativity, and programming [7]. In terms of student feedback, five of the research papers reported on the use of student questionnaires and surveys at the end of their courses [5][6][14][16][20]. In every case the responses of the students was overwhelmingly in favor of the robots.

Therefore, the results of the research I have cited is favorable in relation to the robots. 14 out of 19 research papers have reported positive results, with 5 reporting mixed or negative results. In addition to this, student reaction to the robots was very positive. This is confirmed both by teacher observation of the students reactions in class as well as by student questionnaires and surveys that were used in the 5 research papers mentioned above.

Advantages reported

According to the research, teaching computer programming with robots has several advantages. Almost unanimously the researchers found that students were much more enthusiastic in programming courses that used robots compared to those that used traditional methods of programming instruction. This was confirmed through student interviews and questionnaires. For example, Eric Wang in [7] pointed out that enrollment nearly doubled in the programming course that he studied using robots, compared to declining enrollments in other programming courses. Many of the other researchers had similar experiences.

It was often observed that robots have an appeal to students and are considered interesting and fun to work with. Myles McNally points out in [13] that due to the large number of supported languages for the LEGO Mindstorm that it can be used for real programming. He also points out that there are a lot of resources available on the internet to help out student programmers and that the robots are very reasonably priced and can be afforded by most schools. He also points out that using team projects can reduce the cost since the robots can then be shared among a group of students. Finally he indicates that the robot is better than a simulator due to its tactile nature, immediate feedback and student appeal. This opinion is shared by several other researchers as well.

Since most of the research required that the students work in teams, it was also observed in [16] that the projects taught the students teamwork and communication skills, something that working on individual projects does not do. Finally, it was pointed out by Jerry Schumacher, Don Welch, and David Raymond in [8] that the use of robots in the classroom exposes the students to concepts about embedded programming and robotics that they would otherwise not see.

Disadvantages reported

Many of the disadvantages that are reported by the researchers are echoed by the authors in McNally, Goldweber, Fagin, and Klassner [13]. The first of these disadvantages is the cost. As the authors point out, each robot costs around $300 which is too much to supply each student with their own robot. It is also too expensive to expect students to buy their own robots. The result of this is that the students are usually forced to do their programming assignments in teams and wait for their turn at using the robot. This slows down the learning process as each student has less time to test and debug their program. It also means that the student is unlikely to own their own robot, so they will be limited to working on program assignments during laboratory hours. This reduces significantly the time the student has to work on assignments since they cannot do them at home.

The authors also argue that the computer commands available to manipulate the robot are limiting and thus prevent the teaching of all the computer concepts that are required in a programming course. It also prevents the teaching of more advanced courses on the robots therefore there is a lack of continuum in the teaching methods. Since the robots are operated on batteries and are not a precise device, they are also prone to errors and variations in behavior over time. For example, a right turn may be 93 degrees when the batteries are fully charged but only 87 degrees once they are half dead.

The final disadvantage was mentioned by Barry S. Fagin and Laurence Merkle in [1] and William Isaac McWhorter in [17]. Both of these performed experiments on the learning of the students using the robots versus traditional programming using one group who learned with the robots and one who learned the traditional way and then compared them. Both studies reported a lower level of learning in the robot group versus the control group. They also reported that there was no significant difference in student interest using the robots and that the students using robots were no more likely to choose computer science as a major over those that learned using traditional programming. Although this is not a disadvantage versus using the robots, it is presented as a disadvantage by the authors who assume in this case that mounting a course using robots is more costly and time consuming than creating a traditional programming course and that a neutral result leans in favor of the traditional programming method.

Results

The results of the research into the use of robots to teach computer programming are inconclusive. Many of the papers have reported increased student interest and involvement in the courses where robots were used. On the other hand, in [1] Fagin and Merkle compared the results from over 800 students on identical tests from both robotics and non-robotics based laboratory courses and concluded that the non-laboratory students had learned less and scored lower on the tests. They also found that there was no difference in the choice in major between the two groups. That is to say, the students in the robotics laboratory courses were no more likely to choose computer science as their major than those in the non-robotics group. Even though these results appear negative towards the use of robots, they are not as definitive as they first seem. The authors in this study pointed out that the students in the robot section only had access to the robots in the laboratory and thus could not work on their assignments back in their rooms. Meanwhile the students in the non-robot section were able to work on their assignments during lab hours and at home on their own computers. This may have had an impact on the results. The authors mention that if the students had owned their own robots or could have borrowed the robots from the lab and brought them home, it might have increased their learning and test results. They also mentioned that the instructors may have had an impact on the results as well. Some of the instructors for the robot laboratories had less than one year’s experience teaching using the robots which may have impacted the students’ learning. They also mentioned that they noticed that test results were higher in those laboratory courses where the instructors were more experienced with the robots which may confirm these suspicions.

A second research project by William Isaac McWhorter compared the students taking part in a traditional introductory programming course to those in a programming course using the LEGO Mindstorms robots. The results of this study also indicated no significant difference in the motivation, ability to learn, or the amount learned between the two groups. Again the author mentioned several factors that may have affected the results. The students using the robots only had access to them during laboratory periods and could not work on assignments at night. Also the robot course was given in the spring session while the non-robot section was given in the fall. It could be that students perform differently during different sessions. Also was the sample size large enough in this experiment? The data was only collected for one classroom of students for each section.

Only two of the research papers I studied compare the results of the robot group with a control group [1][20]. In all the other cases, the entire class was obliged to use the robots and then the students were tested to see how well they learned and questioned to see how well they liked the course. Most of these studies showed that the students did learn the material adequately and were very enthusiastic about using the robots. Eric Wang pointed out in [7] that enrollment more than doubled in the course using robots versus those using traditional methods. Many other researchers reported an increase in interest and enthusiasm from the students when using the robots versus traditional programming laboratories. Therefore, in my opinion the signs are positive. While there are definitely some problems to iron out, the potential for the successful use of robots in introductory computing courses does seem to be supported by the research if the right format is chosen.

CHAPTER 3

METHODOLOGY

Background

My research paper investigates the use of small autonomous robots to teach computer programming. Suitable robots have been available since the late 1990's and thus there have been hundreds of research papers written on the value of using them for computer programming instruction. However, since the accuracy of my research project depends directly on the quality of the research papers that I have analyzed during the preparation of the project, it was imperative that these research papers be carefully chosen.

Research Method

In light of the fact that I am performing my research in a remote location, far from any university or scientific libraries, I had to rely completely on the internet for my research. However, this did not turn out to be a disadvantage since there are several good internet sites available that have digitized versions of much of the recent research in this field. In fact, simply searching these sites for the term "computer programming with robots" generated hundreds of pertinent results. Since my goal was to choose from 25 to 35 references in total among the hundreds of articles that are available, a large part of my preparation was in searching the internet for suitable research articles and then retaining or discarding them depending on the criteria I had established. A list of the web sites that I used to locate the research articles is shown below in Table 1.

Table 1 – Web sites used to find research articles

IEEE.org	http://www.ieee.org/index.html
ACM digital library	http://portal.acm.org/
Google search engine	http://www.google.ca/

These web sites were selected based on the quality and number of research papers that they contained. The IEEE and ACM websites are two of the most important sites containing peer reviewed scientific research papers and both are highly recommended by my university and course instructors. The Athabasca University Library website also has an extensive list of research papers and also acts as a portal to several other sites containing scientific journals. Google was selected to complete the list since it is one of the most complete search engines on the internet, and because it now supplies in its search results, research papers from most of the foremost peer reviewed scientific journal sites. I am therefore confident that by using these four websites that I have access to most if not all of the most important research papers on my topic.

Selection Criteria

To filter through the hundreds of search results that were returned by the search engines, it was necessary for me to establish a set of criteria to ensure that only the most appropriate research papers were retained for my references. For the search engines that I used, the most appropriate results were usually displayed in the first 5 -7 pages of the search results. Afterwards, the papers were either repeats of the previous search results or papers that were not specifically on the topic searched for. Therefore I concentrated most of my searching in this area. For any article whose title looked appropriate to my research, I first read the abstract. This gave me a good idea if the research was in the field I was interested in. If it was, I then downloaded the .pdf file to my computer for later reading. Once I had exhausted all the articles that I could find in this manner I then moved on to the second phase of my reference list check.

