

International Conference on
SIMULATION, MODELING and PROGRAMMING
for AUTONOMOUS ROBOTS (SIMPAN 2008)
Venice (Italy), 2008 November, 3-7

TERECOP Workshop

**“Teaching with robotics:
didactic approaches and experiences”**

Venice, November 3, 2008

Program

Black-and-white-box perspectives to distributed control and constructionism in learning with robotics

Chronis Kynigos

Robotics, Computer Science curricula and Interdisciplinary activities

Javier Arlegui , Emanuele Menegatti, Michele Moro, Alfredo Pina

The use of LEGO Mindstorms in elementary and secondary education: game as a way of triggering learning

Soumela Atmatzidou, Iraklis Markelis, Stauros Dimitriadis

Notes on a Scientific Foundation for Constructionism: with a note of some difficulties

Mike Doyle

It Takes a Village... to do Science Education

Paolo Fiorini, Stefano Galvan, Loretta Giuliani, Laura Pighi

Representative examples of implementing educational robotics in school based on the constructivist approach

Stassini Frangou, Kyparissia Papanikolaou, Liliane Aravecchia, Luc Montel, Silviu Ionita, Javier Arlegui, Alfredo Pina, Emanuele Menegatti, Michele Moro, Nello Fava, Stefano Monfalcon, Irene Pagello

CEABOT: Nationwide Little humanoid robots competition; rules, experiences and new challenges

Jardón, P. Zafra, S. Martinez De La Casa, A Giménez

Algorithm, Pseudo-Code and Lego Mindstorms Programming

Anthi Karatrantou, Chris Panagiotakopoulos

Semantic and epistemological continuity in educational robots' programming languages

Emanuele Micheli, Marco Avidano, Fiorella Operto

School Net For a K-12 Small Robots Programming Project: Experiences in Primary Schools

M. Stella De Michele, G. Barbara Demo, Simonetta Siega, A Piedmont

Teachers as designers of robotics-enhanced projects: the TERECoP course in Greece

Kyparissia Papanikolaou, Stassini Frangou, Dimitris Alimisis

Using a Programmable Toy at Preschool Age: Why and How?

Janka Pekárová

Educational Robotics Initiatives in Slovakia

Pavel Petrovic, Richard Balogh

The use of Educational Robotics for the teaching of Physics and its relation to self-esteem

Sarantos Psycharis, Evi Makri-Botsari, Georgios Xynogalas

Gary S. Stager, A Constructionist Approach to Robotics

Chairmen

Prof. Dimitris Alimisis, Department of Education, School of Pedagogical and Technological Education, Patras, Greece, chair

Prof. Michele Moro, DEI (Dept. of Information Engineering), Univ. of Padova (Italy), local chair

Program committee:

Aravecchia Liliane, Institut Universitaire de Formation des Maîtres d'Aix-Marseille, France

Arlegui Javier, Public University of Navarra, Spain

Balch Tucker, Georgia Institute of Technology, USA

Dagdilelis Vassilios, University of Macedonia, Greece

Ezpeleta Joaquin, University of Zaragoza, Spain

Kraetzschmar Gerhard K., Bonn-Rhein-Sieg University, Germany

Marciano Giovanni, IPSIA (Tech. and Profess. Superior Education Institute) Borgosesia, Italy

Menegatti Emanuele, University of Padua, Italy

Mikropoulos Tassos, The University of Ioannina, Greece

Montel Luc , Institut Universitaire de Formation des Maîtres d'Aix-Marseille, France

Muscato Giovanni, University of Catania, Italy

Operto Fiorella, Scuola di Robotica, Italy

Papanikolaou Kyparissia, School of Pedagogical & Technological Education, Athens, Greece

Pina Alfredo, Public University of Navarra, Spain

Pretto Alberto, IT+Robotics, Italy

Ribeiro Fernando, University of Minho, Portugal

Sklar Elizabeth, Brooklyn College, City University of New York, USA

Introductory notes to the workshop “Teaching with robotics: didactic approaches and experiences”

Dimitris Alimisis¹, Michele Moro²

¹Dept. of Education, School of Pedagogical & Technological Education, Greece

²Dept. of Information Engineering, University of Padova, Italy

alimisis@otenet.gr, mike@dei.unipd.it

Over the last two decades interest in educational utilization of robotics at all school levels has increased. Educational robotics is introduced as a powerful, flexible teaching/learning tool stimulating learners to control the behavior of tangible models using specific programming languages (graphical and textual) and involving them actively in facing authentic problem-solving challenges.

The European project “Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods - TERECOP” (2006-2009) is being activated in the field of educational robotics with the participation of 8 European educational institutions from 6 European countries (www.terecop.eu), aiming at the development of a design and implementation framework for activities advisable mainly for secondary school education related to programmable robotic constructions and based on learning methodologies inspired from constructivism and constructionism theory.

Believing that the role of teacher is crucial for the successful introduction of robotics in classrooms, the project activities include also the training of prospective and in-service teachers on the use of robotics technologies (LegoMindstorms Education NXT) through courses implemented in each of the six participating countries, the evaluation of the training courses and the dissemination of the educational results at a European level. Finally the TERECOP project aspires to develop a community of practice between researchers, teacher trainers and teachers that will facilitate and sustain teachers’ professional development in the use of robotic tools in classrooms.

In the frame of its dissemination activities, the TERECOP project organized the workshop “Teaching with robotics: didactic approaches and experiences” hosted by the SIMPAR2008 (Simulation, Modeling and Programming for Autonomous Robots) conference held in Venice, Italy, 3-6 November 2008. The papers presented in this workshop address a wide range of both theoretical and practical aspects of educational robotics.

Some critical theoretical aspects behind the educational use of robotics are discussed and analysed by Kynigos (*Black-and-white-box perspectives to distributed control and constructionism in learning with robotics*) with respect to potential of control technology to generate constructivist learning processes and to address learning domains such as science and mathematics.

Doyle (*Sketch for a Scientific Foundation for Constructionism: with a note of some difficulties*) outlines a model that offers the potential to provide a scientific foundation for the constructionist approach and also offers a possible explanation of the tenacity of the instructionist approach.

Experiences from implementation of various educational robotics activities are reported in other papers related to different school and academic levels extended from kindergarten to computer science education.

Pekarova (*Using a Programmable Toy at Preschool Age: Why and how*) examines the new dimension that Robotic toys bring to role-play activities in kindergarten.

Fiorini et al. (*It Takes a Village... to do Science Education*) describe the efforts undertaken by a small community of concerned teachers to boost science education in the school district of Verona (Italy) by promoting constructivism with the help of various configurations of robotic devices.

Frangou et al. focus on the design of robotics enhanced activities (*Representative examples of implementing educational robotics in school based on the constructivist approach*) and present six examples created for and used in the teachers' training seminars organized in the context of the TERECOP project.

De Michele et al. (*A Piedmont SchoolNet for a K-12 Mini-Robots Programming Project: Experiences in Primary Schools*) present a project originated and carried out by primary school teachers to promote Papert's constructionism in a cooperative environment setting up a model of minirobot programming experiences.

Micheli et al. (*Semantic and epistemological continuity in educational robots' programming languages*) analyse some new open-source software for the programming of educational robotic kits which can accompany the student from pre-school age to high school.

Arlegui et al. (*Robotics, Computer Science curricula and Interdisciplinary activities*) present some interesting examples on how to use robotics in order to foster learning of complex computer science concepts.

Experiences from non-formal education are reported as well including game playing educational activities (Atmatzidou et al., *The use of LEGO Mindstorms in elementary and secondary education: game as a way of triggering learning*) and national competitions in robotics in Spain (Jardón et al., *CEABOT: Nationwide Little humanoid robots competition; rules, experiences and new challenges*) and in Slovakia (Petrovic and Balogh, *Educational Robotics Initiatives in Slovakia*).

Finally, two papers deal with teacher training in educational robotics. Papanikolaou et al. (*Teachers as designers of robotics-enhanced projects: the TERECOP course in Greece*) report experiences and evaluation results from the training course organised by the TERECOP project in Greece. Karatrantou and Panagiotakopoulos (*Algorithm, Pseudo-Code and Lego Mindstorms Programming*) present a pilot study which investigated the way prospective primary school teachers handled the process of converting an algorithm - pseudo-code to a program working with the Robolab programming environment.

This workshop aspires to bring closer researchers, academic and school staff working in the field of educational robotics and to contribute to the further development of the dialogue in this field especially under the light of constructionism, not only within the TERECOP project partnership but within the broader European and international community of educational robotics. This dialogue will hopefully continue and the TERECOP partnership is willing to organise and participate in new relevant initiatives in the future.

