DEPARTMENT OF COMPUTER SCIENCE
SERIES OF PUBLICATIONS C
REPORT C-2008-206

Why Using Robots to Teach Computer Science can be Successful Theoretical Reflection to Andragogy and Minimalism

Marja-Ilona Koski

Jaakko Kurhila

Tomi A. Pasanen

UNIVERSITY OF HELSINKI
FINLAND

Why Using Robots to Teach Computer Science can be Successful
Theoretical Reflection to Andragogy and Minimalism

Marja-Ilona Koski[1], Jaakko Kurhila[1] and Tomi A. Pasanen[2]

[1] Department of Computer Science
P.O. Box 68, Fin-00014 University of Helsinki, Finland
[2] Gamics Laboratory
Department of Computer Science
P.O. Box 68, Fin-00014 University of Helsinki, Finland

Technical report, Series of Publication C, Report C-2008-206
Helsinki, October 2008, 16 pages

ABSTRACT

To help students understand subjects such as theoretical aspects of computation, algorithmic reasoning and intelligence of machines, a number of publications report experiments to teach these topics with the help of Lego Mindstorms robots. In the publications, the researchers report how they have created various ways to approach the issues either in Computer Science or in Artificial Intelligence. The reported results of the experiments are based on the learning outcomes, the feedback from the students, and the perceived informal observations (i.e. “feelings”) of the instructors.

But can anyone else benefit from the reportedly positive outcomes of the experiments? To give an answer to that question, this paper analyses the reported results through two support theories. The two theories chosen for this, andragogy and minimalism, are concerned with adult learning and how teaching adults should be approached. When reflecting the results of the four teaching experiments to the suggestions drawn from the theories, a more comprehensive answer to why the experiments have been successful can be given.

The four teaching experiments analysed here were in many ways similar to each other. A connection to the chosen support theories was straightforward to make. Besides describing the artefacts of teaching with the robots, a deeper discussion on this teaching approach is provided. For an instructor, all these observations offer more concrete evidence about beneficial factors of teaching with robots.

Categories and Subject Descriptors
K.3.1 [**Computers and Education**]: Computer Uses in Education – collaborative learning;
K.3.2 [**Computers and Education**]: Computer and Information Science Education – c*omputer science education, self-assessment*

Keywords
Robots, teaching experiment, adult learning theories, adult education, evaluation.

1. INTRODUCTION

It has been noted that hands-on experience is largely missing from the Computer Science classes while such sessions are common in laboratory sciences (Stein 1998). Some university instructors have chosen a different method to teach basic concepts from their field of specialization. They have decided to give Computer Science or Artificial Intelligence (AI) courses with the help of robots. These researchers have stated that this method helps to make the learning more captivating and interesting (Kumar and Meeden 1998, Imberman 2004 and Kumar 2004). However, none of these teaching experiments reflect their results on anything else than the experience itself and the responses given by the students. When reading these publications, an instructor willing to use such a method in his/her classroom might confront questions whether it is possible to produce the same positive learning outcomes again and does it really engage students to the task in the same way as described in the publications.

The aim of this paper is to clarify *why* the methods used in the teaching experiments can turn out to be effective by reflecting the earlier experiments to theoretical frameworks in two well-known learning theories. The researchers of the chosen teaching experiments described positive and/or negative outcomes, but they did not adequately treat the question whether there is a transfer of success if someone else decides to use the Lego Mindstorms robots in a similar fashion.

Four well-reported teaching experiments with Lego Mindstorms robots were chosen for our reflection (descriptions of teaching experiments in Subsections 3.1-3.4). In addition to the chosen experiments, there are of course several other publications that report the development in the field of the teaching with robots by designing new exercises for the class or innovative ways to use Mindstorms robots (Flowers and Gossett 2002; Imberman and Klibaner 2005; Imberman 2005; Jipping et al. 2007; Klassner and Continanza 2007). There are also other evaluation reports on the use of robots, but the focus of these is more on the robots themselves than on the learning process (Challinger 2005; Gross and Power 2005).

2. SUPPORT THEORIES

The theoretical frameworks used in our paper are andragogy and minimalism. The theories are well-suited for university-level education, as the learners are adult and the efficiency of education tends to be of high priority in formal educational settings. In the following two subsections these theories are explained and described on such a level that further evaluations of the chosen teaching experiments can be understood.

2.1 Andragogy

The theory of andragogy is based on a set of assumptions that describe how adults learn. This idea originates from the fact that adults learn differently than children, and the pedagogical methods used to teach children will not work among adults. The first of assumptions of the theory is *self-concept of the learner*. It describes how it is the job of an adult educator to move the self-concept of the learner from being a dependent personality towards the self-directed learner (Knowles 1980). The theory of andragogy directs the instructor to recognize the duty to encourage the adult learner to move away from his/her old learning habits, and to become a self-directed, independent learner who takes responsibility of his/her own learning activities.

The second assumption is *prior experience of the learner*. Here, the learner’s experience of life is taken into account when new concepts are taught. Adults have a reservoir of experience which is a rich resource in the learning process to themselves and for others (Knowles 1980). Knowles (1980) states that people attach more meaning to the studied matter if they gain it from the experience than if they acquire it passively. Thus, an ideal situation for an adult to learn is with the laboratory experiments, discussions, problem-solving cases, simulation exercises and field experiences (Knowles 1980).

Thirdly comes *readiness to learn*. The idea of the third assumption is to evoke the learner's need to know the matter being taught. People become ready to learn something when they realize that they need to know it in order to perform better with real-life tasks (Knowles 1980).