In this phase I read each of the articles and deleted any that did not meet my specific criteria for this research paper. Table 2 below shows the criteria that I considered when choosing my references.

Table 2: Criteria used when choosing my references

Pertinence to my research question	Research problems indicated
Recent research paper	Results shown
Peer reviewed	Conclusion indicated
Available at a reliable scientific site	Suggestions indicated
Research methodology indicated	Has a solid reference list

The first criteria that I used when choosing references was the pertinence of the research compared to my topic. My preference was to select research papers that studied the teaching of introductory computer programming using small educational robots. While searching the research archives, it became apparent that most of the research had been done using the LEGO Mindstorms Robots and as a result, the majority of my references also use this robot. Due to the large amount of research that has been done on this topic, I was able to restrict myself to articles that had this very narrow focus and still amass a good selection of references.

In addition to this I also preferred to use peer reviewed papers whenever possible. This was made much simpler by the fact that the two principal web sites that I used in my research, notably the IEEE and ACM websites, contain mostly peer reviewed work. Therefore my first approach was to search for papers on these sites first, rather than use Google which returns results to all types of material. When I did use Google, I was very restrictive as to what I selected, usually limiting myself to links to peer reviewed articles at established scientific sites.

My preference was also to use research papers that clearly indicated the method of the research, the problems or pitfalls of the research methodology, the author's conclusions, and suggestions as to what might done differently to get a better result. Finally I preferred papers written within the last 10 years, although in a couple of cases I kept older ones that were very pertinent to my research area.

Search Criteria

To help other researchers locate similar articles using the web sites that I proposed I have included in Table 3 below a list of the search criteria that I used when searching these sites. It is expected that if the reader enters the following search strings in the four search engines that I have mentioned, that they will be able to reproduce similar search results to those that I found while performing my research.

Table 3 – Sample search strings used to find the research papers

Teaching computer programming using robots	LEGO Mindstorms in the classroom
Using robots to teach computer programming	Robots in the classroom
LEGO Mindstorms to teach computer programming	Educational Robots
Robots in Education	LEGO Mindstorms
Programming with robots	Teaching using Educational Robots

Of particular note is that “LEGO Mindstorms” is mentioned three times in my search strings. I noticed while doing my initial title searches that most of the research in this field has used the LEGO Mindstorms robot as the robot for their study. In addition to this, the words “LEGO Mindstorms” is often mentioned in the title of the paper. Therefore, by searching for this word combination I was able to find many research papers suitable as references for my essay. It should be noted though, that a few of the articles did use alternative robots, however searches on these robots turned up very few results due to the high popularity of the Lego Mindstorms. However my research indicates that there was no significant difference between using one small robot or another in terms of the validity or results of the research. Substituting one small robot for another does not change the experience for the student or researcher to any significant degree. The only thing that changes is the way the student downloads the code to the robot and the functions that the robot can perform. None of the research articles that I found on the subject however, mentioned any concern that the robot’s task could affect the research results. My assumption then is that the results would be similar no matter what tasks the robots were asked to perform and thus it should not make any difference which robot the students use to perform these tasks.

Generating my Research

Once I had completed my reference list, my task was to dissect the information contained within them to formulate the answers to my research questions. In particular I wanted to answer the following questions:

1. What are the advantages of using robots to teach computer programming?
2. What are the disadvantages of using robots to teach computer programming?
3. Do students learn computer programming better, the same, or worse when using robots?
4. What are the challenges associated with using robots to teach programming?
5. What are the preferred methods of teaching computer programming using robots?
6. Does the use of robots make computer programming courses more popular?
7. Do robots attract a different type of student to computer programming courses?
8. Do robots help reduce the attrition rate of computer science programs?

9. Do students get higher or lower marks when learning computer programming using robots?
10. Do robots have a future in computer science programs?

To make the process easier, I listed the title of each research article on a separate sheet of paper along with the 10 questions above. Then for each article I noted the answer to each of the questions. Any questions that were not sufficiently addressed in the article received an answer of N/A meaning that the answer was not available. Following this I listed each of the 10 questions on a separate sheet of paper and under each question I listed all of the answers that I had recorded for that question. This allowed me to generate a very good summary of the results for each question and to compare the answers from the different references. It is these results that allowed me to form the basis of my research report.

CHAPTER 4

RESULTS

Discussion

The results of the research on this topic are mixed and therefore it is very difficult to draw a single conclusion. While many of the studies show an increase in student interest and student participation in programming courses using robots, others have found no significant difference. Furthermore, very few of the studies were able to show any direct correlation between the use of robots in programming courses and an increase of learning of the course material. As a result, in order to better present the research results for this topic, I have decided to divide the related research papers into 2 distinct groups and summarize each group in its own sub-section. First, I will present the papers that I conclude have negative results. That is to say, those papers who conclude that the use of robots in the classroom have a detrimental impact on the learning and/or interest of the students. Secondly, I will present those papers that show a positive correlation between the use of robots to teach computer programming and the learning and/or interest of the students. In this case, I have concluded that a research paper is positive if either the student's learning or interest is improved while the other criteria is either unchanged or improved.

Discussion of the Negative Results

Barry S. Fagin and Laurence Merkle, "Quantitative analysis of the effects of robots on introductory Computer Science education", J. Educ. Resour. Comput. 2, 4, Article 2 (December 2002), 2002

This study compared the results of 938 computer programming students over a one year period. Of these 183 were taught using robots while the rest were taught using traditionally methods. Both groups used Ada as their teaching language, and both were given the same mid-term and final exam. At the end of the year the results of these two exams as well as the students' rank in their following computer science course were compared to evaluate the effectiveness of using robots as a teaching method. In addition to this the researchers used surveys and focus group discussions to compile subjective student data.

The conclusions of this study were negative. The author's found that the test scores were lower in the robotics sections than in the non-robotics ones. Figure 1 below summarizes the results of the students for the two terms of the research study. To calculate this information, the authors combined the results for all the students in each section into one total mark as shown here.

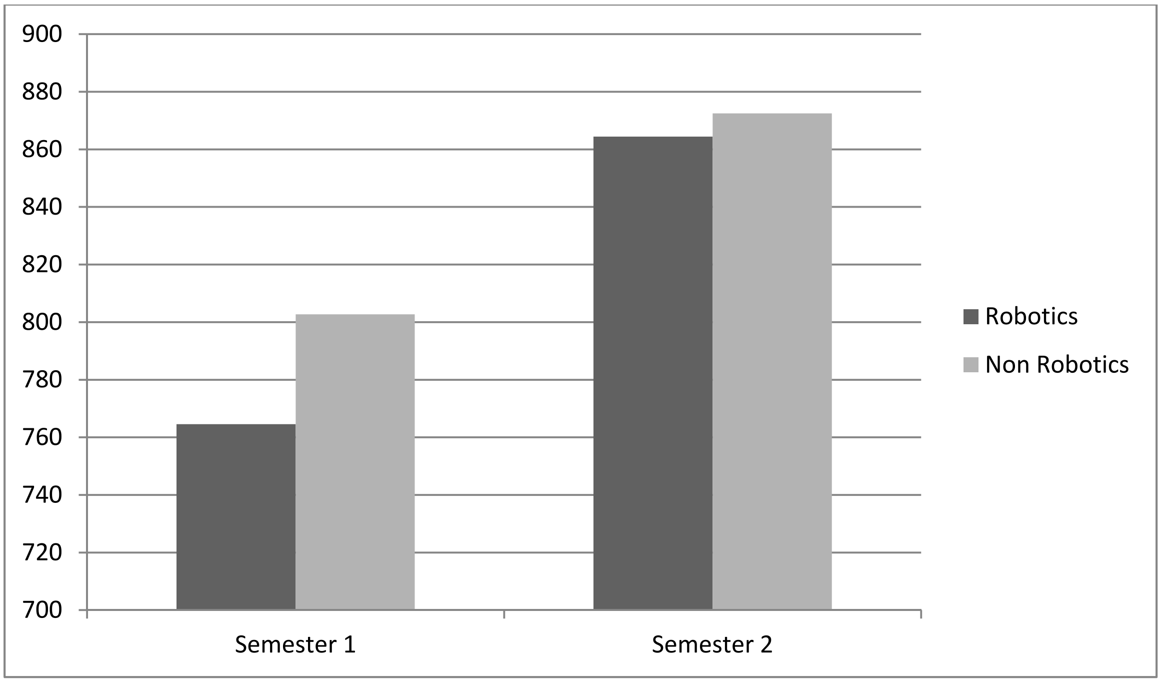

Figure 1. Total course marks for the Robotics versus Non Robotics sections

The subjective data compiled from the students was also negative regarding the use of robots. When asked about the course relevance, the amount learned and the instructor effectiveness of the course, the students in the robotics courses responded with slightly lower marks than those in the non-robotics courses. Figure 2 below shows these results.