We would like to thank the organisers of the *International Conference on Simulation, Modeling, and Programming for Autonomous Robots* (SIMPACT2008)

where this workshop was kindly hosted, the members of the workshop program committee and all the authors of the papers for their valuable contribution to the success of this workshop.

The chairs

Prof. Dimitris Alimisis, ASPETE, Greece

Prof. Michele Moro, DEI Univ. of Padova, Italy

Acknowledgements

The workshop “teaching with robotics: didactic approaches and experiences” was organised in the frame of the SIMPAR2008 Conference and partly funded by the TERECoP project (European Programme Socrates/Comenius/Action 2.1, Agreement No 128959-CP-1-2006-1-GR-COMENIUS-C21 2006 – 2518 / 001 – 001 SO2). The TERECoP project has been co-funded by the European Commission in the framework of the Socrates/Comenius Programme. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Black-and-white-box perspectives to distributed control and constructionism in learning with robotics

Chronis Kynigos

Educational Technology Lab, School of Philosophy, University of Athens
kynigos@ppp.uoa.gr

Abstract. The paper discusses compromises to transparency in the design of robotics kits for learning so that users can engage in meaningful, interesting and challenging constructivist activity through the control of robots and/or their environment. Aspects of control are analyzed with respect to their potential to generate constructivist learning processes and to address learning domains such as science and mathematics. The paper focuses on a set of robotics exhibits specially designed for a serious game exhibition centre called ‘Polymechanon’ in Athens.

Keywords. Constructionism, control, distributed control, generation of meanings

Controlling and constructing robots as a constructionist environment

Construction and control were the first powerful ideas on the use of computational media for learning (Papert, 1980). With respect to digital media, this idea involved the transition from black-box software to the design of transparent (white-box) digital artifacts where users could construct and deconstruct objects and relations and have a deep structural access to the artifacts themselves (diSessa, 2000, Resnick et al, 2000). It also involved the idea of distributed control where multiple users worked with the same digital artifact either in presence or remotely from different computer screens so that they would express their ideas in collectives rather than work individually (Mor et al, 2006). However, the existence of such media did not bring about the envisaged radical changes in learning environments based on their use (Papert, 2002). Students fell onto ‘plateaus’, unable to progress beyond a certain point and found that they could not construct something very interesting when starting from scratch every time. To address this problem, black-and-white-box design perspectives provided users with generic black box artifacts which they could then use as building blocks for their constructions with exploratory digital media (for a discussion see Kynigos, 2004).

In the use of robotics, we saw a parallel transition from black box situations of pre-programmed pre-fabricated robots aimed for the workplace to white box designs

2 Chronis Kynigos

where children can construct and program robots from scratch. However, there has been little or no attention given to distributed control and black-and-white-box solutions where students can start from something complex and interesting and then move on to learning by constructing robots and programs to control them.

So, what kinds of learning can be nurtured in learning environments based on the construction, programming and control of robots? What meanings and concepts can be understood in such environments? Do they afford added value to the fostering of creative thinking?

The main learning theory which has been perceived as useful for addressing the questions has been that of a special kind of constructivism termed ‘constructionism’ by Papert and his group at the Media Lab (Kafai & Resnick et al., 1996). Constructivism originated from Piaget and perceives learning as the generation of meanings from individuals as they eternally strive to bring some cohesion to the ways in which they see the world (Fosnot, 1997, Brooks & Brooks, 1993). Tangible concrete experiences with the physical and social environment are used to create generalizations, discriminate invariants and construct abstractions. Constructionism can be seen as a special case of learning in situations where we make or tinker with an object or an entity. It was seen by Papert as one of the ways in which thinking can be manifested, made public. Constructing was seen as an emergent activity where a lot of back and forth went on, where design is part of the process of building rather than a pre-requisite and where building involves de-construction and re-construction rather than just construction (Kynigos, 1995). In coining the term, Papert wanted to convey a slightly differing perception of learning than Piaget, i.e. that humans do not necessarily strive for cohesion but are by nature engaged in questioning their view of the world. Constructionism was elaborated in the early eighties at a time when individualistic cognitive theories were at the forefront and was thus associated an individualistic perception of learning. However, notions of collaborating and communicating during constructivist activity were firstly articulated as far back as the mid eighties (Rogoff & Lave, 1984,) and have since become more and more pertinent as digital technologies have made it possible for more than one students to have access to the same construction at the same time (Mor et al, 2006). This has not however happened yet with mechanical technologies and robotics.

In any case, these perceptions of learning seem to fit very well with the activities of constructing robots and programs to control them. The robotics industry aims at humans using pre-programmed pre-fabricated robots to do arduous, repetitive, mundane, fast, precise, dangerous or physically impossible things form them. The ways in which the robots are made and programmed is a black box for their users. It is the same paradigm with which many technologies are constructed from hardware to software and digital tools. It is also compatible with the traditional educational paradigm of the teacher or the curriculum book revealing and explaining ready-made ratified and thus unquestioned information.

In the framework of progressive and contemporary educational paradigms, construction and programming of robots have been made transparent so that individuals can engage in building and in programming robots themselves. Two main technologies have been so far designed and built for students to engage in robotics, the Lego-mindstorms and the Pico-crickets kits from the Media Lab at MIT (Resnick et al, 1996, Resnick et al, 2006b). This white-box metaphor for construction and

programming has generated a lot of creative thinking and involvement in learners mainly in informal educational settings. However, as in the case of digital media, there seems to be a plateau which learners reach with respect to what kind of robots they make and what they can program them to do. It quickly becomes very difficult for anyone to construct a technically robust and interesting robot and to program it to do complicated and interesting things. This was noticed some time ago as in the case of Pico-cricket where there was an expansion of the kinds of sensors and the kinds of constructions students could make (Martin et al, 2000) in order to enhance for instance the interest of female students.

An important part of learning with robots, apart from constructing and programming them, is controlling them or their environment in play. This has been rather under-exploited from an educational point of view precisely because of the white-box metaphor of starting with robotics. Controlling robots however, can provide an avenue for black-and-white-box perspectives where students can have distributed control of specific robots in amongst others. This is seen as part of a complex learning environment also embedding the construction of robots and programs to control them as usual but different in that there is also emphasis on interesting learning activity with robot control.

In this paper, I consider robot control as an integral part of constructionism and describe and discuss a series of interactive exhibits designed for learners to control in interesting game situations and made available at a special informal serious games centre in Athens which we call 'Polymechanon'. I suggest that robot control can be perceived as an integral part of constructivist engagement with robotics and that given devices and setups where control is designed to be interesting, students can learn from the kinds of feedback they get from their activities and intentions to control the robots or their environment and from the kinds of representations available to them for control.

Control and constructionism

Robotics are an integral part of control technology. The ways in which humans control machines, the semantics of the interfaces through which they control them and the discrimination of what it is they are controlling in a certain machine behavior are becoming more and more pertinent for people to understand. The number and variety of automated machines that we control in our everyday lives is increasing continually and rapidly. Think of automatic doors, alarms set by motion detectors, lights put on by clapping. We interact with them all the time but have little idea of how they work. On the other hand, these are devices designed for our everyday lives, the workplace, the home, the public places such as airports etc. Consider devices set up for humans to learn things as they control them to do something interesting. For instance, the ways in which robots respond to changes in the environment and to which changes they do respond are very important concepts. Discriminating the kinds of things we can control robots to do and by consequence gaining insight into the way they are programmed in situations which are more complex than what can be constructed by typical construction kits has also been overlooked. The means by which we can

4 Chronis Kynigos

control robots and the semantics of the devices we use to control them can operate as mechanisms through which we express our thinking, as expressive media. We do not need to wait for learners to build their own programmed robots in order to address these issues.

In Polymechanon we thus designed a series of interactive exhibits where visitors would be directly immersed in collaborative games where the more they understand what they control and how the robots respond to environmental change the better players they become. The concepts behind the games are –

- which robot behaviors can the human control,
- what kind of control do they have on these behaviors,
- how do these behaviors affect the game at hand,
- which behaviors are not controllable.

With respect to the robot's environment

- can the human control aspects of the environment and if yes which aspects can they control and which are out of control.
- How do the robots respond to aspects of the environment.
- Do the robots have consistent or changing roles in the game at hand.

A case for control: the 'Polymechanon' site.

At the Educational Technology Lab (<http://etl.ppp.uoa.gr>), after more than 15 years of design research involving the infusion of pedagogical innovations in schools based on the use of digital technologies, we felt it was time to think outside the box and consider informal education contexts where we would be at liberty to think of innovations without the constraints of the schooling system. Our main interest has been in the design of learning environments based on the use of microworlds (Sarama & Clements, 2002) embodying concepts and representations with which students generate meanings through constructions, experimentations and argumentation amongst themselves and with their teacher (Kynigos, 2007, Kynigos&Latsi, 2006).