The fourth assumption is *orientation to learning*. In adult teaching, it is important to acknowledge the fact that adults want to apply the knowledge and skills they learned today into living more effectively tomorrow (Knowles 1980). Adults need to find out what kind of effect the newly learned skill will have in their everyday life. Because of this, adults tend to learn in a problem-centered or performance-centered way of thinking (Knowles 1980).

Learner's need to know is the fifth assumption. Adults want to know the reason why something is important to learn, and how they can benefit from it (Knowles, Holton, and Swanson 1998). The adult learner needs to value the lessons, and his/her expectations should be filled in the class room by including an explanation of the importance of the matter.

The sixth assumption, and the last one, is *motivation to learn*. It is important for an adult educator to realize that potential motivators of the adult learning process are internal, and they come from the learner's own experience of him/herself (Knowles, Holton, and Swanson 1998). This does not exclude the fact that adults also respond to external motivators. Such factors as self-esteem and quality of life are important in giving adults a reason to learn (Knowles, Holton, and Swanson 1998). Expressing the learner's own opinion of the prioritization of the topics covered in class can give a learner the needed boost of motivation to learn.

2.2 Minimalism

The theory of minimalism assumes that when people are engaged in a task they will start to reason creatively and improvise (Carroll 1998). To support this, only the metadata of the matter should be provided to the learner, so that the learner can make the assumptions and reason on his/her own. The second thought derives from the idea of creative reasoning and improvising. When people are creating something or going with their instincts, errors tend to occur. This kind of action path is supported by the theory of minimalism. If and when errors occur, they are recognized and diagnosed with the help of the instructor and the use of a text book or a manual (Carroll 1998).

The first of the principles advises to *choose an action-oriented approach*. Opposite to the typical manuals where the first task is assigned on say page, 15, a manual designed as minimalism suggests introduces the first task on page one or two (Dubinsky 1999).

The second principle is *anchoring the tool in the task domain*. If the documents are done according to the suggestions given by minimalism, the text is presented in a short and simple way, and it must be easy to understand (Dubinsky 1999). The chapters are designed for an average user without long introductions and technical descriptions.

Support of error recognition and recovery is third on the list of principles. The idea behind this principle is the assumption that beginners make mistakes. The intention is to make features into the documentation that help the learner to identify and recover from his/her mistakes (Dubinsky 1999). This way, learning is a process where the learner is directing his/her own learning. The whole process should be seen as discovery learning where the learner is active and highly motivated by the tasks (Carroll and van der Meij 1996).

Support reading to do, study and locate is the last of the principles. The goal in this is to keep every section of the text self-contained (Dubinsky 1999). With the independent parts, the learner is not confused by the cross-references to earlier or later chapters. The main idea, when designing manuals, should be to give a learner the possibility of sequential processing, but also to enable random access approaches as well (Carroll and van der Meij 1996).

3. EARLIER TEACHING EXPERIMENTS

The goal of the following four subsections is to give the reader an idea about the evaluated experiments. Before looking in more detail into the teaching experiments, a short description concerning the tools used is needed. All of the chosen teaching experiments used the same Lego Mindstorms robots (Lego 2007) to teach the desired notions of the Computer Science.

The Lego Mindstorms kit includes Lego bricks to build the robots, and one programmable Lego brick called RCX. The first three experiments used the RCX brick to control the robot. Contrary to the others, the fourth teaching experiment used the Handy Board (Handy Board 2003) as a central unit of the robot. The Handy Board is a microcontroller system for building small, mobile robots mainly for educational or hobbyist purposes.

In the case of RCX, all programs are downloaded to it from a computer through an infrared transmitter which is connected to the computer's USB port. The RCX brick has three outputs A, B and C for the motors and for the lamps, and three inputs 1, 2 and 3 for the sensors. This Lego Mindstorms kit includes a Windows-based visual programming environment, but in all of the experiments it was stated to be too limited in its expressive power for the tasks that needed to be accomplished. The Handy Board works almost in the same way; it only has a few improvements compared to the Lego product.

3.1 Teaching a Computer Science course in the US Air Force Academy

The teaching experiment described in this section is based on the following three publications (Fagin 2000, Fagin 2003, and Fagin, Merkle, and Eggers 2001).

The goal of the Computer Science course is to provide the learners with a strong core competence for future Air Force officers. One of the desired skills is programming. The authors argue that the use of the Ada/Mindstorms and the robots offer a new and interesting way to teach basic computing and controlling concepts.

The course was about introducing basic computing ideas, such as sequential control flow, selection, iteration, input/output, arrays, graphics, procedures and file processing. Six of these concepts are introduced in the publications.

Sequential control flow was taught with an exercise where the students were given a robot with two wheels connected to outputs A and C and the task was to write a program that makes the robot go forward for two seconds, then play a song, and after that go forward for one second, and stop.

To teach the use of variables, the method used was to show how the robot changes its behaviour according to the quantity in question. To demonstrate the meaning of the term in action, an exercise was made where the amount of time that the robot travels is changed by a numeric calculation written in a program code,

The benefit of the use of constants was demonstrated to the students by a problem in which the robot needs to turn right 90 degrees. An accurate amount of time required for an accurate 90-degree turn is represented as a constant.

All the programs consist of procedures. When writing a program for the robot, it can be seen that the smaller tasks, such as the one presented previously, should be written as a procedure. One problem can be reduced into smaller problems and this way reaching the goal step-by-step is easier than solving the problem at once.

