Algorithm, Pseudo-Code and Lego Mindstorms Programming

Anthi Karatrantou[1], Chris Panagiotakopoulos[2]

[1] ASPAITE Patras, Greece, a.karatrantou@eap.gr
[2] University of Patras, Greece, Dept of Primary Education, cpanag@upatras.gr

Abstract. This paper presents a pilot study which investigated the way prospective primary school teachers handled the process of converting an algorithm - pseudo-code to a program while working with the programming environment of the Robolab programming tool of Lego Mindstorms. Participants had to program the behavior of a Lego robotic construction, using appropriate worksheets, analyzing the problem given, designing algorithms composing pseudo-codes and constructing programs in the Robolab environment. Observation of the participants' work showed that they handled all of the aforementioned processes productively and without any difficulties. They composed the algorithms easily in every step, they used the natural language to make the pseudo-codes and they converted them to a program in a simple manner. Participants found the activities very interesting from a pedagogical perspective.

Keywords: Lego Mindstorms, algorithm, pseudo-code, constructivism, ICT in education.

1 Introduction

Research from the past decade has shown that Lego Mindstorms is a powerful educational kit, suitable for teaching introductory science concepts, technology, and programming [1], [2], [3]. Especially for Robolab the programming environment, it has been suggested that this is better for children first attempt at learning to program rather than for serious programmers who want to program robots using high-level languages [1]. The use of the Lego Mindstorms also allows students to learn and have fun at the same time while working within a motivational environment [4].

The exploitation of the Lego Mindstorms in education falls in step with the concept of constructivist learning [5], [6] and the constructionist educational philosophy [7] [8]. Papert has mentioned that constructionism is built on the assumption that children will do best by finding for themselves the specific knowledge they need; organized or informal education can help most by making sure they are supported morally, psychologically, materially, and intellectually in their efforts [8]. These theories argue that children are much more motivated for learning when they can explore the world that surrounds them in a natural way [9]. In a constructionist environment, students

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

act like “real-world” scientists, inventors and engineers. So, as a result, students are in much closer contact with the truly important ideas of science and engineering. They do not simply learn facts, equations, and techniques. They learn a way of thinking critically and systematically about problems, and especially in view of the fact that they learn about the problem-solving process itself [8]. In contrast with the traditional learning environments, the constructivist approach provides tools, which allow children to build their own knowledge. In constructivism, children are explorers of knowledge rather than simple receivers of knowledge. Such a tool is the Lego Mindstorms educational kit, too [10].

On the other hand, computational thinking is a fundamental skill for everyone, not just for computer scientists [11]. However, computer programming is a difficult process [12]. Beyond knowing the syntax of a programming language, this cognitive process requires several skills [13].

In this work small groups of prospective primary school teachers utilized Lego Mindstorms and were asked to complete a number of successive activities using appropriate working sheets. They had basic knowledge on the use of Microsoft Windows but no programming knowledge. Each group was asked to solve a specific problem, working in a constructivist environment, composing the pseudo-code expressing the algorithm for the solution of the problem and finally programming Lego brick, verifying every time their program until the solution of the problem was completed. Their responses were observed and recorded every time during the process, in order to study:

(a) The way they converted the algorithm/pseudo-code to a serious program into the Robolab environment.

(b) The way they worked with the environment of Lego Mindstorms.

2 Pseudo-code and Algorithm

An algorithm is a set of precise rules that specify how to solve a problem or perform a task. The study of algorithms is at the core of computer science. Algorithms are essential to the way computers process information, because a computer program is basically just an algorithm that tells computers what specific steps to perform, and in what sequence, in order to carry out a specified task [14] [18].

Definitions of the term “algorithm” often require that the problem be solved in a finite number of steps. However, algorithms include procedures and it may be difficult to determine whether the algorithm successfully completed its task. Algorithms can be expressed in a variety of ways. Very simple algorithms can be stated using ordinary sentences in any human language. These and more complex algorithms can be shown schematically with flow charts. Programming languages and “pseudo-code” can be used to express complex algorithms [17].