Our aim in venturing towards informal educational settings is to consider ways of using technologies that are becoming available and affordable such as robotics, in order to design learning environments within the above framework but not constrained by the schooling context. With respect to learning process we are interested in exploring fusion between action (movement), representation, construction, experimentation and argumentation. With respect to content we are interested in the fields of mathematics, science-kinematics/mechanics/forces and spatial awareness-orientation. In order to create successful informal settings these environments need to be ecologically and culturally tuned to activities of 8-15 year olds. Our design therefore is based on serious gaming and on relatively quick immersion with games and less support from more experienced others than in the school setting.



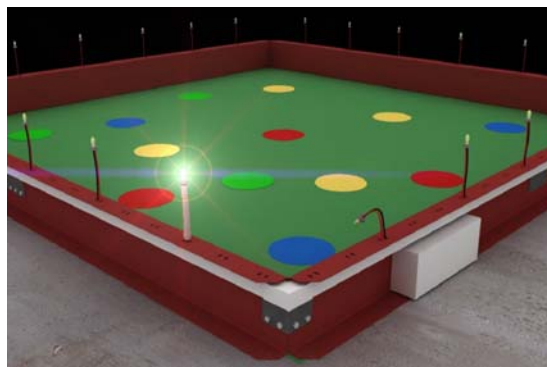
A series of robotics games have been developed and are available at 'Polymechanon', which is a place where visitors can engage in social games which require the use of computational interfaces to control machines and software. In the process of setting the site up, we have collaborated with interaction designers and robotics specialists. A description of the exhibits-

games we are developing as a first phase to setup the Polymechanon site follows. The exhibits are thus based on the principle of quick immersion and low support. However, our next aim is to organize courses and seminars where visitors will spend more dedicated time and will have much deeper access to the rules and relations behind the games, will be able to create their own and try them out.

The main idea for the robotics exhibits is based on communal control of semi-programmed robots. The point is for visitors to get engaged with an interactive game and to generate meanings and intuitions regarding programming and behaviors of robots. Each exhibit consists of a number of robots (8-10) roaming in an arena with a 4x5 meter area. The robots have been programmed to a certain degree, meaning that they have a pre-programmed behavior (reaction to stimuli, roaming under specific constraints). In the 'grazing' game players control the ways in which the robots respond to systematically changing external stimuli. In the 'traffic jam' game, players control the stimuli, i.e. the lines on which the robots roam. In 'the chase' players control the line paths where robots roam, but also have to handle changing roles amongst the robots themselves. Below is a short description of the three robotics exhibits.

Grazing

Eight robots roam in an arena with a 5X5 meter area. A number of lights are placed around the arena. The lights systematically come on and off for a few seconds in a way which is out of control of the players. There are four teams of three players. Each team can control two specific robots out of the eight. They don't know which ones they are and need to find that out by changing controls and noticing how the robots behave. Each team can

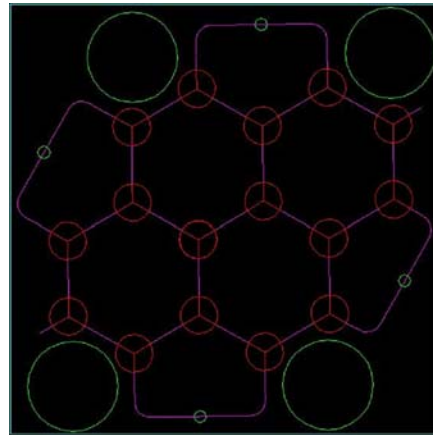


6 Chronis Kynigos

change the type of reaction to light (aversion, attraction) of each of the two robots they control. They can also change the intensity of reaction thus changing the speed with which the robot goes towards or away from the light respectively. By changing these parameters the players try to get their robots to roam over colored areas on the floor and collect points. They need to negotiate and make judgments on how they want each robot to respond to an upcoming change in the lighting and predict such changes. They need to make decisions on which colored areas to go over since each one collects a different number of points and they need to collect a specific number in order to win.

Traffic jam

The players in this game take on the role of traffic controllers, a bit like the well known movie 'The Italian Job'. They have no control on the robots, only on their environment. Five robots roam along a grid of lines in an arena 5x5 meters. Four teams of three players each play the game. Each team tries to lead all the robots under a designated arch. The players control the line which is to be active as a robot approaches a node. The robot follows the active line. The robots will not collide but keep a small distance between them when close. The players control which lines are active by means of a touch screen, one for each team of three players. They can only control three lines at a time. Out of the three players in a team, one controls the node selection and the other two which line to activate. Each team can make life difficult for others by selecting a node and making the robots go away from the others' arcs. They collaboratively develop strategies for estimation, combinatorics and the mapping of the representations on their screens with the physical robots.



The chase

This is a game resembling the digital game 'pac-man'. Eight robots are placed in an arena which has a grid in the form of colored hexagonal figurations so that there are three colors in each node. Grid lines of different colors end up in each node. The players need to guide the robots by defining the color of the line they want the robot to follow after the next node. This is



done by means of a specially designed UI on a touch screen. Several robots are in the arena and each one can be driven independently from one of four touch screens. Each player controls one robot. They can define the speed, the direction and the color of the line as described. The robots are predators or prey. The predators have a red light and the prey a green one. They are evenly split at the beginning. The predators want to get close to a robot – prey and touch it. When this happens the robots exchange roles. The one to get points is the prey (and the predator at the moment of contact). The points are scored at constant rate as long as a robot is prey. When the prey robot moves slowly more points are scored in relation to when it moves quickly. When two robots of the same role touch, nothing happens, they continue after a few seconds. The players can see their name and score on a big screen, the game lasts for a set period of time.



Discussion

This set of exhibits was designed so that players would immerse themselves with games which would require them to progressively understand and discriminate what they are controlling. The semantics of the controlling interfaces were designed for them to make links between the mechanical objects and the controlling symbols. Players could control robot response and its intensity, paths for robots to roam over, robot roles. The setting was designed with a black-and-white-box perspective in that players could change parameters and direction of pre-programmed behaviors as well as aspects of the robots' environment. They could thus think about the kinds of sensors and the kinds of programs built in the system. This whole activity is seen as situated in a broader activity of the visitors to the 'Polymechanon' site where white box kits like Pico-crickets and Lego Mindstorms kits would co-exist.

This design for learning environments raises many challenging questions for further research. How can we develop principles and methods for black-and-white-box oriented design of environments for learning with robotics? What kind of interfaces can enable students to begin from interesting games and subsequently de-construct them, inserting their own rules and robot behaviors? We need to re-think the issue of controlling technologies not only as an object to learn but also as a learning process. This poses pedagogical challenges such as the need to understand possible links with other learning domains such as mathematics and science. It also poses technical challenges, such as the need to find ways of making robots cheap, robust

and in kit form. What new ideas are there for meaningful and practical kits, i.e. robots with a core component and different ‘hats’, pluggable sensors and motors, generic robot parts. Now that technology is allowing us to have access to more complex and robust robots it is an opportunity to re-consider constructionist learning processes within domains which may make robotics more attractive to communities thinking of a school which may become more relevant with today’s society and with learning itself.

References

- 1 Brooks, J. and Brooks M. 1993. *The case for constructivist classrooms*. Alexandria, Virginia: Association for Supervision and Curriculum Development.
- 2 Fosnot, C. 1997. *Constructivism: Theory, perspectives and practice*. New York, New York: Teachers College Press.
- 3 Mor, Y., Hoyles, C., Kahn, K., Noss, R. and Simpson, G. (2006) Designing to see and share structure in number sequences, *International Journal for Technology in Mathematics Education*, 13, 65-78.
- 4 diSessa, A. (2000). *Changing minds, computers, learning and literacy*. Cambridge, MA: MIT Press.
- 5 Goldstein, R., Kalas, I., Noss, R., and Pratt, D. 2001. Building rules. *Cognitive Technology 2001 Proceedings*. 267-281.
- 6 Martin, F., Mikhak, B., Resnick, M., Silverman, B., and Berg, R. 2000. To Mindstorms and beyond: Evolution of a construction kit for magical machines. *Robots for kids: Exploring new technologies for learning*. Morgan Kaufmann.
- 7 Kynigos, C. (1995). Programming as a Means of Expressing and Exploring Ideas in a Directive Educational System: Three Case Studies. *Computers and Exploratory Learning*, diSessa, A, Hoyles, C. and Noss, R. (eds), Springer Verlag NATO ASI Series, 399-420.
- 8 Kynigos, C. (2007) Half-Baked Logo Microworlds as Boundary Objects in Integrated Design, *Informatics in Education*, 2007, Vol. 6, No. 2, 1–24, Institute of Mathematics and Informatics, Vilnius.
- 9 Kynigos, C. and Latsi, M. (2006) Vectors in use in a 3D Juggling Game Simulation, *International Journal for Technology in Mathematics Education*, 13.1, 3-10.
- 10 Kynigos, C. (2004). A Black-and-white-box Approach to User Empowerment with Component Computing, *Interactive Learning Environments*, Carfax Pubs, Taylor and Francis Group, Vol. 12, Nos. 1–2, 27–71.
- 11 Kafai Y., Resnick M. (Eds) (1996) *Constructionism in Practice*, Lawrence Erlbaum Associates Publishers, Mahwah, New Jersey.
- 12 Papert, (2002). The Turtle’s Long Slow Trip: Macro-Educological Perspectives on Microworlds. *Journal of Educational Computing Research*, 27(1), 7-28.
- 13 Sarama, J., & Clements, D. (2002). Design of Microworlds in Mathematics and Science Education, *Journal of Educational Computing Research*. 27(1), 1-3.
- 14 Resnick, M., Martin, F., Sargent, R., and Silverman, B. 1996b. Programmable bricks: Toys to think with. *IBM Systems Journal*. Volume 35, No. 3&4.
- 15 Resnick, M., Bruckman, A., and Martin, F. 1996. Pianos not stereos: Creating computational construction kits. *ACM Interactions*. September/October 1996.
- 16 Papert, S. (1980). *Mindstorms. Children, Computers and Powerful Ideas*. New York: Basic books.