To approach selection and Boolean expressions, authors have chosen a task where the Mindstorms robots react to their environment. Robots can receive inputs though previously stated input ports 1, 2 and 3 and this way the behaviour of the robot can be controlled and influenced.

In a case of teaching arrays the students were asked to capture a sequence of numbers which were given as input through the presses of touch sensor and bumper. Once this sequence was captured, it was the robot's job to "play it back". In other words, it means that the robot needs to examine each number and execute the part of the program where there is a predefined action to that number.

3.2 Use of robots in an Artificial Intelligence course at Ramapo College

The teaching experiment described in this section is based on the following two publications (Kumar 2001 and 2004).

The course was a traditional AI course in the sense that the students were taught representation and reasoning, focus on search, logic and expert systems. The first time the course was held this way was in the fall semester 2000. The results about teaching the course and the learning experiences have been monitored for three years. During that time the basic concepts and ideas of the course have stayed the same, only the minor adjustments stated before were done.

The first task in the AI course was a project on blind searches; depth-first and breadth-first search of a tree. At the beginning of the programming task the students could assume that the tree is a binary tree with tree levels. However, in the later stage they had to generalize their implementation to deeper trees with an arbitrary branching factor.

The second task was about heuristic searches; hill-climbing and best-first search. In the project, the robots were expected to find their way out of a maze. In addition, while searching for the route out, they were expected to build a search tree of the maze. The robots interacted with their environment through touch and light sensors.

In the third task, the robot had to be able to determine the characters printed on a grid. To be able to complete this project, the students had to use the idea of forward and backward chaining in a rule-based expert system. The robot had to be able to go through a grid of pixels and use both the forward (data-driven) and the backward (goal-driven) chaining to determine the character in question.

3.3 An elective AI course in Villanova University's Computer Science program

The teaching experiment described in this section is based on the following publication (Klassner 2002).

It should be noted that Cognitive Science minors and Computer Engineer students also participate in the course. These students have no programming experience or at the most, a one-semester course of introduction to Java. However, the Computer Science majors, who participate in this course, take it in their fourth year, and by then they have taken a course "Programming Languages".

The first project of the course was a one-week project where the students experienced that with the simple stimulus-response rules and a limited model of the environment, the robot could achieve effective behaviours. The first task was to program the robot to move randomly ignoring any stimulus coming from outside. When this behaviour was accomplished, the robots were timed on a short obstacle course with narrow passages.

The second task was that the robots needed to monitor their environment. The feedback from the outside world to the robot came through an infrared, a light or a touch sensor. The infrared or light sensors were used to determine whether the robot was too close to the wall. The light or touch sensors were in use to detect if the robot has wedged into a corner. After either of these changes, the robots were timed on the same course as in the first part of the project. The result from the first task was now compared to the results of the second task and this way the students could observe the improvements.

In the second project, the goal was to show students how sensitive each of the sensors were to various stimulus. The goal was also to demonstrate how some sensors can interfere or simulate other sensors' capabilities. The first part of this project required a team of students to work with all types of sensors (touch, light, infrared and rotation) to generate different kinds of inputs and this way to study the various responses that the sensors could generate. The second part of the project was to design in teams a simple robot that used only a touch, an infrared and/or a light sensor to duplicate the accuracy and sensitivity of the rotation sensors. With this, the purpose was to demonstrate the concept of the functional emulation.

The third project was a project of two weeks where the students acquainted themselves with an important issue in the navigation process. The goal was to help the students understand factors that could cause error in the robot's internal representation of where it thought it would be located in the world.

Projects four and five were about building a ball-playing robot to compete against the robots built by the other teams. In these projects the students combined the previously learned skills, but also encountered new problems and possible solutions. In the project four teams built only one robot that played the game against the other robots, but in project five, three robots per team entered the ball field.

The sixth and the last of the projects was a two-week project with the goal of showing the students that the knowledge representations that speed up the search-based problem solvers can produce such a solution presentation that cannot be easily translated into the control programs of the hardware. In the first part, the students had to solve an 8-Puzzle by developing a knowledge representation and a Lisp search program. The teams developed a set of four operations to conceptually move the robot. These four movements reduced the search compared to 32 operations that would be needed to move each of the numbered tiles. In this way the students could experience the reduced branch factor of the search tree, leading to a faster execution time for the game solver. In the second part, the students were asked to write a program that invoked functions in an ad hoc library, developed by the instructor, to send remote-control messages to the robot's arm mechanism and this way move pieces on the 8-Puzzle.

3.4 An elective AI course in College of Staten Island

The teaching experiment described in this section is based on the following two publications (Imberman 2003 and 2004).

The chosen method to control the robots on the course was the MIT Handy Board. The controller of the Handy Board contains 32K of battery-protected RAM, and it has four DC motor outputs, nine digital and eight analogue inputs. These inputs support diverse sets of sensors. Several compilers are available for the Handy Board, including the Interactive C. Using the Handy Board for the Lego-based robots enables creation of sophisticated behaviours. Because of this, the Interactive C was chosen as a programming language to the project.

The overall objective of the project was to design and build a robot that will use a neural network to successfully navigate a circular path. The project started with the instructions on how to build a gear box for the robot to work properly. Instructions were also provided to build the robot so that it would be suitable for the path-finding task. Even though building the robot is an important part of the whole project, the goal here as well as in the other teaching experiments was not to spend too much time on concrete construction of the robot.