A review of the literature easily confirms that there are a lot of definitions for the meaning of “pseudo-code”. It is difficult to define what pseudo-code is exactly [14], but from all definitions it can be concluded that pseudo-code is an outline of a program, written in a form of spoken language using common words that can easily be converted into real programming statements. It is a technique for describing a

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

computer program by using more general wording rather than the specific syntax and keywords of a programming language [18]. Pseudo-code cannot be compiled nor executed, and there is no real formatting or syntax rules. In other words, pseudo-code aims to fill the gap between the informal (spoken or written) description of the programming task and the final program (code) that can be executed or at least automatically converted into an executable form [15]. Pseudo-code has some advantages over ordinary human language in specifying algorithms with precision in their structure and generality. It derives its name from the fact that it resembles the source code of widely used programming languages [17].

In general, students are faced with difficulties when they work with basic algorithmic structures, as well as with the variables in programming [16] [13] [18]. The students' ability to construct or to understand an algorithm depends on their ability to construct a system of representation. One of these systems is pseudo-code. In general, since students can express their thoughts in various representing systems they can make connections between concrete, intuitional and symbolic knowledge [19]. So, the ability of every one to compose a pseudo-code (expressing an algorithm) for an activity is important, even for everyday life activities.

3 Lego Mindstorms and Robolab

The Lego kit includes hundreds of lego pieces, wheels, lamps, input sensors of various kinds, the programmable RCX (Remote Command System) brick, an infrared transmitter that establishes a wireless link between the computer and the RCX and a visual programming environment. All these permit the construction of programmable robots with remarkably sophisticated behavior [1].

Robolab is the visual programming environment (built upon the graphical programming language of LabVIEW) that enables the user to create programs using icons representing all the basic programming structures, commands and data types composing flow charts. One of the basic advantages of such programming languages is that the syntax details that students have to use, are limited, resulting in a teaching approach of the programming that is oriented to the algorithm development as well as to the development of students' critical thinking.

4 Methodology

The sample consisted of 9 fourth-year, female students, prospective primary school teachers, who worked in three separate groups with three students per group. Their average chronological age was 22 years (st.d. = 0.7 years). The participating students had already completed the course requirements for their degree and were waiting to graduate from the Dept of Primary Education of University of Patras, in Greece. The research took place in the beginning of June 2008, at the Computers and Educational Technology Laboratory (CETL) of the Department of Primary Education of the University of Patras (Greece).

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

The sample was able to work with a computer using Microsoft Windows and the Microsoft Office suite of programs. They also were experienced in using the computer as a teaching tool for searching information and as a platform for educational software aimed at the primary school level. They had no programming knowledge. Lego Mindstorms had been exhibited, in the framework of a course entitled "Computers and Education" one year earlier (during their 3rd year of studies), without the active involvement of the students.

Every group worked together with two experimenters for two sequential sessions of two hours each. Starting with the first session, about thirty minutes was spent in order to discuss with each group about Lego Mindstorms and the way they operated. The experimenters asked subjects to touch and inspect for a while one Lego RCX brick, with two motors (an assembled car).

After this, their work was supported by 6 worksheets, corresponding to 6 discrete steps. The two experimenters were watching carefully the subjects' work, keeping notes without intervening unless they were asked to help or until the experimenters decided it was necessary. So, the subjects in each group worked in collaboration in order to accomplish their mission. Their mission each time was based on the programming of the car's behaviour, since they had composed the algorithm/pseudo-code for this. The six steps with the corresponding problems for solution and the questions made were as follows:

1. Can you describe a sequence of steps in order to move the car forward for a specific time interval and then to stop it? Can you describe a sequence of steps in order to move the car forward for a specific time interval, to stop it for a specific time interval, to move again backward for a specific time interval and then to stop it?
2. Can you describe a sequence of steps in order to move the car forward for a random time interval (between 0 - x seconds) and then to stop it? Can you describe a sequence of steps in order to move the car forward for a random time interval (between 0 - x seconds), to stop it for a specific time interval, to move it backward for a random time interval and then to stop it?
3. Can you describe a sequence of steps in order to turn the car around (in the same direction) for a specific time interval and then to stop it? Can you describe a sequence of steps in order to turn the car around (in the same direction) for a random time interval (between 0 – x seconds) and then to stop it? Can you describe a sequence of steps in order to turn the car around (in the same direction) for a random time interval (between 0 – x seconds), after this to turn it round again but to the opposite direction for a random time and then to stop it?
4. Mount a light sensor on the car. Place the car on different locations in the Lab. Keep writing the different values of the light sensor. Keep writing again the different values of the light sensor when a white or a black paperboard is been placed about 15-25 centimetres in front of the car. Keep writing the value of the light sensor without any paperboard in front of the car.
5. The car is stopped. Can you describe a sequence of steps in order to move the car forward when a black paperboard is been placed in front of the sensor and not responding when a white paperboard is been placed in front of the sensor ?
6. The car is stopped in the middle of a "circle" of white and black paperboards, each one 20 centimetres width (Figure 1). On the car a light sensor and a green