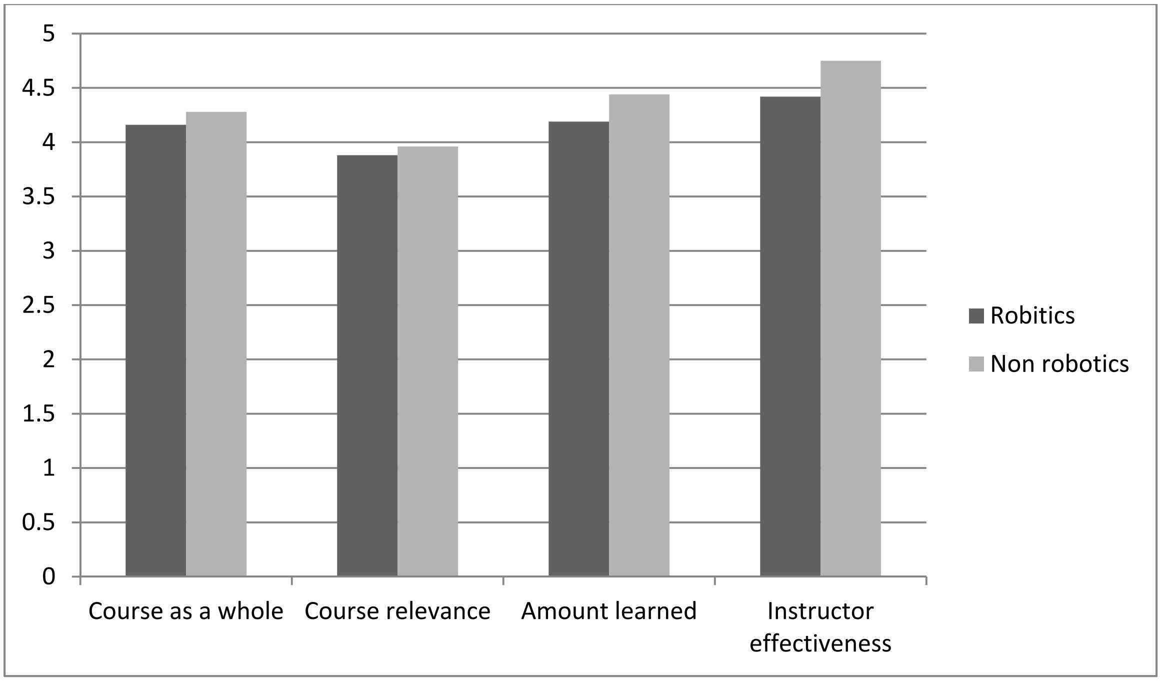

Figure 2. Subjective results of the Robotics versus the Non Robotics courses

Daniel C. Cliburn, "Experiences with the LEGO Mindstorms through the undergraduate computer science curriculum", The 36th ASEE/IEEE Frontiers in Education conference, San Diego, California, October 2006.

The author of this paper has incorporated robots into five different computer science courses at Hanover College in Indiana with mixed results. His most successful implementation was in an introductory computer science course for non-majors. During this course he presented the LEGO Mindstorms robot using the visual programming language provided by LEGO. While this language is deemed too simplistic for programming courses, it was a perfect fit for this type of course. As a result the reaction of the students was very positive. The author observed that the students were motivated and excited to use the robots and most were very successful at programming them using the visual programming language. He concludes that the LEGO Mindstorms robot is a good tool to use for the programming part of an introductory computer science course.

For the programming courses in the Computer Science program however, the author observed different results. He adopted the robots for four different courses at this level: Programming I, Programming II, a survey or Programming Languages, and a general programming language practicum. In all four cases his experience was mostly negative. In the Programming I course, the students had several technical difficulties getting the robots to respond appropriately. This wasted time and frustrated the students who remarked that even the simplest of tasks often took a long time to program. Also, due to the number of robots in the classroom the students had to work in teams. This meant that the teacher was not able to give individual assignments, which is often desired for a first programming course.

The experience in the Programming II course was for the most part unsuccessful. Students were assigned individual programming projects but had to share the robots due to a limited supply. As a result the students found it very difficult to find sufficient time to test and debug their programs. As a result, by the end of the term none of the students were enjoying working with the robots. The students learning also suffered due to the limited time they had to work with the robots.

In the survey or programming languages course, the robots were used in only one of the course assignments. For this assignment the students were given a challenge to solve using the robots. However, most found the challenge too time consuming and difficult and did not leave sufficient time to find a suitable solution. They also commented that building the robots was a pain and a waste of valuable time that could have been used programming instead.

In the general programming practicum course the author divided the class into groups of 3 or 4 students that were to use the robots to program a series of challenges. As the course was a practicum, there was no theory part to the course and the students were to work mostly on their own, meeting the teacher only to ask specific questions. Despite the informal approach to the course, the initial reaction of the students was very positive. Most were excited about the opportunity to use the robots and learn more about working in a team environment. However, once the students started the course their attitudes became very negative. Most found that the course took far too much of their time for the credit received. As a result most did not enjoy the course or their experience with the robots.

Therefore, the conclusion drawn by the author and teacher of these 5 courses is that the use of robots is not suitable for all courses. In fact, the only real successful experience he had was using the visual programming language supplied with the robots to teach an introductory computer science course to non-majors. While initially all the students find working with the robots interesting and fun, this quickly wears off and frustration sets in when the projects become long and complex or when the students start to have technical difficulties with the robots themselves.

Maya Sartatzemi, Vassilios Dagdilelis and Katerina Kagani, "Teaching Introductory Programming Concepts with LEGO MindStorms in Greek High Schools: A Two-Year Experience, Service Robot Applications", Yoshihiko Takahashi (Ed.), ISBN: 978-953-7619-00-8, InTech, 2008

This research paper presents the findings of studies conducted over a two year period in which 2 different classes in a Greek high school were taught computer programming using robots. Both classes used LEGO Mindstorms robots and worked in pairs. To program the robots, the more junior class used the ROBOLAB language that accompanies the robot kit, while the more senior class used NQC. To evaluate the students a series of lessons were given to the students to teach them the required material. Each lesson was followed by an exercise to complete on that lesson. These exercises formed the basis of the evaluation of the students.

The results of the evaluation were mixed or slightly negative. While the students seemed to enjoy using the robots and were enthusiastic about the course, many had trouble completing the work. In certain lessons less than 20% of the students were able to get the correct answer. In others the percentage rose to 60 or 70%. Interestingly, the students who used the Robolang language, which is supposed to be easier, had more trouble with the lessons than those using NQC. However, these were also the youngest students so that may have been one of the reasons for this.

Also, since there was no control group to compare against it is difficult to determine if the problems the students had were due to difficulties using the robots or difficulties with computer programming in general. Furthermore, the authors pointed out that since the students were all in high school and participation in the course was mandatory, there may have been a less motivation for some of them than there is for college students who take the courses by choice.