The third part was to write a program with the Interactive C where the robot moves forward for half a minute, then turns right, and then again goes forward for half a minute and turns, this time to the left. The goal was to find a minimum motor power to make the robot move. For the training examples of the neural network, the turning power needed should be written down.

In the fourth task, the students first wrote a program that would display the readings from both the robot's photo sensors. Again the readings should be written down because they would be used later in the project. To continue the project, the students had to estimate the parameters needed for the left and right motor functions to control the real wheels.

After these four tasks, the students used the earlier modified Generation 5 code to program their robot with a neural network. Once the training was done, the generated neural network was tested on another robot with the same type of sensors. When the students had the weight values for the neural network equations, they incorporated them into the Interactive C neural network and tested their robot. It was important at this phase to make the robot move slowly enough that it would have enough time to take the readings from the road and act upon them.

4. ANALYSIS

4.1 From mistake to understanding

In the results from the chosen teaching experiments, researchers report a better learning outcome in certain topics which are usually considered to be difficult to the students. During the AI course, if the students were asked to make a conceptual difference between training a neural net and a finished product, a trained neural network, they usually had problems in answering the question (Imberman 2003). Due to the architecture of the robot, there was not enough memory to train a neural net on the robot. Students soon realized this, and they started to do the first part on a computer and then transfered the finished product to the robot (Imberman 2003).

Similar results were observed when the students were learning the concept of procedures. Students added a new procedure to the code, but forgot to call it in the later stage of the code (Fagin, Merkle, and Eggers 2001). As a result, the robot did not present its newly added behaviour. Because the robot visualizes the commands in the code the students could observe the incompleteness of the code immediately and that helped them locate the problem.

Learning, as it is described in these experiments, can look like learning by trial-and-error. However, it can also be seen as a learning process where the learner is directing his/her own learning. It was stated that students thought they used the procedure call correctly, and only after testing found out what was missing (Fagin, Merkle, and Eggers 2001). Researchers also stated that when the students looked through the sequential control flow of their program code, they immediately saw that they did not program the robot to do its new behaviour (Fagin, Merkle, and Eggers 2001). So when the robot's actions were not the wanted ones, students needed to reconsider the solution. According to the theory of minimalism, beginners make mistakes, and the use material should support error recognition and recovery from mistakes (Carroll and van der Meij 1996). When the action path is as it is described in the experiment, the robot supported the recognition of the error by showing the missing part in the program code with its behaviour.

4.2 Designing a course

Traditional courses create a more comfortable learning environment because the instructor has years of experience in what to teach and how to teach it. How well the instructor handles the studied topic and the study material is reflected by the students' experience of the course (Fagin and Merkle 2002). However, the little amount of experience can also be turned into the strength of the course, and as a possibility for the learner to take charge of his/her own learning activities.

Students often feel that education is something done to them instead of experiencing it as something that they are actively doing for themselves (Beer, Chiel, and Drushel 1999). With the change in the attitudes of the students, the encountered situation of uncertainty could be seen as an instructor’s way of supporting the students to become independent and self-directed learners. The theory of andragogy states that the problem with the adult learner is a learning model from previous schooling (Knowles 1980). A more familiar approach to students is to get the answer of what to do than to figure it out by themselves. Also, based on the same theory, the adult learner has a need for autonomy (Knowles 1980). Therefore, by providing guidance to the learner, the instructor can be more beneficial in the learning process than by being a person telling students exactly what to do.

The theory of minimalism also suggests that the manuals used for studying would not be totally complete (Carroll and van der Meij 1996). This does not mean that the students are left without any guidance or help, but to encourage them to use their abilities and knowledge to “fill in the gaps”. The material designed to help the students solve their problems should give enough support but also leave space for their own interpretations and ideas (Carroll and van der Meij 1996).

4.3 Workload of the course

With the Mindstorms robots the workload of the course is bigger than course credits may predict (Klassner 2002). Because many universities are not willing to raise the number of credits gained from the course, instructors had to make a decision that the course will have an open lab work (Kumar 2004) or allow the students to take material out of the lab to work on it at home (Imberman 2004). Contrary to the author's beliefs, the students did not consider this a drawback of the course. The students reported that they spent a vast amount of time on constructing the robot and testing their code, but by the end of the course, it all appeared to them as a good investment (Kumar 2001).

The reason why students considered the workload of the course rewarding could be that the students were ready to learn the subject that was taught. The theory of andragogy describes how to evoke the learner’s readiness to learn (Knowles 1980). To make the learner realize the importance of a certain knowledge or skill, an instructor can design experiences of the situations, where the learner needs that knowledge or skill.

The same theory states that adults become ready to learn something when they realize that they need the knowledge to cope better in real life (Knowles 1980). Working with robots can be also seen as an answer to why he/she needs to learn it. It is important for an adult learner to have a reason why he/she needs to know the subject (Knowles 1980). With the robots the studied matter made more sense because the abstract theory or algorithm was presented in a way that students could relate to.

With robots it is easier to create an image of a situation in real life than with a program that is only showing something on a computer screen. Working with robots, students face the non-idealistic situations where the real-world problems occur (Beer, Chiel, and Drushel 1999). However, this happens in a safe environment. Furthermore, the use of robots associates computers to toys and this way reduces any possible fear of trying out and exploring (Lawhead et al. 2002). This way the learning situations with the robots are seen as an opportunity, and the effort put into them is worth it.