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

lamp are mounted. Can you describe a sequence of steps in order to turn the car around for a random time (between 0 – x seconds), then to stop it and if a black paperboard is placed in front of the car then the green lamp should turn on, otherwise if a white paperboard is placed in front of the car then nothing should happen?

Fig. 1. The "circle" of white and black paperboards.

For each one of the six steps, subjects had to:

(a) Make the appropriate algorithm - think and write on a paper sheet the sequence of actions in their natural language (a pseudo-code) in order to describe the algorithm.

(b) Convert the pseudo-code to a program using RoboLab, in order to verify the algorithm made and to program the car.

Every time, the subjects could see the result of their program and could make it again and again, if necessary, trying to find out the correct solution.

When the educational activity was finished, a discussion took place based on a set of questions (semi-structured group interview), in order to evaluate the whole procedure and explore subjects' attitudes with regards to:

(a) The use of pseudo-code in programming.
(b) The programming in Lego Mindstorms environment.
(c) The conversion of a pseudo-code into a program in the environment of Lego Mindstorms and Robolab.
(d) The use of Lego Mindstorms in the classroom (advantages and disadvantages).

All discussions between the participants and between participants and experimenters during the experimentation process were recorded, in order to analyse it afterwards.

5 Findings - The way participants worked

While observing the subjects' work, during the implementation of the activities, as well as during the analysis of the audio recordings, the students' continuously increasing interest for the activities and dedication to their work was demonstrated. They were discussing, arguing, testing solutions and deciding in every step of the procedure.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

At the beginning, a familiarization phase took place, during which the experimenters just presented the Lego tool kit to the participants and let them touch and inspect the elements included. During this phase, the participants were in contact with the tool, their interest was triggered off and the basic idea of their work put down.

After inspecting and examining the constructed car that they would use during the whole activity, they started to work on the six worksheets. Working on the 1st one, questions like '*how will the car start moving*?', '*the wheels must turn on'*, '*yes, but how we can move it forward?*', '*the two wheels must rotate in the same direction*", '*the car has to move for a specific time interval, how?*' arose and a brainstorming of solutions took place. They had been encouraged by the experimenters to write down in physical language the sequence of actions (a pseudo-code) that they thought could move the car. A characteristic solution is: '*rotate the two wheels in positions B and C (meaning the ports B and C) simultaneously for 2s and then stop*'. Experimenters helped them to convert their pseudo-code to a program in the Robolab environment, explaining the philosophy of the software to them. The program development offered them the opportunity to test and watch the result of their designs each time, to find the correct answers to their questions and to solve practical problems concerning the move of the car.

All three groups worked successfully on the second part of the worksheet '*rotate the two wheels in positions B and C simultaneously for 2s then stop for 1s then rotate the wheels in the opposite direction for 2s and then stop*'.

It was not difficult for them to work with the 2nd worksheet but the meaning of 'random' time interval as well as its implementation in the car's move was under question. After the experimenters' explanations of 'random', the participants completed their mission with success '*rotate the two wheels in positions B and C simultaneously for a random time interval between 0 and 3s then stop for 1s then rotate the wheels in the opposite direction for a random time interval between 0 and 3s and then stop'*.