Black-and-white-box perspectives to distributed control and constructionism in learning

with robotics 9

- 17 Resnick, M., Berg, R. and Eisenberg, M. (2000) Beyond Black boxes: Bringing transparency and aesthetics back to scientific investigation, *Journal of the Learning Sciences*, (9) 7-30.
- 18 Rogoff, B., Lave, J. (Ed.) (1984). *Everyday Cognition: Its Development in Social Context*. Cambridge, MA: Harvard University Press.

Robotics, Computer Science curricula and Interdisciplinary activities

J. Arlegui¹, E. Menegatti², M. Moro², A. Pina¹,

¹ Public University of Navarra, Dept. of Psychology and Pedagogy & Math and Computer Engineering
31006 Pamplona, Spain,
{arlegui,pina}@unavarra.es

² Univ. of Padova, Dept. of information Engineering
35131 Padova, Italy
{emg, mike}@dei.unipd.it

Abstract. In this paper, we present four examples of how to use robotics to foster student learning of complex Computer Science concepts. We propose to use Robotics not as a subject on its own, but as a tool for teaching/learning purposes. Following the examples presented in this paper, we discuss several ideas about Computer Science curricula, inter-disciplinary activities and teaching-learning methodologies.

Keywords: Robotics, Constructivism, Computer and Information Science Education, NXT.

1 Introduction

The 2005 ACM Computing curricula report [6] presents a reasoned guide to the topics in the different kinds of computer science degree programs they are proposing. Among the computing and non-computing topics, we find that learning these topics could be reinforced by the use of Robotics as a learning tool (especially for Computer Architecture and Organization, Software design & development, Mathematical foundations and Interpersonal communication). In this work, we are focusing on Robotics not as the field of study, but as a tool to teach other subjects in a computer science curricula (or more in general, in scientific curricula). From the point of view of innovation in the computer Science Curricula, Denning and McGettrick point out that “The first challenge is to embed the foundational practices of innovation into the curriculum, so that students learn innovation by doing.....The intention is that innovation should become an essential aspect of their attitude of mind....”[7]. The curricula in computer science (an other disciplines) should innovate, using for example “learning by doing” formula and should re-think the Theory-Practice equilibrium during the 3/4/5 years study of a degree (and maybe offer topics like Robotics in the first year, with a “Learning by Doing” approach). To carry out this kind of curricula innovation, we need to deeply revise methodological issues [8][9]. From our didactical experience, we see that an adequate educational use of robots in

computer science can promote a proactive learning and a cooperating work by groups (defining the right group problems to be solved and leaving the groups to evolve themselves); and this has to deal with methodological issues. Even if these aspects are not discussed in this paper, the authors are working under a theoretical constructivism/connectionism background and with enquiry based or group project based approaches. This is the approach followed in the TERECoP project (Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods, <http://www.terecop.eu/>). The main goal of the project is to develop a framework for secondary-level teacher education courses in order to enable teachers to implement a robotics-enhanced constructivist learning in school classrooms [1][2][10]. At the same time, the authors of this work at the University of Padua have a long-term experience in RoboCup, one of the most important robotics competitions [<http://www.robocup.org/>]. The University of Padua since 1997 has a RoboCup team composed of master students and organizes competitions like the Seventh RoboCup International Competitions and Symposium in 2003 (Padua, Italy [3]). The activity of coordinating and guiding several teams of students in building and programming the autonomous soccer robots gave us the possibility to understand how a practical realization of a robot can contribute to stimulate the students' interest and skills in ICT related technologies (and other non computing abilities). Two examples of robotic projects not related to soccer but realized by students previously involved in our RoboCup team can be found in [4] and [5]. In this paper we present four examples of possible implementations of interdisciplinary activities for Computer Science curricula using robotics as a tool.

The paper organization is as follows. In Sec. 2 we describe the robotic platform we selected for implementing the proposed didactical experiences. In Sec. 3, 4, 5 & 6 we present examples in the typical computer science field (thread synchronization and multitasking, analytical vs numerical approaches applied to the robot self-location problems, sorting problems, and a simple implementation of the Turing machine). At the end some conclusions and reflections are outlined.

2 Needed/Wanted features for the robotic platform

We considered different robotic platforms that could fulfill some requirements like to allow programming with different paradigms & levels, to offer many degrees of complexity (to be able to be used in pre-university levels) or to remain simple but with significant possibilities of expansion . Our final choice was the NXT LEGO technology, because it fullfills the previous requirements and moreover it is possible to start working with it almost immediately (no electrical or other hardware or software arrangements are necessary). Another advantage of the NXT LEGO technology we are interested in is the different programming languages and programming environments available. For instance, with the NXT LEGO is possible to use the original LEGO graphical programming environment NXT-G, or the C-like NXC or the Java based LeJOS-NXJ. Moreover, one has the possibility to use several operating systems and/or platforms (URBI, Universal Real-time Behavior Interface,

for Windows, Mac OS X, Linux or NXT-Symbian running on Symbian 6.0 Java-enabled mobile phones).

3 An example in Synchronization & Multitasking (Operating Systems topic)

3.1 Objectives

Multitasking and Synchronization are fundamental concepts in courses like: Operating Systems, Advanced/Concurrent Programming, Real-time Programming. A deep comprehension of the reasons of introducing multitasking can be achieved only running simulated or real examples of simultaneous tasks, particularly when they show interferences and synchronization/communication needs. Robotics can provide a real environment where the need of multitasking is easily shown by means of simple multi-behavioral examples.

3.2 Carrying out the experience

The NXT robot is constructed as a basic “tribot”, a cart with two independent driven wheels and a caster wheel on the rear. This enables to turn left or right applying different powers to the two motors. A third motor moves up and down an arm: this action is independent from the turning motion. Three sensors are connected: one light sensor directed to the ground, one sound sensor, and one touch sensor enabling the user to provide an asynchronous signal.

Three robot behaviors are programmed into the robot. The first behavior is the so called line follower: the robot follows the edge of a thick black line by swinging left to right and vice-versa depending on the reading of the brightness sensor. (figure 1). The robot follows clockwise the internal edge of the line by turning left when the brightness is over a certain threshold, and by turning to the right when the reading is under the threshold. For this first behavior, the controlling program is a infinite loop with a switch, based on the light sensor, between the two described motion commands.