4.4 Teamwork

More complicated assignments invited the students to start working in groups. This was due to the large workload of the projects from the beginning. Forming groups showed that they became more competent in estimating how much time completing a project actually takes. This happened in a sense that students started to set more realistic goals for themselves compared to the beginning of the course (Kumar 2001).

Moreover, the theory of andragogy talks about using the experience of the learner as part of the teaching (Knowles 1980). The experience should be seen as a starting point to the learning process (Knowles 1980). When the students worked in teams or discussed their solutions, they used someone’s experience of something. In that sense, the previously mentioned adaptation can also be seen as a result of using the expertise knowledge of what different fields of studies or different specialization directions provide. It can lead to a better adaptation to the subjects in later courses or, as students in one publication have reported, the course problems had a positive influence on their learning (Imberman 2004).

In addition, there is an interesting possibility for the students to learn to express their ideas, but also to give and receive criticism. Students learn a valuable lesson if they see that the variety in perspectives can be helpful for solving a hard problem (Beer, Chiel, and Drushel 1999). In one of the publications, the students reported that in the beginning they had doubts about the usefulness of the course, but by the end they admitted that the course offered useful skills for the future (Imberman 2004). This course offered an opportunity to naturally work in groups, making it possible to practice both Computer Science and social skills for future needs.

4.5 Building the robots from the model

One of the problems in using the robots to teach Computer Science concepts is finding the balance between how much time can be consumed on building the robot and how much on programming. Building the robot can be fascinating and inspiring, but it can also be time consuming and frustrating. Some of the authors have resolved this problem by giving instructions on how to build the robot, and simply minor moderations are left to the students (Imberman 2004, Klassner 2002, and Kumar 2004).

But can the robot still serve the same purpose as a factor of inspiration in the learning process if the model to build the robot is given to students? It is stated that it was difficult to make the robot behave reliably (Kumar 2004), projects were more difficult than expected because the sensors did not work reliably enough (Klassner 2002), and adjustments needed to be done in the testing phase to both, the robot and the testing surface (Imberman 2003). So reflecting this to the problem of whether giving the instructions to build the robot or not is justified.

Experience states that students still became enthusiastic about building the robots even if it was just the customizing and fine tuning (Imberman 2003). The reaction of the Computer Science students also supports this when they have written on the feedback form that instructors should spend more time on planning how to build the robots, so that the time spent on designing could be reduced (Klassner 2002).

4.6 Learning more than what was taught

When analysing the exit surveys of the courses, the researcher noticed that the students learned concepts outside the curriculum. Klassner (2002) reports that the students were more confident about their skills to do multithreading tasks after taking the course with the robots than before when the course was taught in a more traditional way. Besides learning the desired notions, the students were able to obtain a skill to evaluate that their knowledge is sufficient.

According to the theory of andragogy, motivation for learning comes from the learner’s own experience of him/herself (Knowles, Holton, and Swanson 1998). Because of this, it is relevant for an instructor to acknowledge the learner’s need to have trust in his/her own abilities. With adult learners especially, these internal motivators are the most important (Knowles, Holton, and Swanson 1998). In this case, studying with multitasking programs for the robots became an accelerator for moving on to more complex domains. Because of the nature of the robot problems and solutions, the need to try something more challenging comes naturally.

4.7 Orientation to learning

The students evaluated that working with the robots helped them in understanding the complexity issues of the algorithms (Kumar 2004). With the experience of testing and seeing what the result is, the students could be able to see right away what the behaviour would look like. The robots create a performance-centered atmosphere for the learning, which is an ideal environment for adults to learn according to the theory of andragogy (Knowles 1980).

The robot can be seen as something interesting to apply the newly learned skill to. The robot offers an incentive to learning because students want to see their invention succeed (Kumar and Meeden 1998). When developing the right solution, students experience many different variations of a possible solution. Because students have a need to find the best possible solution to a problem they have encountered, the explanation for the search comes from their own need. This motivates students to learn about the less glamorous theoretical aspects of Computer Science (Kumar and Meeden 1998). With this method students are introduced to new aspects of theories behind the solutions, and they encounter aspects that may not be visible in the normal search of a solution. A researcher writes in his publication that the students stated that after the course, they have learned how to apply an algorithm to a certain problem (Kumar 2004). The process where the understanding of a problem becomes clearer little by little could be seen as a reason why the students were able to choose the right algorithm to a problem.

5. DISCUSSION

Working with the robots offers a chance to implement the code as a real-world construct. It offers a unique possibility to test the design in action right away with a minimal effort. The programmers with ten years of experience have complained that young programmers depend too much on the technology in order to complete the tasks given to them (Wolz 2001). When error is seen in action, it introduces a possibility to the students to test every modification of the code on the robot. In one of the teaching experiments, it was stated that students thought their solution was correct before testing it on the robot (Fagin, Merkle, and Eggers 2001), but can it be proven that it did not happen in all the other cases? Because if it does, it proves that designing before doing is still a skill that was learned in the old days when a batch submitted to be compiled required two days of waiting (Wolz 2001).

As much as teaching with robots has been praised, it has also been criticized. Learning to program through trial-and-error can easily be compared to learning with robots. However, it has been shown that the students tend to consider the decisions they made when writing the code, and after that they transfer fully ready solutions to the robot (Imberman 2003). Also Kumar (2004) reported that the students have shown better understanding of the complexity issues of the algorithms, and the results of the tests have revealed better knowledge of how to apply an algorithm to a problem. So if the students have the ability to decide and design the correct solution to a problem and according to that start executing their answer into a program code, it gives enough proof that code designing and management can be taught as well in the 21st century.