Myles McNally, Michael Goldweber, Barry Fagin, and Frank Klassner, "Do LEGO Mindstorms robots have a future in CS education? ", SIGCSE Bull. 38, 1 (March 2006), 61-62, 2006

In this paper three of the authors face-off in a discussion over the merits of using robots to teach introductory computer programming courses. All have had experience in the use of robots in their classrooms and therefore have a good basis for their opinions. Of the three, only Frank Klassner remains a believer in the use of robots in the classroom. However, even he warns that computer departments that are considering moving to the use of robots, must do so very carefully and give careful consideration to how they are going to introduce them. The other two authors are against the use of robots in the classroom. Michael Goldweber points out that the robots are both expensive and imprecise and thus not an ideal medium for teaching computer programming. He also believes they are too limited to do an adequate job of teaching object oriented programming, which is becoming the standard programming methodology today. Barry Fagin on the other hand believes that the use of robots is less efficient than using a simple compile, run, test method of programming due to the large turnaround time required to run and test programs using the robot and due to the limited time that students often get to spend with the robots. He also disputes any evidence that robots attract students to computer programming or that it makes them better programmers, referring to a research study on the subject that supports this argument.

William Isaac McWhorter, "The Effectiveness of Using Lego Mindstorms Robotics Activities to Influence Self-regulated Learning in a University Introductory Computer Programming Course", Dissertation Prepared for the Degree of DOCTOR OF PHILOSOPHY, University of North Texas, May 2008

This research was done as a requirement for the author’s Doctorial degree at the University of North Texas in 2008 and is comprised of data taken from students in the Introduction to C++ course given during the 2005 – 2006 school year. The data for the control group was taken from two sections of the course in the fall session totaling 40 students. The data from the robotics group was taken from two sections in the spring session and totaled 43 students. The robotics section used the LEGO Mindstorms robot and the NQC language with the robotics part of the course being introduced following exam 2 of the session.

As part of this study the author tested 20 different hypotheses relating to the students' motivation, learning methods, and mastery of the course material. To test the students' motivation he used the Motivated Strategies for Learning Questionnaire which he administered immediately after exam 2 and at the end of the course for both groups. The information was tallied and compared using an Excel spreadsheet. To test the mastery of the course material, three major exams were given to the students during the session. The exams were identical for both the robotics and non-robotics groups, allowing the author to compare the corresponding results. Furthermore, since both groups followed the same course up to and including exam 2, the results from exams 1 and 2 were able to be used as baseline data to compare the results of the final exam for the two groups.

The author collected and compared extensive data on the two groups and tallied it using a variety of statistical methods. After compiling and analyzing all of the results it was found that there was no significant difference in student motivation, mastery of computer programming, or the learning methods of the students in either group at the end of the course. While this result is not completely negative in relation to the use of robots in computer programming courses, it does suggest a lack of benefit in using them.

In response to these results, the author administered a small questionnaire to five of the students from the robotics group to try to better grasp the results. A sample of some of these questions is given in Table 4 below.

Table 4 – Follow-up questions used by McWhorter

What was your favorite or most memorable part of your introductory programming class and why?
Do you think the LEGO activities helped motivate students to be interested in computer programming?
Do you think students tend to be less motivated in the spring semester compared to the fall?
Would you have enjoyed the LEGO Mindstorms activities more if you had been able to build the actual robots?
What were your feelings about the LEGO activities? Did it seem too much of a child's toy / activity? Was it fun or just something else to learn?
Did the lack of outside class availability affect the effectiveness of the LEGO activities?
Did the LEGO Mindstorms activities have any influence on your learning and study strategies in your introductory programming class?
Did the LEGO Mindstorms activities help in your understanding of computer programming concepts such as selection structures, loops, functions, and arrays?

Interestingly, the responses to these questions were more positive in relation to the use of robotics. All of the students questioned indicated that using the robots was the most memorable part of their programming course. They also believed that using the robots could help motivate students to be interested in computer programming. None of the students however believed that using robots had an influence on their study habits or programming approach.

The author also mentions a few factors that may have affected the results. The sample size of this experiment was small with only 40 students in the control group and 43 in the robotics group. This can skew the results one way or another. The author also mentions that since the students only had access to the robots during laboratory periods, they were at a disadvantage in relation to the control group since they could not work on assignments on their own time. Many researchers have suggested that this may have an effect on both the student's motivation and success in the course. Another problem mentioned was the length of time it takes to run and test a program using a robot versus the quick test, run, debug cycle time of the typical IDE compiler. It has been suggested that this difference gives an edge to the traditional method of programming instruction. Finally the author did voice a concern that defects or inaccuracies in the robots themselves may affect the ability of the students to properly test and debug their programs.

In response to these factors the author suggests that another study be done using a larger sample size to verify these results. He also mentions that the use of a simulator would allow students to work on their assignments outside class hours and allow them a quicker test, run, debug cycle which may remove one of the primary concerns of the robot lab.

Discussion of the Positive Results

Chuck Leska, "Introducing undergraduates to programming using robots in the general education curriculum", In Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '04). ACM, New York, NY, USA, 263-263, 2004

In this research paper, the author discusses the introduction of an undergraduate level programming course to non-majors. During the course the students were required to complete several labs and 3 small projects using programmable robots. All of the programming was done using the NQC (Not Quite C) language and most of the course work was done in pairs. Although no official measurements were done, the authors did notice several positive results. Among them they reported that the course was filled to capacity, the students were very enthusiastic throughout the course, and that their programming ability increased steadily.

David A. Gustafson, "Using robotics to teach software engineering," Frontiers in Education, Annual, pp. 551-553, 28th Annual Frontiers in Education - Vol 2 (FIE'98), 1998

Software engineering is a relatively advanced course given to undergraduate students towards the end of their studies. In this case, the students were offered a 2 semester course on software engineering using robots. The idea behind this was to give the students experience in a team environment and to give them practice at programming a device that had actual motion and real-time characteristics. In this case the robot chosen was the Nomad200 from Nomadic Technologies Inc. The robots used the Linux operating system onboard and all the programming was done in the C++ language.

During the course, the students were divided up into teams of 4 to 6 members and given different programming tasks to solve in their teams. Since this was an advanced course, the tasks were of the intermediate to advanced level. Although all of the results in this research were subjective, the authors report a very positive experience throughout these two courses. The authors report that the students stayed positive throughout the two sessions of the course and most learned at an expected pace. Also, two of the teams of students went on to compete in robotics competitions at the national level.

Debra T. Burhans, "A Robotics Introduction to Computer Science", Canisius College, Computer Science Department, 2001 Main Street WTC 207, Buffalo, NY 14208, 2006.

While the sample size of this research paper is rather small at only 11 students, the results of the research are favorable. The basis of this research was an 8 week course on robotics given to freshmen undergraduate students at George Washington University. In this case the students used the LEGO Mindstroms robot and an in-house developed algorithmic language called Robolang which they use to write their programs which are then compiled and downloaded into the robots.

The students were given a series of programming tasks using the robots and at the end of the course they were given questionnaires to complete to survey their impressions of the course. Example questions are:

- The robots helped me understand programming
- The class made me more interested in Computer Science

The results of the questionnaires show a number of positive results. Three quarters of the students reported enjoying the course and the course sparked an interest in robotics and AI in over 80% of the students. And although the course did not attract any new students to Computer Science it did seem to be a good introduction to the discipline to those already contemplating it.

Debra Burhans, R. Mark Meyer, Patricia VanVerth, David Puehn, Victoria Steck and John Paul Wiejaczka, "Introductory computer science with robots. ", Paper presented at the proceedings of the 21st National Conference on Artificial Intelligence, July 16-20, in Boston, Massachusetts, 2006

In 2005 the authors researched the introduction of LEGO Mindstorms robots into two different Computer Science classes at the Canisius College in Buffalo, NY. One of the courses contained primarily non-major students and used a more generic algorithmic language. The other was a first programming course for those students majoring in Computer Science and used the LeJOS language, which is a variant of Java specifically designed for the LEGO Mindstorms robots. Although the quantitative results were not available at the time the research paper was written, the subjective results of the research are very favorable. Of the students surveyed, over 70% reported very favorable on using the robots, indicating that they both enjoyed using the robots and believed that the use of robots helped them understand the programming concepts.