The second behavior is to lower the robot arm for a certain number of seconds when a loud sound (e.g. a beat of hands) is detected. Also this behavior can be implemented with an infinite loop. The code of these two loops on their own is straightforward and not particularly significant, therefore we do not present it in detail. However, if one wants to activate both behaviors at the same time, a simple solution could be to insert the body of the two loops above in sequence as the body of a single loop.

```
LoopUntil(FOREVER) { // "Sequential" solution
  if (LightSensor(IN_PORT_3) > THRESHOLDLIGHT)
  { "turn left" } else { "turn right" }
```

```

if (SoundSensor(IN_PORT_2) > THRESHOLDSOUND)
{ "arm down" "wait" "arm up" }}

```

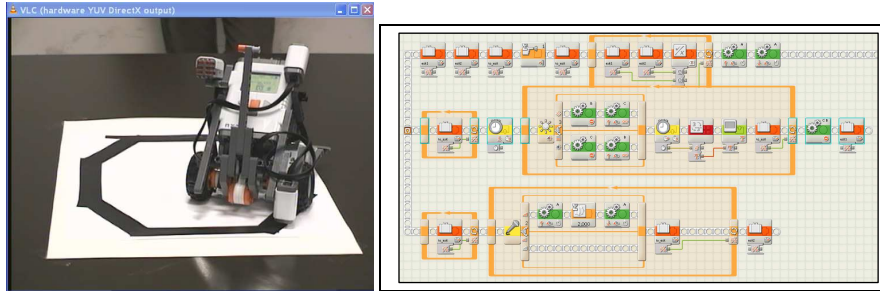


Figure 1 & 2. Scenario for the first example & defining 3 threads in NXT-G

Running the program, the robot shows very effectively the non controlled interaction between the two behaviors that arises. If the “wait” before the “arm up” command is of several seconds the robot will not turn left when crossing the black line (because the processing of the brightness sensor is delayed) and the robot will exit from the circuit stripe, failing its main behavior of line following.

This negative interference can be avoided allowing two different tasks to control separately the two behaviors, provided that some form of time sliced scheduling is implemented in the run time environment, as in the case of NXT. Next code allows to verify a correct multitasking behavior for the robot (the scheduler actually maintains active both tasks).

```

// "Multitasking" solution
Task followLine() {
  LoopUntil(FOREVER) {
    "Follow line code"
  }
}
Task Arm() {
  LoopUntil(FOREVER) {
    "arm down up code"
  }
}

```

Now, think to add a third behavior to stop the robot when the touch sensor is pressed. This lead to the need of a synchronized solution where the two controlling loops are exited in specific points as soon as possible after the touch sensor has been pressed (using common synchronization variables, see code & figure 2).

```

Task followLine() { // with synchronization
  LoopUntil(LOGIC, Var(toExit, READ), FALSE) {}
  // wait initial synchronization
  LoopUntil(LOGIC, Var(toExit, READ), TRUE)
  {"Follow line code"}
  // main loop exited when the variable is true
  Move (OUT_PORT_BC, STOP, BRAKE);
  // stop definitely
  Var(exit1, WRITE, TRUE); // ended signal
}
Task Arm() {
  LoopUntil(LOGIC, Var(toExit, READ), FALSE) {}
  // wait initial synchronization
  LoopUntil(LOGIC, Var(toExit, READ), TRUE)
}

```

```

    {"arm down up code"}
    // main loop exited when the variable is true
    Var(exit2, WRITE, TRUE); // ended signal
}
Task StopRobot() {
    Var(exit1, WRITE, FALSE); // init variables
    Var(exit2, WRITE, FALSE);
    Var(toExit, WRITE, FALSE);
    WaitUntil(TOUCHSENSOR, IN_PORT_1, PRESSED);
    // wait for touch sensor pressed
    LoopUntil(LOGIC,
    And(Var(exit1, READ), Var(exit2, READ)), TRUE){}
    // wait for two tasks completion
    Move (OUT_PORT_A, FORWARD, 30, DEG, 50, BRAKE);
    Move (OUT_PORT_A, BACK, 30, DEG, 50, BRAKE);
    // a final event, the arm moves up/down for 50 degrees}

```

3.3 Analysing the results

The usefulness of both multitasking and synchronization is made evident with simple robotic experiments that manifest concurrency problems, when present, in quite natural manner. We used these examples during 3rd year Computer Science Engineering degree in Operating Systems topic.

4 Analytical vs Numerical solution of a self-positioning problem

4.1 Objectives

A common problem in robotics is to permit the robot to calculate its current position with respect to a given 2D Cartesian reference using its sensors' data. Powerful robots can perform this calculation with sufficient precision thanks to complex sensors like cameras, lasers or sonars and some landmarks. In NXT the only basic sensor giving a sufficient degree of precision is the sonar sensor able to return its distance from an obstacle within a reasonable range (less than 2.5 m) with a precision of +/- 3 cm.

If the robot knows its distance, namely d_1 and d_2 , from two obstacles, it can be easily shown that the position of the robot is given by one of the two the intersection of the two circumferences centered in each one of the two obstacles and with radius r_1 and r_2 respectively equal to d_1 and d_2 . This analytical solution may be problematic in case of NXT because its run-time allows only integer calculation. This suggests to examine a different approach that calculate the position through subsequent approximations.

4.2 Carrying out the experience

The setup of the experiment includes a tribot with the sonar sensor mounted on the third motor making possible an horizontal exploration, two narrow obstacles put on known positions in front of the robot and a target point (figure 3). Assuming that all other objects (or walls) within the angle of observation are more distant than a minimum, the obstacles are identifiable when the sensor gives distances significantly less than that minimum or simply they are the closest objects in the surrounding world.

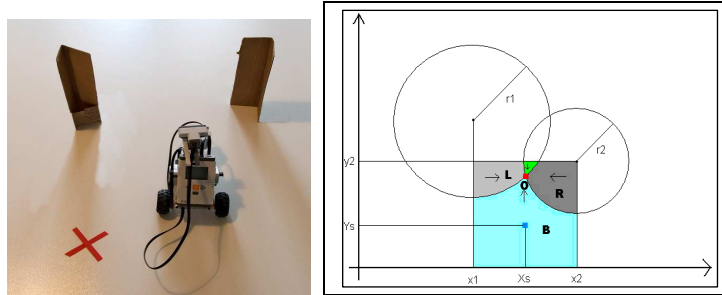


Figure 3 & 4. Scenario for the second example & its “geometric” solution

Given (x_1, y_1) , (x_2, y_2) and (X_r, Y_r) respectively the coordinates of the two obstacles and the unknown coordinates of the robot, and r_1 and r_2 the two distances returned from the sonar sensor, the analytical solution is given by:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 - r_1^2 = 0 \\ (x - x_2)^2 + (y - y_2)^2 - r_2^2 = 0 \end{cases}$$

To simplify the calculation, one of the two equations can be substituted by their difference:

$$Ax + By + C = 0$$

$$\begin{cases} A = -2x_1 + 2x_2 \\ B = -2y_1 + 2y_2 \\ C = x_1^2 - x_2^2 + y_1^2 - y_2^2 - r_1^2 + r_2^2 \end{cases}$$

which is the equation of the so called radical axis, the set of all points equidistant from the two obstacles. We must then calculate the solutions:

$$\begin{aligned} y^2 \left(\frac{B^2}{A^2} + 1 \right) + y \left(2 \frac{BC}{A^2} + 2x_1 \frac{B}{A} - 2y_1 \right) + \\ + \frac{C^2}{A^2} + 2x_1 \frac{C}{A} + x_1^2 + y_1^2 - r_1^2 = 0 \end{aligned}$$

Knowing *a priori* that Y_r is less than $\min(y_1, y_2)$, this allows to choose the correct solution between the two ones calculated from the previous equation X_r is then obtained from Y_r and the axis equation.

The second method starts with a first approximation of the solution given by:

$$\begin{cases} X_s = \frac{x_1 + x_2}{2} \\ Y_s = \frac{\min(y_1, y_2)}{2} \end{cases}$$

The area of interest is divided into four convergence areas denoted in the figure with the letters L (left), O (over), R (right), B (below) that recall the relative position of the approximation (later on called AP) with respect to the final solution (the intersection of the two circles, later on called SOL). The following rules are applied (say $d_1=d(\text{AP}, O_1)$ and $d_2=d(\text{AP}, O_2)$ the distance of AP from respectively obstacles 1 and 2):

- AP is in L if $d_1 < r_1$ and $d_2 > r_2 \Rightarrow$ increase X_s
- AP is in R if $d_1 > r_1$ and $d_2 < r_2 \Rightarrow$ decrease X_s
- AP is in O if $d_1 < r_1$ and $d_2 < r_2 \Rightarrow$ decrease Y_s
- AP is in B if $d_1 > r_1$ and $d_2 > r_2 \Rightarrow$ increase Y_s
- $\text{AP} \equiv \text{SOL}$ if $d_1 = r_1 \pm \epsilon_1$ and $d_2 = r_2 \pm \epsilon_2$

When the calculated distance of AP from the two obstacles coincides to the corresponding radius, apart from a small resolution imprecision (given by ϵ_1 and ϵ_2), AP represents the final solution (implemented in NXC, <http://bricxcc.sourceforge.net/nbc>).

The first part of the program must detect the two obstacles and to measure their distance from the robot exploring the space with the sonar head.