The 3rd worksheet put a great question to them: *How can they make the car turn around for a time interval?* Some characteristic dialogues between them were: '*should the wheels rotate? Of course yes, but how?', 'if we put the one wheel to rotate and not the other?* (solution 1)*, 'should the car move forward in the same time?', 'lets try to turn round the car using our hands…. look it turns round and watch the one wheels rotate forward and the other one in the opposite direction …yes! That's it!!!*' (solution 2). Two of the groups implemented the 1st solution and one group the 2nd '*rotate the wheel in position B for 3s forward and at the same time rotate the wheel in position C in the opposite direction for 3s and then stop'*. All of them were sure that they could complete their mission with the 3rd worksheet '*Its very easy…*', '*rotate the wheel in position B forward for a random time interval between 0 and 3s and at the same time rotate the wheel in position C in the opposite direction for a random time interval between 0 and 4s and then stop for 1s. Then rotate the wheel in position C forward for a random time interval between 0 and 3s and at the same time rotate the wheel in position B in the opposite direction for a random time interval between 0 and 4s and then stop'*.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

As the participants were working it was obvious that their confidence was increasing and their pseudo-codes became more and more accurate with discrete sentences, as well as more and more complex.

The 4th worksheet gave the experimenters the opportunity to explain the use of the light sensor and its function to the subjects of the study. The participants tested the function of the light sensor, measure the light intensity under different conditions and wrote down the measurements in the environment, in front of a white paperboard or in front of a black paperboard.

The 5th worksheet put a more difficult task to the participants. The car should be able to start moving forward if a black paperboard was in front of it and stay stopped if a white paperboard was in front of it. After this, for the participants the car could '*see*' the white and the black paperboard but how it could react in a different way in each case? Characteristic parts of their dialogs are: '*we say if… Is there any IF command? Can we use something for IF? How?*', '*yes, lets think what to do with IF…*', '*well, if you see (the car) a black paperboard move forward if you see the white one... Do nothing*?', '*how can the car see the black and white…*', '*the light sensor can measure the light intensity… yes, that's it…' '...watch in front of the black paperboard it can measure the values lower than 45….', '… and in front of the white paperboard higher than 45...', 'so, we found it!'*. One solution they found was: '*If in front of you (referring to the car) there is a black paperboard then start moving forward. If in front of you there is a black paperboard then it stays stopped. Black means light < 45 and white means light > 45*'. The experimenters explained to them how to use the icon corresponding to the "IF" structure in the Robolab environment and they developed their program correctly after a few trials '*If the light sensor measures a value < 45 then moves forward (rotate both the wheels forward) - if the light sensor measures a value > 45 then does nothing'*.

The 6th worksheet was a complex one and they had to solve a more completed problem '*...here we have to use all we used before!*'. All the groups had discussions in order to decide what the car should do and how to organize its behaviour '*the car must turn round for a random time interval and then has to stop'*, 'why?...'. *'...because it has to stop in order to have the time to see what paperboard is in front of it…', 'ok... if it see a black paperboard then the green lamp turns on... How long?' 'Should we set the time interval?' 'Yes because if not the lamp will be on forever….', 'ok… and with a white paperboard then it should do nothing…*'. After a few trials they found appropriate solutions. They faced problems with the light intensity values that the light sensor was measuring because now the car was in the middle of the "circle" paper-wall and the light of the sensor read was less than before. So, they had to 'calibrate' again the sensor in order to 'see' black and white correctly. All the three groups solved the problem and a characteristic pseudo-code was: '*rotate the wheel in position B forward, in the same time the wheel in position C backward for a random time interval between 0 and 4s and then stop. If the light intensity is < 40 then turn on the green lamp for 4s. If the light's intensity is > 40 then does nothing.'*

It must be noticed here that during the procedure of trying to compose the pseudo-codes, participants realized that they should be extremely accurate in their statements as well as in the sequence of the actions to be completed, because in problem solving and in programming everything must be accurately organized and designed.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

One of the groups was satisfied with just this work but the other two would like to add something more. The experimenters let them think about extensions of the program concerning the behaviour of the car. Both groups would like to command the car to start, to turn around again and again for several times. While discussing the problem they found the need of a repeat structure and asked for help. The experimenters explained about the use of the JUMP and LAND icons, as a structure of repeat of a part of a program for several times (infinite). Both groups managed to moderate their pseudo-code and program correctly in that direction and both thought to put a red lamp on the back place of the car in order to turn on in the case of the white paperboard. Difficulties arose because of the limitation, concerning the available I/O ports on the RCX Brick. The red lamp should be put on the same port of a wheel, that means both lamp and wheel start together their work. The one group could not find a solution and the experimenters helped them. In their trial and error attempts, the 3rd group found the solution: they put together the two motors (wheels), wired in different directions and alone the lamp in a different port. In this way, one of the wheels rotated forward and the other backward. Their pseudo-code where: *'the car is stopped in the middle of a paper-wall with black and white pieces of paperboard. The car starts to rotate the wheels in position B for a random time interval between 0 and 4s and then stop. If the intensity of the light is less than 40, then turn on the green lamp for 4s and then turn off. If the intensity of the light is higher or equal than 40 then the red lamp turns on for 4s and then turns off. The car repeats the procedure again and again until we press the off button'.*