Eric Wang, "Teaching freshmen design, creativity and programming with LEGOs and Labview," Frontiers in Education, Annual, pp. F3G-11-15vol.3, Frontiers in Education Conference, 2001. 31st Annual, 2001

This research paper was based on a 3 year period in which the LEGO Mindstorms robot was used to teach mechanical engineering students programming at University of Nevada. The course was set up as a project based course whereby the students progress through a series of 10 small projects, each one more difficult than the last. The course was then culminated with a robot competition in which the students battle their robots against others at a local tournament.

The authors documented an extensive list of results of their study which are primarily positive in nature. Below is a table of the most important results.

Table 5 - Results from Eric Wang, 2001

The use of LEGOs was found appealing to the students while providing an excellent medium for teaching design, programming, and creativity.
The use of LEGOs in the freshmen program has proved to be an excellent recruiting tool. The enrollment has more than doubled in three years, despite the national trend of decreasing enrollment in engineering programs.
The use of ROBOLAB was found to significantly increase the student's ability to program the robots as compared to the other two languages used.

Jerry Schumacher, Don Welch, and David Raymond, "Teaching introductory programming, problem solving and information technology with robots at West Point," Frontiers in Education, Annual, pp. F1B-2-7vol.2, Frontiers in Education Conference, 2001. 31st Annual, 2001

In the early 2000's the instructors at the US Military in West Point introduced the LEGO Mindstorms robot into the introductory computer programming course at their base. This course is mandatory for all trainees and has an enrollment of over 500 students per session. Although no quantitative results were tabulated, the subjective results that were reported are very positive. According to the researcher, the students were both motivated and successful in their assignments and the robots provided invaluable insight into robotics and embedded design that the students will need throughout their military careers.

Deepak Kumar, Doug Blank, Tucker Balch, Keith O'Hara, Mark Guzdial and Stewart Tansley, "Engaging computing students with AI and robotics", AAAI Spring Symposium Series, presented at the Symposium on Using AI to Motivate Greater Participation in Computer Science, tech. report SS-08-08, AAAI Press, 2008

This research effort took the approach that each student should own their own robot so that they can personalize and adapt their robots to their own situation as well as have access to it both at school and at home. This removes one of the major difficulties that many of the teachers have had when using robots in the classroom, which is specifically the lack of access that students have to them. In many courses the students must share a robot which is owned by the school, and which must remain in the laboratory. This means that the students have very little access to the robot to test their programs. The suggestion has often been made that the students have more access to the robots to increase the student satisfaction rate and success rate in the courses.

In this case the Scribbler robot by Parallax Corp combined with an add-on board by IPRE was used for the courses. Python and the Myro API were used for the programming environment for the robots which according to the researchers, provided a powerful and easy to use interface to the robots.

The research covered a period of 6 course offerings and over 200 students at Georgia Tech and Bryn Mawr College in 2006 and 2007. Extensive feedback was collected from the students using surveys and interviews both during and after the courses and summarized into the following comments (Table 5).

Table 6 – Results from Deepak Kumar, Doug Blank, Tucker Balch, Keith O'Hara, Mark Guzdial and Stewart Tansley

The students successfully learn Computer Science concepts through robots
The robots made the learning experience more hands on and exciting
The students discovered the link between Computer Science and the real world
The students understood the need for patience and thought when solving robotics problems

Michael W. Lew, Thomas B. Horton and Mark S. Sherriff, "Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course," Software Engineering Education and Training, Conference on, pp. 121-128, 2010 23rd IEEE Conference on Software Engineering Education and Training, 2010

The consensus in much of academia is that teaching computer programming using robots is better suited to introductory courses, and that the robot languages lack the complexity needed for more advanced courses. However, the Department of Computer Science at the University of Virginia decided to counter this advice and design a course in advanced software engineering based upon the LEGO Mindstorms robot and the LeJOS language (Java). In this course the students were given programming assignments on 4 major software engineering topics followed by a survey to see how the students rated their experience.

The results of the survey were mostly positive with approximately 75% of the students responding that the robots and LeJOS language were useful in learning the concepts of the course. The students did point out certain problems with the robots, the most important being the Bluetooth communications module which was often flakey and easily overloaded. However, after tallying the results, the authors determined that the positives were sufficient to continue offering the course using the robots in the future, albeit with a few adjustments.

Michael Goldweber, Joe Bergin, Raymond Lister and Myles McNally, "A comparison of different approaches to the introductory programming course", In Proceedings of the 8th Austalian conference on Computing education - Volume 52 (ACE '06), Denise Tolhurst and Samuel Mann (Eds.), Vol. 52. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 11-13, 2006

This paper is a discussion by the four authors about the merits of teaching computer programming with robots. The arguments put forth by them are based solely on references to past research as they did not perform any new research themselves for this study. However, I do consider their opinions valuable since it is based on recent peer reviewed research in this area that is valuable and reliable. The basic consensus of the authors is that robots are a useful tool for teaching introductory computer science since it does create interest and is capable of teaching the basic programming constructs required at that level. However, beyond that the authors do not see them as being useful. In other words, they do not feel that the robots and the associated languages are complex enough to teach advanced computer programming constructs.

Pauline Mosley and Richard Kline, "Engaging Students: A Framework Using LEGO Robotics to Teach Problem Solving", Information Technology, Learning, and Performance Journal, Vol. 24, No. 1, 2006

In 2005, Pace University in New York came up with a novel approach to teach computer programming to their students using LEGO Mindstorms robots. Their approach was to offer a 3-credit course to their students to not only teach them to program the robots but also to become robot programming instructors for middle school students. At the end of the course the students then go out to various local middle schools and show the teachers, coaches and students about the LEGO robotics technology. This not only reinforces the students learning of the subject matter, it also makes the course much more meaningful to the students since they have a concrete goal at the end of it. The success of this method was reinforced by the surveys and questionnaires given to the students at the end of their experience. All of the students were very positive about their experience in the course and in the middle schools, with many saying that their time spent in the middle schools was the favorite part of the course. They also expressed satisfaction at the opportunity for them to apply what they had learned by teaching it to others.

Sebastian van Delden and Wei Zhong, "Effective integration of autonomous robots into an introductory computer science course: a case study", J. Comput. Small Coll. 23, 4 (April 2008), 10-19, 2008

This research paper describes how the LEGO Mindstorms robot using Java was successfully integrated into an introductory computer programming course at the University of South Carolina. The course was offered as a first semester freshman course for those with no previous programming experience. The robotics programming part of the course consumed approximately one-third of the course, with the rest of the course being taught using traditional methods. The result of the research shows that the LEGO robots can be successfully integrated into an introductory programming course provided that a certain set of issues are addressed. In particular the authors mention the following concerns:

- You must have the budget to purchase and maintain the robots
- The classroom must be fairly large to support many robots being used at once
- Spare robot parts and batteries should be kept on hand to be ready when needed
- Allow time to setup the lab and clean up at the end
- It may be a good idea to pre-build the robots so that the students do not have to spend valuable lab time doing this
- If the students do not have access to the robots outside of laboratory time then make the exercises accordingly. That is to say, keep the assignments small enough to finish during one lab setting or break them up into many sections that can be spread over two or more lab sessions

Soumela Atmatzidou, Iraklis Markelis and Stavros Demetriadis, "The use of LEGO Mindstorms in elementary and secondary education: game as a way of triggering learning", Workshop Proceedings of SIMPAR 2008, Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS, Venice(Italy) 2008 November,3-4, pp. 22-30, 2008

Robot challenges are often a great way to maintain the students' interest in computer programming courses that use robots. These challenges give the students something to focus on and work towards. In this paper, the authors summarize a course that uses this approach. The course was divided up into a series of 6 lessons followed by a robot challenge in which the students would compete against each other for robot supremacy. As is usually the case, the students were divided into teams at the start of the course and all work thereafter was done within the boundaries of the team. In this way, the students learn about teamwork and cooperation as well as programming.

Research data was collected through observation as well as student questionnaires, interviews, and surveys. Upon compilation of the results, the authors made the following observations:

- The work the students did with the robots contributed to their overall learning of the programming constructs. It was noted that many of the students understood some of the basic programming constructs more easily when using the robots than with traditional methods. This observation was supported by some of the students' comments.
- The students were more likely to relate the computer programming to real world examples when the used the robots.
- The robot challenge was an incentive that made the students work hard to outdo the opponents and surpass any difficulties they had with the material.
- The students related new programming commands to the functions they performed on the robots, thus reinforcing their learning of the material.