Got the two distances in the r_1 and r_2 variables, the calculation of the analytical solution is straightforward even it presents some difficulties (no floating point computations, no sqrt function available, etc...). For these reasons the terminating condition is evaluated on the square of d_i and r_i previously calculated and avoiding the square root calculation. In fact it results (a similar relation stands also for d_2 and r_2):

$$d_1 = r_1 \pm \epsilon_1 \Rightarrow d_1^2 = r_1^2 + \epsilon_1^2 \pm 2r_1\epsilon_1 \Rightarrow d_1^2 - r_1^2 = \epsilon_1^2 \pm 2r_1\epsilon_1$$

Considering the limitation of the sonar sensor, a value of 1 as the minimum for ϵ_1 (and ϵ_2) is reasonable: when such a value is approached, you obtain: $|d_1^2 - r_1^2| = |1 \pm 2r_1|$

The 'numerical' solution appears a bit simpler and more understandable:

```
// calculate the square of the distances
r1=r1*r1;
r2=r2*r2;
// first approximation
xr=(x1+x2)/2;
if (y1<y2) yr=y1/2;
else yr=y2/2;
// loop to converge to the solution
do {
// square of the approximated distances
// from the two obstacles
d1=(x1-xr)*(x1-xr)+(y1-yr)*(y1-yr);
d2=(x2-xr)*(x2-xr)+(y2-yr)*(y2-yr);
// update the approximation on the basis of
// the area of proximity (see explanation above)
if ((d1<r1) && (d2>r2)) yr=yr-1;
else if ((d1>r1) && (d2>r2)) yr=yr+1;
else if ((d1<=r1) && (d2>=r2)) xr=xr+1;
```

```

else if ((d1>=r1) && (d2<=r2)) xr=xr-1;
// evaluate the approximation
if (d1>=r1)
  conf1=1+2*r1;
else
  conf1=2*r1-1;
if (d2>=r2)
  conf2=1+2*r2;
else
  conf2=2*r2-1;
d1=abs(d1-r1);
d2=abs(d2-r2);
}
while ((d1>conf1) || (d2>conf2));

```

Given the calculated position in X_r and Y_r , the code to reach a target position requires to know the ratio between the angle performed by the motors connected to the wheels and the linear movement of the robot. The rest of the code presumes this knowledge and, apart this important detail, it is straightforward and not presented in detail. An implementation in NXT-G has been also done even though it gives a very large and not so easily understandable program.

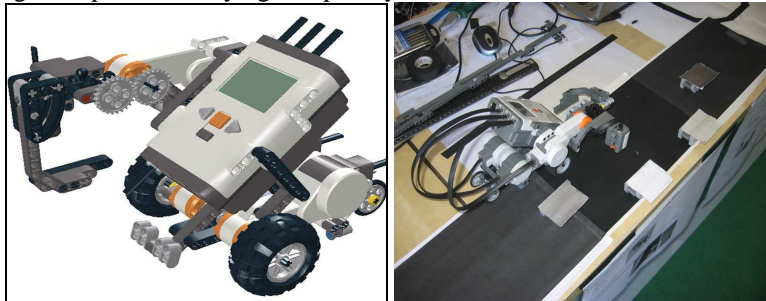
4.3 Analysing the results

NXT is enough powerful to support a rather difficult task like self-positioning, even with evident limitations. The analytical solution requires a knowledge about 2D analytical geometry which is common for an engineer student. The proposed solution shows the differences between the two approaches and makes the students appreciate the suitability of the numerical approach.

5 Sorting

5.1 Objectives

Apart from their practical applications, sorting algorithms are a wide class of interesting examples for studying complexity.



Figures 5 & 6. The special “tribot” used for selection sort & sorting 4 items

We chose two of them, selection and heap sort, as representative respectively of the $O(n^2)$ and $O(n \log n)$ subclasses, because of their relatively simple implementations with NXT. The detailed theory of these algorithms are out of the scope of this presentation and it can be found in any book on fundamentals of data structures and algorithms (for instance in [11]). Moreover the heap sort NXT implementation is still under development, so we limit ourselves to the description of the selection sort implementation.

5.2 Carrying out the experience

For this example the robot is the usual tribot with two motorized wheels, plus a motorized rotating arm used to shift items laterally (fig. 5). Limiting ourselves to the standard sensors included in a kit, we decided to sort objects on their brightness, so we used a light sensor to measure the reflected light of gray colored paper labels glued on the items to be sorted.

One of the initial decisions was to select a physical characteristic we could use to provide values to be compared during the sorting. Limiting ourselves to the standard sensors included in a kit, we decided to use a light sensor to measure the reflected light of gray colored paper labels glued on the items to be sorted. The robot moves back and forth along one of the sides of a black strip on which n items with different gray labels on the top are initially put on predefined positions along a straight line but in a random order. When the robot moves the light sensor, mounted on the robot on the same side of the rotating arm, can read the grey level of each label (fig. 6, with 4 items). The robot makes n passages: during each passage it reads all the n positions looking for the item with the lightest label. When found, it (possibly) comes back to it and activates the rotating arm to shift the item: this action corresponds to the 'selection'. Even if it is not shown, you can imagine that the shifted item drops down on a slide so that, one by one, the sorted items are enqueued in the decreasing order. The black strip has the lowest gray level and therefore the absence of an item previously shifted is recognizable.

5.3 Analysing the results

The more meaningful result of this experiment is the 'live' quadratic behaviour of the robot which makes actually n^2 light readings to complete the task. This can be easily put in relation with the two nested cycles in the code.

```
int i, j, count, n, found, max, read;
task main()
{
  SetSensorLight(IN_1);
  n = 4;
  for (i = 1; i <= n; i++) { // external cycle
    max = 0;
    found = 0;
    for (j=1; j <= n; j++) { // internal cycle
      RotateMotor (OUT_BC, 40, 360); // go forth
      read = Sensor(IN_1);
      if (read > max) {
```



```

// new max
found = j;
max = read;
}}
// back to max
RotateMotor (OUT_BC, 40, ((found-n)*360));
RotateMotor (OUT_A, 40, 360); // select item
// back to start
RotateMotor (OUT_BC, 40, -(found*360));
}}

```

6 A Turing machine

6.1 Objectives

A Turing machine (TM) is a well known computer theory model to study function computability [12]. Formally is a model of computation controlled by a finite state machine equipped with a read/write head on a unbounded sequential tape: depending on the current state and the symbol read on the tape, the machine can change its state, write a new symbol onto the tape, and move the head to the left or right. When for each couple (state, symbol) the specified action is unique, the machine is deterministic (DTM), non-deterministic (NTM) otherwise; due to the theoretical proof of equivalence between a DTM and a NTM, in the following we talk simply to TM referring to DTM. In the proposed experiment, we implemented a didactical TM (with one-direction tape, an alphabet of 2 symbols and 2 possible states) performing integer additions with operands encoded with short bit streams. In our case the necessary limitations are represented by a binary alphabet and a tape with a limited number of slots.

6.2 Carrying out the experience

The read/write head of the simulated TM is a car able to shift LEGO blocks: some blocks are put on predefined positions that represent the limited number of slots of the simulated tape. Each block can be shifted on one of two positions which represent the binary value assigned to the slot; the current position is 'read' using the sonar sensor (fig. 7 and 8).



Figure 7 & 8. The Turing car & the car moving and "writing"

The problem to solve in this experiment with the TM is to perform an add function on integer values. A value i is represented by a sequence of i bit 1, whereas a sum

expression of two values is the concatenation of the two coded value separated by one 0. For instance: 111=3, 1111=5, 111011=3+2.

The rules the TM must apply are summarized in the following table; the initial state is 0 and the slots contain the expression sequence to be evaluated. The number of necessary slots can be estimated in reason of the input expression and the sum value, padding to the right with zeros, at least one terminating zero, the initial sequence if shorter than such a number.

Current state	Input read symbol	Next state	Symbol to be written	Tape (i.e. car) direction
0	0	0	0	>
0	1	1	1	>
1	0	2	1	>
1	1	1	1	>
2	0	3	0	<
2	1	2	1	>
3	0	ERR	--	--
3	1	4	0	>
4	0	END	--	--
4	1	END	--	--

With 7 slots, an input 1110110 is elaborated as follows (underlined the slot under reading, in square brackets the state):

[0] 1110110 – [1] 1110110 – [1] 1110110 – [2] 1111110 – [2] 1111110 – [2] 1111110 – [3] 1111110 – [4] 1111100

6.3 Analysing the results

TM is a very general computation model over which a teacher can deal with a large variety of interesting problems. Its simple definition and elegant power can be appreciated when you see the TM car simulating it. Our implementation can be easily modified to study and implement different resolving algorithms.

7 Conclusions

Using different approaches for programming the robot, it is possible to introduce in an easy way advanced programming skills and motivate the students to examine and exploit complex models and programming paradigms.