As much as the programming languages have developed, the platforms and programs have improved. Being able to perceive the outcome of the code is an important skill to master, but testing a program is still different from mindless re-testing. Nowadays, there are different techniques to do the coding and testing, and because of the nature of the robots, the test results give reasonable feedback and with this they direct correction in the right direction. Re-testing and negative outcomes can provide important lessons (Wolz 2001). Therefore, the whole concept of teaching and learning with robots should be seen differently. The traditional approach to programming gives students few opportunities to observe the behaviour of their code in any other context than in the debugging phase (Stein 1998). In this sense robots should not be used only to give hands-on experience, but to create an atmosphere that resembles something from real life. For future needs, it is important for the students to see how the environment around the robot affects the design. So maybe re-testing should not be compared to the designing of the code, but it should be seen as testing what effect the outside world has on the design.

With robots, the designing and implementing invites students to think of more options for how to plan the code to solve the problem, and with that students experience more aspects of the concept. When teaching is done this way, it invites students to consider not only how to build the program, but to think about what the behaviour will be and modify that behaviour (Stein 1998). Not only is programming as a skill hard to achieve, but the science of programming also includes a lot of details which are not easy to explain, nor is it easy to give a reason to the students why they need to learn them (Lawhead et al. 2002).

When teaching with robots, the programming is not a separate phase of the project, but it is attached to many parts of the project, such as designing and testing. The validation of the programming comes naturally with robots because the students are eager to see the robots work in action (Lawhead et al. 2002). The importance of connecting all these parts and making them work together is acutely present with robots. The construction of a physical entity joined with the code designed by the students themselves gives a unique opportunity to directly confront the central issues of Computer Science (Kumar and Meeden 1998). After students have designed a working robot, they have experienced some of the convergence of Computer Science, and thus can better perceive the interplay between various concepts. This is crucial because understanding the interactions between the program and its behaviour is critical in modern applications (Stein 1998).

Because of the small amount of research done on the use of Mindstorms robots to teach bigger concepts in the field of Computer Science, it was significant that one out of four reported a negative outcome. Some insight about this unsuccessful experiment by the US Air Force Academy has already been given in the analysis part of this paper. However, there are other observations as well that might explain the reasons behind the failure in the experiment.

For the limited amount of money to be spent on the robots, the researchers had to make a decision to use the robots only inside the classroom (Fagin and Merkle 2002). Even with the effort of giving as many lab sessions as possible, the simulation and testing phase was too short to make the use of the robots worth while. Researchers admit that in their experiment they were not able to give enough resources to one of the most important parts of the development of the robots (Fagin and Merkle 2002). The students also saw the unlimited time reserved for the projects as a big disadvantage. In a more traditional class, the way subjects are presented is a result of many years of teaching and examining student feedback. The reason why students in the class with robots showed worse results than those in the class without them (Fagin and Merkle 2002), could lie in this limited amount of resources.

Instructors of the Air Force Academy Computer Science course report that their students are not representative of the whole population of students, and hope that other researchers in different environments attempt a similar experiment (Fagin and Merkle 2002 and Fagin and Merkle 2004). Unlike the students in the other experiments, the students of the Computer Science course in the Air Force Academy had to design their code and built the robot within lab hours (Fagin and Merkle 2002 and Fagin and Merkle 2003); other researchers favoured and recommended to their students to use time more flexibly in their experiments. Even though the students in the other teaching experiments reported large time consumption on working on the robots at home, in the end it was considered to be rewarding, and a positive factor in their learning from both the students' and the instructors' point of views (Imberman 2003 and 2004, Klassner 2002, and Kumar 2001 and 2004). The researchers who received the negative result, acknowledge the fact that their choice to limit the time used on testing and debugging the robots is partly the cause why results were not what they expected (Fagin and Merkle 2002).

To defeat the ongoing competition inside of the university of which course gets enough students to enrol, robots can be one solution. Robots fascinate the typical student, and this interest should be used to invite students into the Computer Science curriculum (Kumar and Meeden 1998). Imberman (2004) reports that after starting to use robots in the AI course, the enrolment rate is better than before. Also Kumar (2001) reports that in the end survey when students were asked if they would recommend the course to their friends, over 90% answered yes. Besides this, those instructors who use robots in their class argue that they bring a fun factor to the class (Imberman 2004 and Kumar 2001). Even if university studies are not meant to be fun and entertaining, the experience of enjoying the class and having done exercises without feeling frustrated, should have a positive influence on the students' attitude towards studying.

The nature of the learning process is different when studying with robots than in more traditional ways. It could be considered as one option to create some variation in the Computer Science curriculum. We can still rethink the fundamental notions of computation in a way to bring teaching much closer to today's practice (Stein 1998).

This paper does not give an answer to the question what the best way to teach or approach an adult learner is. It only focuses on giving an explanation to why diverse methods could be taken into consideration when designing a course within the Computer Science curriculum.

6. CONCLUSION

These notions from the teaching experiments reflected on the support theories and diverse remarks give instructors a reason why to consider using robots in university-level education. The support theories give context-free responses to the observations reported in the publications. The positive or negative outcome can now be mirrored to the known behaviour or preference of an adult learner. With this an instructor can be confident that the outcome can be reproduced in his/her classroom. As a final aid for instructors we have collected and organized support theories and case studies in table form, see Table 1, for giving a summary of our research results.