Ursula Wolz, "Teaching design and project management with LEGO RCX robots", SIGCSE Bull. 33, 1 (February 2001), 95-99, 2001

This research paper describes the implementation of a robotics unit in a CS1 course at the University of New Jersey. The unit spanned 2 weeks out of a 14 week course, and required that the students solve one task using a LEGO Mindstorms robot. The results of the study are based on instructor notes, a video of the student robot demonstrations and on student summative essays.

The 2 week unit was given towards the end of the semester and was divided up into 2 individual lessons and a student demonstration. During week 1 the students were given instruction on the robots and several small exercises which they were to do individually. During week 2 the students were divided into teams and were instructed to work on their task. The task was to have the robot negotiate from one room to another and stop at a precise location. At the end of week 2 the students demonstrated their robotics programs.

The results of the experiment indicate that it was a success. A majority of the students enjoyed using the robots and found them instructional. From the teacher’s perspective, it gave the students a much better appreciation for thinking through the problem before implementation which is something that tends to be missing at this level. Overall the experiment was considered a success.

Vassilios Dagdilelis, Maya Sartatzemi and Katerina Kagani, "Teaching (with) Robots in Secondary Schools: Some New and Not-So-New Pedagogical Problems", IEEE Advanced Learning Technologies Conference, 2005

In this research the authors present a series of pilot lessons to introduce the LEGO Mindstorms robot and the visual programming environment ROBOLAB to students in the Greek high school system. In total the students were taught 3 different lessons for a total of 6 hours of instruction. Each lesson was followed by exercises and a questionnaire to complete. The authors compiled their results based upon this information.

The first observation made by the authors was the significant motivational factor that the robots had on the students. The students enjoyed the lessons and working with the robots. Observation showed that they also were able to perform adequately on the exercises at the end of each lesson. The authors concluded that some of the more elementary programming constructs were more easily understood using the robots than with traditional methods, while other more advanced methods remain difficult. They also mentioned that the ROBOLAB language was a very good one for this level of student.

The experiment was not without its problems however. The authors mentioned the long time it takes to edit, compile, and download a program to the robot to be a bit of a irritation to the students. The robots also tend to be imprecise entities, especially as the battery wears down, giving slightly different results at different times. In addition to this, there is often a slight difference in the way individual robots work and thus this can affect the results for students if they are given different robots on different days. Overall however, the teachers and researchers were satisfied with the way the lessons went and planned to keep the robots in the curriculum.

Summary

Of the 19 research papers that I have retained as references for this paper, 14 conclude that the use of robotics in computer programming courses has an overall positive effect. This effect can be in terms of student learning, motivation, or interest. By far the more frequently mentioned benefits are in student interest and motivation. The overall consensus of the papers I have studied is that the use of robots both interests and motivates students in their programming courses. Far fewer have noticed a significant benefit in student learning.

One reason for this might be that student learning is much harder to measure than either interest or motivation. Measuring student learning requires that the researcher has both a control group and a test group with a large enough sample size to extract statistically significant results. It also requires that the research develop a mechanism to measure baseline data for the two groups so that the measured differences in the results can be statistically calculated. Of the research papers I have found, only 2 went to this complexity. Interestingly, of the two, one was extremely negative towards the use of robots and one was neutral, finding no significant difference between the two groups.

Of these, the largest and most elaborate study was that by Fagin and Merkle in 2002. Their sample size was over 900 students and included both a control group and a robotics group. The authors tested the students learning through a series of tests and queried the students using questionnaires afterwards. The results of this research showed that the robotics group scored lower in learning and had a slightly less favorable attitude towards the course than those in the control group.

While these are results are slightly worrisome in terms of the implementation of robots in programming courses, the authors did mention several remedies which may help improve the robotics scores if implemented. One of the most important ones would be the addition of a robotic simulator to speed up development and testing. The long time it takes to compile and download programs to the robot and the fact that students cannot run and test programs outside the lab are considered one of the biggest problems with implementing them in the courses right now. It is also mentioned that the quality of the teacher may have an impact on the test results. It was noted that the teachers who were more comfortable using the robots had more satisfied students who performed better in their robot assignments.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The purpose of this research paper was to investigate the use of educational robots to teach introductory computer programming. Specifically, I wanted to investigate the effect that the robots had on the following criteria:

1. Student learning
2. Student interest
3. Student attraction to computer programming (both number and type)
4. Student retention

Initially I had hoped that the research on the subject would be conclusive on all the criteria. That is to say that the robots would affect all criteria in a similar and consistent manner either for or against the use of robots in the classroom. However, it quickly became apparent that this is not the case. The research has proven that each criterion is affected differently by the use of robots. Furthermore, the research on this subject is not conclusive, either on the overall use of robots in the classroom, or on their effect on any one of my research criterion. Therefore, the best way for me to summarize the conclusions on this topic is to analyze each of the criteria individually, and follow this with an overall conclusion based on all criteria taken together. The reader can then draw their own conclusions as to whether or not it is worth it to incorporate robots in the classroom. Following my summaries I have included a small table to help the reader see how each paper rated the above four criteria.

Student Learning

Perhaps the most important criteria studied were the effect that the use of robots had on student learning. If the use of robots in computer programming courses increased the rate of student learning of the subject, this would be a strong incentive to introduce them. Unfortunately, the results of the research on this criterion were mostly negative. The two most thorough studies done were by Fagin and Merkle in [1] and by William Isaac McWhorter in [17]. Both showed that overall the students who used the robots in the labs learned less and scored worse on the tests that followed the instruction. Other research showed mixed results. Some papers did show a moderate increase in the learning of the students using robots [2][8][16]. However, these papers had very small sample sizes and a lack of concrete quantitative analysis which put their results into question. Therefore overall, even with the mixed results, my conclusion on this topic is negative. There are however, certain conditions which all authors agree could improve or even reverse the results in this category, which will be discussed later in the recommendations section.

Student Interest

This was by far the most positive of the 4 criteria that I studied. Virtually all of the researchers found that student interest was improved by using robots in the classroom. Overall the students were excited and interested in the robots and found their use a refreshing change from the traditional teaching methods. In fact this was the only criterion in which Fagin and Merkle in [1] or William Isaac McWhorter in [17] reported positive results. Despite being negative in almost every area of the research, both of these papers did report an increase in student interest and involvement when using the robots. Positive results were also reported in this area by the authors in [2][3][5][6][10] and [19]. Therefore, I conclude that the robots do indeed increase student interest and involvement in the course.

Student Attraction

The purpose of this criterion was to judge if the use of robots in the classroom attracted more students to computer programming courses or if they were able to attract a different type of student to the computer science programs. Historically computer science programs have attracted a low percentage of females compared to males. Also, some programs that are suffering from low enrolments are concerned that the computer programming courses are regarded as boring or isolationist. Thus the hope is that the excitement and teamwork involved in using robots in the courses would attract either more students to the computer programming courses, more females to the computer programming courses, or both [9]. There is also a hope that this increased interest in computer programming courses may carry over, resulting in increased enrolments to Computer Science programs.

While not all the research papers measured this criterion, of those that did the results were mixed. On one end of the spectrum, Chuck Leska in [2] reported that in his courses the use of robots was very popular and that the courses were filled to capacity. Eric Wang in [7] had similar results, reporting that enrolment in his courses more than doubled after the introduction of robots to the curriculum. On the other hand, the other papers that measured this criterion were not able to prove that there was any increase in the enrolment in either the computer programming courses or the Computer Science programs as a result of the introduction of robots. Therefore I conclude that the effect that robots have on this criterion is neutral due to the mixed results.