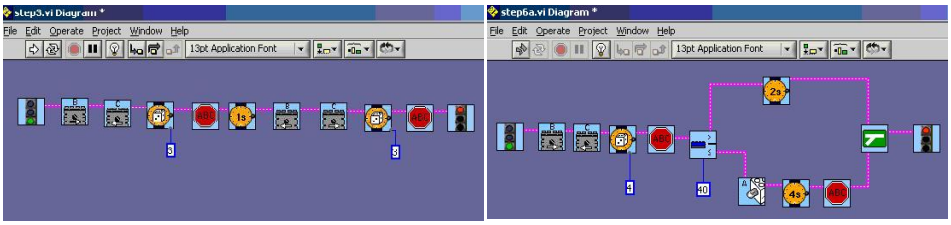

Fig. 2. Characteristic programs in Robolab of the subjects' work (from steps 3 and 6)

During the semi-structured group interview, after the end of the procedure, the attitudes of the participants appeared more intensive. From this interview we took interesting answers for the use of the algorithms, pseudo-codes and programming. More or less, all participants stated that it is easy to make an algorithm, to express it with a pseudo-code and to convert it to a program, if you are working in an environment, in which you have the opportunity to test and validate every time your action: *'…Lego Mindstorms and Robolab gave us the opportunity to work testing our actions… So whenever our actions were wrong we could reform them immediately...'*.

In addition, they stated that Lego Mindstorms could help users pleasantly, giving the motivation to compose an algorithm in order to give the desired behavior to their construction. Robolab offers a simple way to convert the algorithm expressed in natural language (pseudo-code) into a program in order to implement the desired behavior of the construction. The icons, representing commands and structures could help everyone, without previous programming knowledge to build a program. In other words, they supported that using Robolab everyone can make a program without

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

using commands with difficult syntax and strictly rules. In relation with the usefulness of the algorithms and pseudo-codes, participants argued that: *'The use of Lego Mindstorms helps you thinking reasonably and organizing the steps in order to solve a problem. The pseudo-code, especially could help to this direction…'. 'It is important for the children to learn to make algorithms, because the algorithm is necessary in every day life, in order to solve problems in a more accurate way…'. 'It is important for children to learn thinking structured…'.*

Participants found the activities very interesting and very useful from a pedagogical perspective: *'it is very important to have the opportunity to see the result of your program immediately on a 'live' construction that reacts in the way you have designed it...' 'you can learn from the mistakes … with no problem…and when you do a mistake it is the opportunity to discuss with the teacher for many things concerning programming, physics, maths', '…it a new way to learn playing!..', 'you learn how to think in order to solve a problem'* .

All of them suggested that they should try to use the Lego Mindstorms with the Robolab in the future with their students because: *'it is very important for the teacher to think and work with the students and this kit offers this opportunity… it is a new way..', 'students have to think, to write down accurate sentences in order to solve the problem and that helps them also into critical thinking and language development', 'they have to argue in order to explain and support why they design the program in the way they did and that helps them to express themselves and support their ideas'.*

On the other hand *'the cost may be high for the teacher or the school to buy the kits', 'it is time consuming for the teacher to organize the lesson', 'it is time consuming during the lesson and maybe it is difficult to fit in the daily schedule'.*

6 Conclusions

From the participants' work during the experiments and the group's interview we can conclude that they handled the process of the conversion of the algorithm/pseudo-code to a serious program effectively and without any difficulties. The Lego Mindstorms environment helped and motivated them to compose the algorithm expressing it with a pseudo-code in every step, and to convert it into a program in a simple and easy way. They worked in a constructivist environment, trying every time to find the specific knowledge needed to solve the problem. The visual environment of the Robolab, allowed them programming without text based commands and strictly rules, variables etc.