7. REFERENCES

Beer Randall, Chiel Hillel and Drushel Richard (1999): Using Autonomous Robotics to Teach Science and Engineering, Communications of the ACM **42**(6):85-92.

Carroll John (1998): *Minimalism Beyond the Nurnberg Funnel*. 1-18. John M. Carroll. MIT Press.

Carroll John and van der Meij Hans (1996): Ten Misconceptions about Minimalism, *IEEE Transactions on Professional Communication* **39**(2):72-86.

Challinger Judith (2005): Efficient Use of Robots in the Undergraduate Curriculum, SIGCSE, Special Interest Group on Computer Science Education, *Proc. of the 36th SIGCSE Technical Symposium on Computer Science Education*, St. Louis, Missouri, USA, 436-440.

Dubinsky James (1999): Fifteen Ways of Looking at Minimalism, *Journal of Computer Documentation* **23**(2):34-47.

Fagin Barry (2003): Ada/Mindstorms 3.0: A Computational Environment for Introductory Robotics and Programming, *IEEE Robotics & Automation Magazine* **10**(2):19-24.

Fagin Barry (2000): Using Ada-Based Robotics to Teach Computer Science, SIGCSE, Special Interest Group on Computer Science Education, *Proc. of the 5th annual SIGCSE/SIGCUE ITiCSE Conference on Innovation and Technology in Computer Science Education*, Helsinki, Finland, 32:148-151.

Fagin Barry and Merkle Laurence (2003): Measuring the Effectiveness of Robots in Teaching Computer

Science, SIGCSE, Special Interest Group on Computer Science Education, *Proc. of the 34th SIGCSE Technical Symposium on Computer Science Education*, Reno, Nevada, USA, 307-311.

Fagin Barry and Merkle Laurence (2002): Quantitative Analysis of the Effects of Robots on Introductory Computer Science Education, JERIC, *ACM Journal of Educational Resources in Computing* **2**(4):1-18.

Fagin Barry, Merkle Laurence and Eggers Thomas (2001): Teaching Computer Science with Robotics Using Ada/Mindstorms 2.0, ACM SIGAda Ada Letters **21**(4):73-78.

Flowers Thomas and Gossett Karl (2002): Teaching problem solving, computing, and information technology with robots, Journal of Computing Sciences in Colleges **17**(6):45-55.

Gross Paul and Power Kris (2005): Evaluating Assessments of Novice Programming Environment, ICER, The First International Computing Education Research Workshop, *Proc. of the 2005 International Workshop on Computing Education Research*, Seattle, WA, USA, 99-110.

Handy Board (2003): The Handy Board microcontroller system, http://www.handyboard.com Accessed 13 Aug 2008.

Imberman Susan (2005): Three Fun Assignments for an Artificial Intelligence Class, *Journal of Computing Sciences in Colleges* **21**(2):113-118.

Imberman Susan (2004): An Intelligent Agent Approach for Teaching Neural Networks Using LEGO Handy Board Robots, JERIC, *ACM Journal of Educational Resources in Computing* **4**(3):1-12.

Imberman Susan (2003): Teaching Neural Networks Using LEGO Handy Board Robots in an Artificial Intelligence Course, SIGCSE, Special Interest Group on Computer Science Education, ACM SIGCSE Bulletin **35**(1):312-316.

Imberman Susan and Klibaner Roberta (2005): A Robotics Lab for CS1, *Journal of Computing Sciences in Colleges* **21**(2):131-137.

Jipping Michael, Calka Cameron, O'Neill Brian and Padilla Christopher (2007): Teaching Students Java Bytecode Using Lego Mindstorms Robots, , SIGCSE, Special Interest Group on Computer Science Education, *Proc. of the 38th SIGCSE Technical Symposium on Computer Science Education*, Covington, Kentucky, USA, 170-174.

Klassner Frank (2002): A case study of LEGO Mindstorms'™ suitability for Artificial Intelligence and robotics courses at the college level, SIGCSE, Special Interest Group on Computer Science Education, *Proc. of the 33rd SIGCSE Technical Symposium on Computer Science Education*, Covington, Kentucky, USA, 8-12.

Klassner Frank and Continanza Christopher (2007): Mindstorms without Robotics: An Alternative to Simulations in Systems Courses, SIGCSE, Special Interest Group on Computer Science Education, *Proc. of the 38th SIGCSE Technical Symposium on Computer Science Education*, Covington, Kentucky, USA, 175-179.

Knowles Malcolm (1980): *The Modern Practice of Adult Education, from Pedagogy to Andragogy*. 40-59. Englewood Cliffs, Prentice Hall, Cambridge.

Knowles Malcolm, Holton Elwood and Swanson Richard (1998): *The Adult Learner: The Definitive Classics in Adult Education and Human Resource Developmen*t. 2-5. Gulf Publishing, Houston, Texas.

Kumar Amruth (2004): Three Years of Using Robots in the Artificial Intelligence Course - Lessons Learned, JERIC, *ACM Journal of Educational Resources in Computing* **4**(3):1-15.

Kumar Amruth (2001): Using Robots in an Undergraduate Artificial Intelligence Course: An Experience report, 31st Annual ASEE/IEEE Frontiers in Education Conference, Reno, Nevada, USA, **2**:10-14.

Kumar Deepak and Meeden Lisa (1998): A Robot Laboratory for Teaching Artificial Intelligence, SIGCSE, Special Interest Group on Computer Science Education, *Proc. of the 29th SIGCSE technical symposium on Computer Science Education*, Atlanta, Georgia, USA, 341-344.