Student Retention

One of the suspicions of Computer Science programs that are experiencing higher than average drop-out rates is that the students are either quitting or failing as a result of computer programming courses that are boring, hard, and uninspiring [9]. Thus, they are looking towards robots to help spruce up these courses and help retain students that would otherwise fall by the wayside. Unfortunately, very few of the papers that I researched were able to draw any conclusions about this criterion. Either they did not attempt to evaluate it or the results were inconclusive. My conclusion based on the results of the previous criteria is that the use of robots may help retain some of the students, particularly those who are quitting due to lack of interest. The only criteria that showed a positive correlation with the use of robots was that of student interest as mentioned above. Thus this increased interest could carry over and help increase student retention in certain cases. However, since the robots are only a very small part of any Computer Science program, it is unlikely that the overall affect will be significant. Therefore I conclude that the effect of robots on student retention is minimal or nonexistent.

Below I have included a table to show the results for each of my research criteria. Note that the research papers are only recorded under the criteria that they evaluated.

Table 7 – Summary for each of my research criteria

	Student Learning	Student Interest	Student Attraction	Student Retention
Positive result	2, 3, 7, 8, 9, 11, 16, 18, 19	2, 3, 4, 5, 6, 7, 8, 9, 10, 11,14, 17, 18, 19	2, 7	7
Negative result	1, 10, 17		13	13

Overall Conclusion

Based on the results of the individual criteria mentioned above, I have to conclude that the use of robots in the classroom does not make a significant difference to student learning, student retention, or to attracting students to Computer Science as reported in [1][4][10][17]. Ironically, the students do seem to enjoy working with the robots and find the courses more interesting with the robots than without as shown by the authors in [2][3][5][6][9] and [19]. Just why this interest does not carry over into the other criteria is not clear. The authors of the papers I have studied have made several observations on this and have suggested the following possible reasons:

- The students enjoy the robots but are frustrated by complications programming them [4][9].
- The students only have access to the robots in the lab. This means that they have much less time to test and debug their assignments since they cannot work outside the class [1][4][17].
- Testing and debugging programs on the robots takes much more time than with modern development environments. This results in fewer run, test, and debug cycles which results in longer program development times [1][13].
- Due to funding restrictions, students often have to work in teams and share robots. This can result in frustration as they may have to wait their turn to use the robots. It may also result in some students relying too much on their teammates and not learning what they should. This can affect their results on the final examination [10][17].
- The languages that must be used to program the robots are not identical to the standard programming languages and thus there can be a small learning curve in using them. Also, there are some programming constructs that cannot be taught properly using the robots thus creating a gap in the students learning [4][13].

Recommendations

The results of my research into this subject make it very difficult to recommend the use of robots in computer programming courses. Certainly there are reasons why a teacher may want to use robots in their programming courses beyond the criteria I studied. The use of robots gives the students a new insight into the world of technology and gives them experience in an area that they otherwise would not have [8]. It also allows them to see a more concrete and realistic example of the results of their computer programming efforts [5][8]. Indeed, these reasons are all valid, and thus a teacher may decide to go ahead with robot programming despite the rather negative results provided by the research. To this end, the research papers that I studied do provide some very concrete suggestions as to the means that teachers can take to make sure that their experience with robots is most productive. I will present the most important of those recommendations below.

Prepare adequately

All the research indicates that it takes a lot of planning, money, and preparation to properly use robots to teach computer programming. It is a fundamental shift from the way traditional computer labs are taught and as a result much of the infrastructure to do it is not yet in place. Teachers who are considering the use of robots in the classroom have many decisions to make before doing so. Some of these decisions are summarized in the following list:

- How many robots to buy?
- Will the students work in teams or individually?
- What robot language will be used?
- What percentage of the course should be designated for the use of robots?
- How will lab time be divided so that all students get enough access to the robots?

In addition to this, the teacher must give themselves sufficient time to become familiar with the robots and language before teaching the course. The research shows that the students did better in classrooms where the teacher was more knowledgeable about the robot environment compared to those where the teacher was not. Many of the research papers also mentioned the need for adequate lab space since the robots require a lot of room to maneuver. They also warned that a certain amount of time is lost for both setting up and tearing down the lab, and thus the teacher must incorporate this lost time in their preparation plan. The teacher should also budget time for broken robots, communication problems between the robot and computer, and similar delays. In general, programming with robots is a lot more time intensive than using a simple IDE, and thus the course must be designed accordingly [9][15].

Start slowly

Many of the researchers started by using robots for only a single lesson or section of the course rather than try to build the entire course around them the first year. I recommend this method of integration as well. Moving from a course that is entirely non-robotics to one that is entirely based on robotics in one year is too risky and demanding. Too many things can go wrong putting at risk the overall success of the course. A good compromise is to incorporate part of the course using robots and teach the rest of the course using the traditional method. The authors that did this seemed to prefer putting the robotics towards the middle or end of the course rather than the beginning so that they would have time to teach some of the basic programming constructs before starting the robotics section [15][18]. I recommend this approach as well. Doing so will ensure that all the students have a grasp of the fundamentals or programming before starting the robotics section, which should allow a smoother transition. It should also minimize the possibility of the robotics section having a negative impact on the students' learning since they will have a mix of both types of programming rather than only the robotics method. It also allows the teachers to introduce certain programming constructs in the non-robotics section that perhaps are harder to teach using robotics.

Choose the language and environment carefully

If the teacher is using the LEGO Mindstorms robots, which generally seems to be the case in most of the research I found, then there are several choices of the language that can be used. Out of the box, the LEGO is supplied with its own visual language. The research shows that it is very easy to use and the students have great success and enjoyment when programming the robots using this method [4][7]. The problem is that the LEGO visual programming language it is not one of the common core languages taught in C/S programs or one that is used in mainstream industry. Therefore, it is better suited for a younger audience or non-major programming courses where the programming constructs are less important [4][13]. For those courses in the Computer Science or science fields, it is better to use one of the more standard languages that have been adapted to the LEGO robot. These languages usually require that a special version of firmware be flashed to the robot before downloading the programs to it, but once this is done, the robots can be programmed as with any other language. The advantage of this method is that the teacher can choose a language that more closely resembles the one that has been traditionally taught in the course or one that is required in the program. The disadvantage is that the language is not native to the robot so there may be problems or inconsistencies that show up while running the program. For this reason it is recommended that the teachers perform adequate tests themselves of any languages that they intend to use before actually introducing it in the class.

If possible, arrange for students to use the robots outside of class time

One of the big complaints that many of the teachers had was the lack of time that students had to experiment with the robots [1][4][10][17]. Due to budget restrictions, most schools can only afford a few robots. This generally works out to fewer than one per student. This means that students have to share the robots in the labs and may not get to use them as often as they want to. To compound the problem, the schools usually do not allow the robots to leave the labs. This means that the students cannot bring them home with them to work on their assignments. With the lack of a suitable emulator, this means that students have a very short timeframe in which to get their programs running and tested. Compare this to the traditional method of computer programming instruction whereby the students have the IDE on their own laptops and can work on their assignments at any time, and you can see the justification for the complaint.

The researchers have proposed a few possible solutions to this problem. Finding a suitable emulator for the robot language you are using will allow the students to write and test their programs without the need for a robot [17]. They then can download to the robot only when their program is working. This reduces the demand for the robots significantly. Some schools have asked the students to buy their own robots. This is more appropriate if the school is going to make significant use of the robots over several courses, otherwise it is not really worth it. If this solution is chosen, then students can have complete access to a robot both in the labs and at home, giving them much more time to work on their assignments. A final solution mentioned to this problem, is to allow the students to sign out robots or access the labs outside of class time [1]. This requires more control over the whereabouts of the robots and there is a risk that the robots get broken, lost, or forgotten. Therefore this is a solution that not many schools have adopted.

Areas of further research required

Despite the numerous research papers written on the subject of using robots to teach computer programming, I find the quality of the research lacking. Only a few of the papers that I found performed research on a large enough scale to be truly reliable. By this I mean an experiment with a large enough sample size, a control group and a research group, and a method to compare both the starting and ending results of the two groups. Most of the research that I found had no control group. Instead they taught only using the robots and the results were formulated based on that group's data only. Furthermore, much of the data was subjective based on students' comments and the teacher's observations. Another problem that was frequently mentioned was that the students did not have enough access to the robots. The students using the traditional programming method had unlimited access to their IDE and could work both at home and in the class. Meanwhile, the robot group was usually restricted to working only in the classroom, sometimes having to share a robot with others. This meant that the control group had an unfair advantage over the robot group thus skewing the results.