Topics and experiences presented in this paper were related to computer science at university levels, but robots as “learning tools” can be exploited by also teachers from different disciplines and from previous education levels, as demonstrated by other examples developed in the TERECOP project framework. Guiding examples must be used as suggestions to teachers to prepare their own experiences taking into account their specific didactical objectives, the initial competence of their classrooms, the

operative environment. In any case, it is important to use adequate methodologies, to coordinate/integrate the activities within the curricula and with the other colleagues. In the next months we have to deal with the practical and organization issues to apply these issues at high school and university levels.

Acknowledgments. This paper was partly based on work done in the frame of the project “Teacher Educations on Robotics-Enhanced Constructivist Pedagogical methods” (TERECOP) funded by The European programme Socrates/Comenius/Action 2.1, Agreement N° 128959-CP-1-2006-GR-COMENIUS-C21 2006-2518/001-001 S02.

References

1. Alimisis D. et al, 2007. Robotics & Constructivism in Education: the TERECOP project. In Proceedings of the 11th European Logo Conference (<http://www.eurologo2007.org/>, 19 - 24 August, Bratislava, Slovakia)
2. Arlegui J. et al, 2007. Los entornos LEGO y LOGO en robótica educativa. In Proceedings of EDUTEC 2007 (Latin American conference on Education & Technology, <http://www.utn.edu.ar/edutec2007>), October 2007, Buenos Aires (in Spanish)
3. Enrico Pagello, Emanuele Menegatti, Daniel Polani, Ansgar Bredendel, Paulo Costa, Thomas Christaller, Adam Jacoff, Martin Riedmiller, Alessandro Saffiotti, Elizabeth Sklar, Takashi Tomoichi RoboCup-2003: New Scientific and Technical Advances AI Magazine, American Association for Artificial Intelligence (AAAI) Volume 25, Num. 2, Summer 2004, pp. 81-98
4. E. Menegatti, A. Pretto, A. Scarpa, A. Tellatin, S. Tonello, A. Lastra, A. Guatti An interactive robotic sculpture EUROBOT 2006 Workshop on Educational Robotics, Acireale (ITALY), June 2006
5. A. Lastra, P. Alberti, E. Menegatti Experimentations on advanced robotics for academic education EUROBOT 2006 Workshop on Educational Robotics, Acireale (ITALY), June 2006
6. Computing Curricula: 2005 Overview Report (The Joint Task Force for Computing Curricula 2005, http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf), ACM, AIS & IEEE-CS.
7. Peter J. Denning and Andrew McGettrick. Recentering Computer Science. Communications of the ACM, November 2005/Vol. 48, No. 11.
8. Harel, I. and Papert S., Constructionism, Learning by Design, and Project-based Learning, 2001. In M. Orey (Ed.), Emerging perspectives on learning, teaching, and technology. Available Website: <http://www.coe.uga.edu/epltt/LearningbyDesign.htm>
9. Loidl S. et al : Preparatory Knowledge: Propaedeutic in Informatics, in Proceeding of ISSEP 2005: pp. 104-115, Klagenfurt, Austria.
10. Alimisis P.: Designing Robotics-Enhanced Constructivist Training for Science and Technology Teachers: the TERECOP Project, Proceedings of ED-MEDIA 2008-World Conference on Educational Multimedia, Hypermedia & Telecommunications, p. 288-293, Vienna, Austria.
11. Goodrich M. T., Tamassia R., Data Structures and Algorithms in Java, 4th edition, Wiley & Sons, 2006.
12. J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, 3d Ed., Addison-Wesley, 2007.

The use of LEGO Mindstorms in elementary and secondary education: game as a way of triggering learning

Soumela Atmatzidou, Iraklis Markelis, Stavros Demetriadis

Computer Science department, Aristotle University of Thessaloniki, Greece
{atmatzid, markelis, sdemetri}@csd.auth.gr

Abstract. In this work we present a didactic approach that investigates the effectiveness of using the Lego Mindstorms robots as tools for introducing students to basic concepts of programming through game play activity. Our approach comprises collaborative and entertaining features and emphasizes the element of competition between student groups in elementary and secondary education. Overall, the paper provides research evidence that approaching learning as an entertaining activity, through the use of LM robots and the spirit of team competition, offers a pleasant, creative and effective method of instruction for the acquisition of introductory programming knowledge.

Keywords: *LEGO mindstorms, edutainment, learning through play, competition, constructionism*

1 Introduction

The educational robots of the Lego company (Lego Mindstorms, henceforth called “LM” <http://mindstorms.lego.com/>) have been systematically used for the introduction of novice students to learning programming [8], [11], [4], [2]. The design philosophy of the Lego instructional material is based on the concept that kids should not only construct the knowledge by themselves, and specifically on the thought that learning is established through play (“*learning through play*”) [8], [11]. This opinion has its roots to the approach of Constructionism [16] according to which learning through play can contribute to the construction of new knowledge which is based on the students pre-existing knowledge. As the kids work on subjects meaningful to them, they are motivated [9] and they act as real scientists or inventors by having a more direct contact with the concepts underlying the domain. Therefore, the goal of the use of LM is the integration of play into the educational procedure by offering to students the opportunity to be entertained and develop their imagination.

In this work we present an effort to use the LM robots for introducing students of elementary and higher secondary schools to issues of programming. Our approach comprises collaborative and entertaining features and emphasizes the element of competition between the student groups.

Specifically, what is being studied is the degree to which the use of LM can help a play activity to a) reinforce the interest of students to be creatively, pleasantly and effectively engaged in programming activity and b) to help them transfer their programming knowledge from the environment of LM to more typical programming environments (e.g. Visual Basic).

In the following section, the theoretical framework of this educational approach is presented together with a brief review on the LM robot (the hardware and the software that come along with it). A presentation of the lesson-training program with the use of LM follows and finally a brief description of the hands-on experience and the first survey results are presented.

2 Lego Mindstorms & edutainment

2.1 Lego Mindstorms

LM is a rather new Lego product (first out in the market in 1998) which belongs to the so called “3rd generation kit” category (<http://mindstorms.lego.com/>). It is about an easily programmable robot which is accompanied by a great variety of bricks, motors, sensors and other equipment which help in building actual models. These robots can be programmed, in order to execute orders and react to different stimuli received through their sensors, by using the proper environments of programming development.

In issues of introductory programming, the use of robots is expected to have positive impact, since it can help – among others – towards the understanding of an accurate and logical machine instructional language [18]. LMs are used as a tool for teaching problem solving methods, being a very pleasant and interesting past-time, offering at the same time a simple and educational interface. Students see them more as a game rather than educational tools since the majority of the kids have played with Lego bricks in the past. The game part is a very important factor promoting and motivating students to learn [20].

However, studies focused on the use of robots for learning programming concepts are inconclusive as regards the emerging learning benefits [5], [14]. Furthermore, as mentioned above, the use of robots limits the instruction of advanced programming concepts such as that of object oriented programming [14]. There is though a number of some research projects which claim that robots helped significantly in the impartment of basic programming concepts [17], [2]. A study with high school students' has reported positive results concerning the student's class interest during their lessons as well as the accomplishment of their educational goal [4].

2.2 Edutainment

The term edutainment means an educational approach combining games and learning. The general concept of edutainment is related to almost every game which has an educational role. Its goal is to turn education into a fun activity, since it is widely known that learning is more easily, more substantially and more quickly achieved when combined with playing [12]. Edutainment is about activities through which students interact with a computer or other artifacts such as robots aiming at winning a prize or creating something that gives them moral satisfaction. This experience helps them broaden their knowledge and at the same time practically integrate the terms that he has been taught in different subjects.

Over the last years many scientists have been studying the impact of using LMs in education, adopting the ideas of Constructivism and Edutainment [4], [1]. Researchers dealing with taking up the Edutainment method have come to encouraging results [3], [12]. Chandana, Hafner και Bongard (2000) claim that students have not only learned to comprehend the terms of every lesson but also, most importantly, have integrated them into their own knowledge structures as tools and constructive material which could have a future use. Moreover, researchers report that the only negative factor of their lessons, according to the students, is that “they should have lasted longer”.

One of the difficulties that students face when dealing with a problem by using a programming environment is the use of representations required to be constructed during the problem solving process [19]. The comprehension of data processing operations being executed by the computer is of great importance to the student [20]. In addition, the development of necessary mental models is very important, especially during the use of programming environments where the transfer from 'objects of the world' to 'informative objects' is required [6]. However, the usual introductory problems to issues of programming do not challenge students' interest because they deal with the processing of numbers and symbols [22]. We suggest that difficulties such as the above can be overcome with the proper approach, adopting game as a way of triggering learning.