Lawhead Pamela, Bland Constance, Barnes David, Duncan Michaele, Goldweber Michael, Hollingsworth Ralph, Schep Madeleine (2002): A Road Map for Teaching Introductory Programming Using LEGO Mindstorms Robots. The Annual Joint Conference Integrating Technology into Computer Science Education, Working group report from ITiCSE, Innovation and Technology in Computer Science Education, Aarhus, Denmark, 191-201.

Lego (2008): Lego Education – Mindstorms, the Lego Group, http://www.lego.com/eng/education/mindstorms/default.asp Accessed 13 Aug 2008.

Stein Lynn (1998): What We Swept Under the Rug: Radically Rethinking CS1, *Journal of Computer Science Education* **8**(2):118-129.

Wolz Ursula (2001): Teaching Design and Project Management with Lego RCX Robots, SIGCSE, Special Interest Group on Computer Science Education, *Proc. of the 32nd SIGCSE Technical Symposium on Computer Science Education,* Charlotte, North Carolina, USA, 95-99.

Table 1. Summary of analysis of case studies

		Case studies			
Support theory	Assumption/Principle	US Air Force Academy	Ramapo College	Villanova University	Staten Island College
Andragogy	Self-concept of the learner	In the class with robots, students were not encouraged enough to move away from their old learning methods. Students felt that their learning is related to the instructor's level of knowledge about the matter taught. The course failed to help the students obtain a more independent and self-directed way of learning.	Not mentioned in research reports.	The variation in the learning process helped students obtain more than what was expected from them in the course curriculum.	Instructors were able to encourage students so that they learned more than what was expected in the course curriculum.
	Prior experience of the learner	Not mentioned in research reports.	Learning more and realizing how much resources projects need students started to set more realistic goals for themselves than in the beginning of the course.	Course problems have positive influence on students' learning.	Course problems have a positive influence on their learning.
	Readiness to learn	Even though the robot projects were hard and time consuming, students pointed out that an advantage in working with them is that they make one want to learn and that they give an opportunity to learn something totally new.	According to the end survey: students spent more time on robot projects than on the traditional projects, but they also enjoyed them more.	Not mentioned in research reports.	Students felt that the course with robots offered useful skills for the future.
	Orientation to learning	Robot projects were described as mentally challenging. They are a great application to real life computing and students learned logical problem-solving skills.	Students evaluated that working with robots helped them understand complexity issues of algorithms. Students learned how to apply an algorithm to a problem.	Not mentioned in research reports.	Machine learning is a hard concept for the students to understand. With the traditional way of teaching, students may lose their focus on what is the real problem. With robots it was easier to demonstrate and this way point out the significance.
	Learner's need to know	Projects have been described as fun and magical, but it was unclear for the students why they needed to learn the matters. Students felt that projects were irrelevant for Computer Science majors and not practical unless one is going into that career.	According to the end survey: Learning and working with robots appeared to students as a good investment.	Students valued the matter taught and therefore the course had a positive influence on students' appreciation of the issues behind the design of agents.	The fun factor has introduced the course to students who normal would not take AI. Students have returned afterwards and express that the topics covered were relevant to their work experience.
	Motivation to learn	The projects failed to connect into the students' inner motivators. Students experienced that either one understands what the problem is or then one does not, but there is no in between. Even with instructors' efforts to provide as much help as possible, students felt isolated.	With the robot projects, instructors were able to get the students to use their imagination and therefore the learning method became more effective.	Because building actual robots and programming them as a constructive activity can be viewed as inherently motivating, especially because of the rapid feedback of success and failure, students became more confident about their skills and learned to evaluate their knowledge.	Working with robots offers practice that is better related to real-world problems. Therefore students reflected their learned skills as something that is useful in future studies and working life.
Minimalism	Choose an action-oriented approach	Exercises on the course were designed in a way that students used robots built from a model, to be able to start working faster on asked problems.	Exercises in the course were designed in such a way that students used robots built from a model, to be able to start working faster on required problems.	Exercises in the course were designed in such a way that students used robots built from a model, to be able to start working faster on required problems.	Exercises in the course were designed in such a way that students used robots built from a model, to be able to start working faster on required problems.
	Anchoring the tool in the task domain	Students were expecting more answers from the instructor and not supported enough to solve the problems by themselves or find answers of their own.	Robots were used to connect the matter better into something that presents the concepts in a way that students can relate to.	Robots were used to connect the matter better into something that presents the concepts in a way that students can relate to.	Robots were used to connect the matter better into something that presents the concepts in a way that students can relate to.
	Support of error recognition and recovery	Students looked through the program code, and they immediately saw that they did not program the robot to do its new behaviour. The robot visualizes the commands, so that the students could observe the incompleteness of their code immediately and locate the problem.	Robots visualize the previously hidden process of code execution. Due to the robots' instant feedback, error recognition is more straightforward and therefore recovery is improved.	Robots visualize the previously hidden process of code execution. Due to the robots' instant feedback, error recognition is more straightforward and therefore recovery is improved.	The architecture of the robots made the conceptual difference between different notions more visual than the traditional method used before.
	Support reading to do, study and locate	Dave Baum's book was used to help the students solve problems that occurred during the learning process.	Dave Baum's book was used to help the students solve problems that occurred during the learning process.	One part of the book of Russell and Norvig, was used to clarify the topics.	One part of the book of Russell and Norvig, was used to clarify the topics.