In addition, participants found the activities very interesting from a pedagogical perspective. They considered that the role of the teacher is different when using the Lego Mindstorms rather than the traditional one. From this point of view, they supported that teachers may be more like experienced advisors and their instructions are context-driven to supply what is needed.

All of them should try to use the Lego Mindstorms with the Robolab in the future with their students, because they think that this is a very important learning tool, that motivates students to think, to write down accurate sentences in order to solve problems, helping them also into critical thinking and language development.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79

References

1. Fagin B., Merkle, L., Eggers, T.: Teaching Computer Science With Robotics Using Ada/Mindstorms 2.0, Proceedings of the 2001 annual ACM SIGAda International Conference on Ada, pp. 73--78 (2001)
2. Niederer, H., Sander, F., Goldberg, F., Otero, V., Jorde, D., Slotta, J., Stromme, A., Fischer, H., Lorenz, H., Tibergkien, A., Vince, J.: Research about the use of Information Technology in science education, In Psillos D., Kariotoglou, P., Tselfes, V., Hatzikraniotis, E. Fassoulopoulos, G., Kallery, M. (Eds), Science education research in the knowledge- based society, pp. 309--322. The Netherlands: Kluwer (2003)
3. Karatrantou, A., Panagiotakopoulos, C., Pierri, E.: The influence of Lego Mindstorms Robotics constructions in the understanding of Science meanings in Primary Education: A case study. Proceedings of 5th Panhellenic Conference with International participation of ETPE - ICTs in Education, pp. 310--317 (2006)
4. Garcia, M. and Patterson-McNeill, H.: Learn how to develop software using the toy Lego Mindstorms. 32nd ASEE/IEEE Frontiers in Education Conference, Available online: http://fie.engrng.pitt.edu/fie2002/papers/1644.pdf (2002)
5. Piaget, J.: The Principles of Genetic Epistemology. New York: Basic Books (1972)
6. Piaget, J.: To understand is to invent. New York: Basic Books (1974)
7. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas, New York: Basic Books (1980)
8. Papert, S.: The Children's Machine. Rethinking School in the Age of the Computer. New York: Basic Books (1993)
9. Noss, R., & Hoyles, C.: Windows on mathematical meanings: Learning Cultures and Computers. Dordrecht: Kluwer Academic Publishers (1996)
10. Sipitakiat, A., Blikstein, P., Cavallo, D.: The GoGo Board: Moving towards highly available computational tools in learning environments. Interactive Computer Aided Learning International Workshop. Carinthia Technology Institute, Villach, Austria (2002)
11. Wing, J.M.: Computational Thinking. Communications of the ACM, 49, 3, pp. 33--35, (2006)
12. Bravo C., Marcelino, M.J., Gomes, A., Esteves, M., Mendes A.J.: Integrating Educational Tools for Collaborative Computer Programming Learning. Journal of Universal Computer Science, 11, 9, pp. 1505--1517 (2005)
13. Milková, E., Turčáni, M.: Digital objects supporting development of algorithmic thinking. In A. Méndez-Vilas, A. Solano Martín, J.A. Mesa González and J. Mesa González (Eds), Current Developments in Technology-Assisted Education, 376--380, Badajoz, Spain: Formatex (2006)
14. LINFO: Algorithms: A Very Brief Introduction (2007), http://www.linfo.org/algorithm.html
15. Roy, G.G.: Designing and explaining programs with a literate pseudocode. Journal on Educational Resources in Computing, 6, 1, pp. 1--18 (2006)
16. Spohrer, J., & Soloway, E.: Novice Mistakes: Are the Folk Wisdoms Correct? In E. Soloway & J. Spohrer (eds), Studying the Novice Programmer, pp.401--416. Hillsdale, NJ: Lawrence Erlbaum Associates (1989)
17. Roark, K.R.: Pseudocode. Available online: http://www.profroark.com/Intro_Prog_ Lecture/Pseudocode.pdf (2008)
18. Garner S.: The Development, Use and Evaluation of a Program Design Tool in the Learning and Teaching of Software Development. Issues in Informing Science and Information Technology, 3, 2006, pp. 253--260 (2006)
19. Noss, R., Healy, L., & Hoyles, C.: The construction of Mathematical meanings: Connecting the visual with the symbolic. Educational Studies in Mathematics, 33, pp. 203--233 (1997)

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 70-79