As a result I would like to see much more research done in this area with these problems addressed. My suggestion would be perform similar experiments using both a control group and a research group. Standardized tests should be given before, during, and after the robot teaching section so that the two groups can be compared. This experiment should be run over several sessions at several different locations so that we can build up a large sample of reliable data. Furthermore, during these experiments the students in the research group should be supplied with their own robot so that they will have equal opportunity to test and debug their programs as the students using the regular IDE. This will remove one of the major deficiencies that were mentioned in the research to date.

One other area of research that has not been adequately addressed in my opinion is that of the robot language. The robots currently are delivered with a visual language that is very easy to use. However, since this language is visual and not very complex, it is rarely used in Computer Science programming courses. As a result, many organizations have developed other firmware and languages that are compatible with the robots. Little or no research however has gone into studying which of these languages is the most successful at teaching the students the required concepts. Usually the schools simply choose a language that is based on the requirements of the computer program, without much regard to how well the language works with the robots.

However, given the correct research, the schools would be able to choose the language based on some concrete evidence of which language is the best for student learning and interest. This should improve the results of the research groups in the future and improve the experiences of the students in robot programming courses.

Final Summary

Teaching introductory computer programming has always been a difficult task. Students usually arrive with little or no computer programming experience and little comprehension of what it entails. They often have unrealistic expectations about how difficult it is and frequently expect to create large exciting programs with very little effort. When they realize the length of time it takes to create and debug even the simplest programs, they often get disillusioned and lose interest. For this reason, teachers are always searching for novel techniques and tools that can both teach students how to program and at the same time retain their interest.

At first glance, educational robots seem to be just such a tool. They are affordable, interesting, and can be programmed in many different languages. Students get immediate feedback from the robot and they get to experience a real world example of what can be done with a computer program. Compare this to a simple text based program on the computer screen and you can immediately see the robot's appeal. Unfortunately, in the real world there are many difficulties that arise that make robots a less than ideal tool.

The research shows that the use of robots does not do a better job at teaching students how to program computers than the traditional methods. And even though they do seem to interest the students, they do little to attract more people to Computer Science programs. However, the researchers have raised several valid concerns which, if addressed, can make robots a more attractive alternative. First of all, the students must have sufficient time to test and debug their programs. To do this they will need access to either the robot or an emulator outside of class time in order to finish their assignments. During most of the research projects, the students were forced to share the robots and only had access to them during lab periods. In addition to this, at the time of the experiments there were no emulators available that were reliable enough to substitute for the robots themselves. This created a very limiting factor in the research.

The students also mentioned having difficulties with the robots themselves, such as defects, communications problems, and weak batteries. This created frustration for them and could have been addressed by simply keeping several spare robots and batteries on hand to replace those not working. It is also advisable that the teachers be very well versed in programming and repairing the robots to limit any such delays in the laboratory.

Finally the teachers should use a mix of robotics and traditional teaching in their courses. While robots are very educational and interesting tools, there are some programming techniques and constructs that are more easily taught with traditional methods. Teachers have to be aware of what works well with the robots and what does not and divide their courses accordingly.

Therefore, I believe that educational robots can be a useful tool to teach introductory programming but only if they are used under the right conditions. The research shows that students enjoy using the robots and are eager to take courses that include them in the curriculum. Teachers have to make sure that they employ the right techniques to ensure that their courses succeed with a minimum of delay or problem.

REFERENCES

[1] Barry S. Fagin and Laurence Merkle, "Quantitative analysis of the effects of robots on introductory Computer Science education", J. Educ. Resour. Comput. 2, 4, Article 2 (December 2002), 2002

[2] Chuck Leska, "Introducing undergraduates to programming using robots in the general education curriculum", In Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education (ITiCSE '04). ACM, New York, NY, USA, 263-263, 2004

[3] David A. Gustafson, "Using robotics to teach software engineering," Frontiers in Education, Annual, pp. 551-553, 28th Annual Frontiers in Education - Vol 2 (FIE'98), 1998.

[4] Daniel C. Cliburn, "Experiences with the LEGO Mindstorms through the undergraduate computer science curriculum", The 36th ASEE/IEEE Frontiers in Education conference, San Diego, California, October 2006.

[5] Debra T. Burhans, "A Robotics Introduction to Computer Science", Canisius College, Computer Science Department, 2001 Main Street WTC 207, Buffalo, NY 14208, 2006.

[6] Debra Burhans, R. Mark Meyer, Patricia VanVerth, David Puehn, Victoria Steck and John Paul Wiejaczka, "Introductory computer science with robots. ", Paper presented at the proceedings of the 21st National Conference on Artificial Intelligence, July 16-20, in Boston, Massachusetts, 2006.

[7] Eric Wang, "Teaching freshmen design, creativity and programming with LEGOs and Labview," Frontiers in Education, Annual, pp. F3G-11-15vol.3, Frontiers in Education Conference, 2001. 31st Annual, 2001

[8] Jerry Schumacher, Don Welch, David Raymond, "Teaching introductory programming, problem solving and information technology with robots at West Point," Frontiers in Education, Annual, pp. F1B-2-7vol.2, Frontiers in Education Conference, 2001. 31st Annual, 2001

[9] Deepak Kumar, Doug Blank, Tucker Balch, Keith O'Hara, Mark Guzdial and Stewart Tansley, "Engaging computing students with AI and robotics", AAAI Spring Symposium Series, presented at the Symposium on Using AI to Motivate Greater Participation in Computer Science, tech. report SS-08-08, AAAI Press, 2008.

[10] Maya Sartatzemi, Vassilios Dagdilelis and Katerina Kagani, "Teaching Introductory Programming Concepts with LEGO MindStorms in Greek High Schools: A Two-Year Experience, Service Robot Applications", Yoshihiko Takahashi (Ed.), ISBN: 978-953-7619-00-8, InTech, 2008

[11] Michael W. Lew, Thomas B. Horton, Mark S. Sherriff, "Using LEGO MINDSTORMS NXT and LEJOS in an Advanced Software Engineering Course," Software Engineering Education and Training, Conference on, pp. 121-128, 2010 23rd IEEE Conference on Software Engineering Education and Training, 2010

[12] Michael Goldweber, Joe Bergin, Raymond Lister, and Myles McNally, "A comparison of different approaches to the introductory programming course", In Proceedings of the 8th Austalian conference on Computing education - Volume 52 (ACE '06), Denise Tolhurst and Samuel Mann (Eds.), Vol. 52. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 11-13, 2006

[13] Myles McNally, Michael Goldweber, Barry Fagin, and Frank Klassner, "Do LEGO mindstorms robots have a future in CS education? ", SIGCSE Bull. 38, 1 (March 2006), 61-62, 2006

[14] Pauline Mosley, Richard Kline, "Engaging Students: A Framework Using LEGO Robotics to Teach Problem Solving", Information Technology, Learning, and Performance Journal, Vol. 24, No. 1, 2006.

[15] Sebastian van Delden and Wei Zhong, "Effective integration of autonomous robots into an introductory computer science course: a case study", J. Comput. Small Coll. 23, 4 (April 2008), 10-19, 2008

[16] Soumela Atmatzidou, Iraklis Markelis, Stavros Demetriadis, "The use of LEGO Mindstorms in elementary and secondary education: game as a way of triggering learning", Workshop Proceedings of SIMPAR 2008, Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS, Venice(Italy) 2008 November,3-4, pp. 22-30, 2008

[17] William Isaac McWhorter, " The Effectiveness of Using Lego Mindstorms Robotics Activities to Influence Self-regulated Learning in a University Introductory Computer Programming Course", Dissertation Prepared for the Degree of DOCTOR OF PHILOSOPHY, University of North Texas, May 2008

[18] Ursula Wolz, "Teaching design and project management with LEGO RCX robots", SIGCSE Bull. 33, 1 (February 2001), 95-99, 2001.

[19] Vassilios Dagdilelis, Maya Sartatzemi, Katerina Kagani, "Teaching (with) Robots in Secondary Schools: Some New and Not-So-New Pedagogical Problems", IEEE Advanced Learning Technologies Conference, 2005