Another important issue in the framework of playing games is of course the competition among individuals and/or teams. The majority of related studies suggest collaborative and not “competitive” learning [21]. However, a study analyzing the consequences of competition in teaching informatics underlines that this kind of circumstances can promote learning only if the teacher uses competition efficiently, i.e. turning it into a strong motive for engaging children in the subject of robotics and programming [13]. In particular, students participating in such activities managed to improve greatly in terms of their grade performance [15].

Bearing in mind the previous research results, we suggest that promoting controlled competition among teams participating in “trainings” with the aim of success in a final “challenge”, can result in a multiply efficient learning experience for the introduction to programming issues. Next we present the way in which we designed such a competition-based edutainment activity and our preliminary research results..

3 Design of a competition-based edutainment activity

Through personal experience in teaching lessons of the basic principles of programming, we have noticed that students encounter problems concerning the comprehension of basic concepts, such as variables, conditions, the loop structure etc., when the problems that need to be solved do not capture their interest. This point of view is supported by related studies [19] which claim that the use of robots can build an environment in which the students' interest in solving problems can be notably high, resulting in better learning outcomes.

The two main problems that we had to deal with, during the design of the robot lessons were how children would better understand (a) the loop and control structure and (b) ways of using the robot sensors through the programming environment. To be more specific, students had to understand the relation between the execution of iteration (loop) or conditional commands and the existence of events (e.g. execute a series of commands until the touch sensor is pressed). Furthermore, programming a robot through the use of sensors was an unknown experience for the students. That was a fact that we had to consider if we wanted the kids to be able to complete the final stage of the lessons.

After reviewing the available programming environments for the RCX programmable brick, we discovered that there are many languages that we could use as a teaching tool. Each one of those serves a different teaching purpose [7]. In this work the programming environment which comes with the Robotics Invention System 2.0 (RIS) was chosen. This tool is designed for kids, it only requires basic knowledge on the use of computers without expecting from anyone to have any experience on programming principles [10]. Furthermore, the way the environment RIS represents the program's commands is very similar to the logic of developing flow diagrams.

Taking into account the Lego company's directive instructions for the way the lessons are to be carried out (Constructopedia), we developed a series of lessons which we named "trainings" and a final activity between the teams which we called "challenge". The students knew from the beginning that in the challenge phase their team would have to successfully complete a specific activity, the winning team being the one to accomplish it in the best way.

The "class" consisted of two teams of three students each. During the trainings the students were supported to gradually understand the robot's programming technique. At the same time, they were encouraged to experiment, observe and record the effect that the value changes of the program input parameters would have on the robot's way of functioning. The main goal was for the students to become familiar with the robot programming techniques so that they would get properly prepared for the implementation of the "challenge" activity. Moreover, during the trainings as well as during occasional breaks the contestants were discussing, exchanging opinions on possible scenarios that the teams could implement by themselves.

The learning environment's basic characteristic was a communication model which allowed the participants to interact within conditions of "controlled" competition. During the lessons, the trainers tried to convey to the kids the message that:

- equally dividing tasks among all team members,
- working towards the goal's achievement simultaneously,

- communicating effectively with one another,
- acting with the proper behavior and
- maintaining the spirit of fair play,

are the elements that would help the teams achieve their goal faster and more successfully [13].

Also, the goals that had to be achieved, within the final challenging activity, were made clear and both of the teams were instructed that after all they would be winners by actively competing in the above educational procedure [13].

The “trainings” general structure was as follows:

1st Training: LM robots are presented. Afterwards, the student teams are formed and each one decides on their name. A functions demonstration follows, of the two robots, constructed by the instructors. At the same time, guideline sheets are given to the students, in which the following are described: a) the lessons time schedule b) a plan for each one of the lessons c) a short introduction to Lego Mindstorms, the RIS environment and the encyclopedia named “Constructopedia”.

2nd Training: The instructors underline the importance of teamwork and cooperation, reinforcing the spirit of fair play among students. Next, the instructor assigns a day's project to both of the teams and hands out supportive material in digitized or printed form. The students construct their own robot by following the step-by-step instructions and finally they execute their first built programs (motors usage). At the same time, trainers approach the problems occurring among the team members and use them to give feedback to the rest of the class, promoting in this way cooperation among the kids.

3rd - 4th Training: Includes the use of basic input-output commands, using the touch sensor. An introduction of the basic programming structures (sequential and conditional structures), is made. During the training, the students use ready-made blocks, experimenting by changing the values of various parameters, creating new blocks of orders.

5th - 6th Training: Includes the use of repetition structure commands by using touch and light sensors. During the training, ready built repetition blocks are applied for the implementation of more complex activities compared to the ones completed during the previous trainings. New programming terms, such as that of the counter, are presented at the same time. Having the previous experience of the trainings they had competed in, both of the teams try to develop their programs in the best possible way (speed and efficiency of execution). For an easier comprehension of the repetition orders, the first activity is carried out with the help of the instructors.

Challenge Phase: The instructors present the final challenge and give detailed orders to the teams. The students receive a brochure with detailed steps about the scenario which they have to implement but also about the way they are graded. The description

of the scenario, which represents the course that the robots have to follow, appears in this brochure in text form as well as in diagram form.

4 Implementation and Results

LM robots were used within the didactic approach framework described earlier, aiming at teaching basic programming concepts to students of the 5th and 6th year of an Elementary school (aged 11-12) at the city of Serres and the final year of a Technical High school (aged 17-18) at the city of Kozani. For each of the above cases two groups of three persons each were formed. Both Elementary and High school students used the same training material and were guided according to their needs.

The didactic application was separated in two phases: a) the “Training” phase and b) the “Challenge” phase.

The training phase lasted for six sessions and the kids were prepared for the final test-challenge. Realistic queries-problems were given to the students during the training, for example: “If the robot collides with an obstacle what should be done so that it continues its route?”.

During the “challenge” phase, which lasted for two sessions, the final test was assigned to the groups and they had to bring it to an end based on the knowledge acquired during the training phase. Finally, worksheets, implemented programs and photographic material from the sessions are included within the data collected.

A qualitative type of methodology was applied in our research, which had as follows: During the implementation of our didactic approach we created an activity log with the comments and the observations of our students as well as our personal ones. What the students were thinking as well as their views on their experience was recorded through semi-structured interviews.

After collecting and grouping the research data, the following results were extracted:

- The engagement of children with LM robots, within the course they participated, contributed to their familiarization with structured programming principles, a fact that had a positive influence on developing problem solving skills. We observed that they understood more easily programming concepts (e.g. counter, flag, repetition, etc) which they had difficulties to realize and apply during the Computer Programming courses (Pascal, Visual Basic). A characteristic quote from a student: “..I understand better a repetition structure when it is to make the robot hit an obstacle three times and then stop. It is interesting like this.. ”
- Using robots, the programming concepts acquire meaning for the students due to the direct feedback which exists between the algorithm and its implementation.
- The children demonstrated a tendency to outdo the opponent, more specifically tried to think of ways to undermine the operation of the robot of the other team. A characteristic question by student: “.. Could we send a erroneous command to the

other team's robot?..". In that case, the role of the trainer was very important because not only the knowledge of how to intervene on the other robot's operation should be given to the students but, at the same time, the importance of fair play should be noted, cultivating this spirit among them.

- The observations and the reports of the students during the programming lessons within the framework of their studies (learning Visual Basic) were very important. It was noted that on teaching new commands, students related them to the relevant activities on the robots and this helped their better and easier understanding of programming commands such as If, For or While. A characteristic quote from a student: "I never thought Visual Basic could be so interesting. Could we use it to program the Lego robots?"
- From discussions, interviews and comments by the children it became obvious that competition between the two groups during the final challenge was the motivation that kept the interest of the students undiminished and helped surpass any difficulties. Additionally, it greatly increased the desire of the students for engagement with programming.
- The game's aspect which is embedded in programmable robots prompted children to be more creative, facing robot programming as an entertaining and easy occupation. The children's enthusiasm was obvious in their comments: "Why don't we use them at lessons?", "I would like to have one at home. How can I buy it?", "Can we play with the robots afterwards?".

5 Conclusions and future research

This paper presents the experience of an educational activity in the form of a competition-based game, aiming at introducing the students to issues of computer programming. It also provides preliminary research evidence that approaching learning as an entertaining activity, through the use of LM robots and the spirit of team competition, offers a pleasant, creative and efficient method of instruction for the acquisition of introductory programming knowledge.

The enrichment of the lessons by introducing new material in order to evaluate in more detail the level of the knowledge obtained by the students with the use of this specific educational tool-artifact is within our future aims. Another issue for consideration is determining the most appropriate duration of the training sessions since it was shown to be an important factor during the activities the results of which are presented in this paper.

Another interesting subject that we are planning to deal with is the use of the new Lego brick known as NXT (Next). The Lego's new creation has been considerably improved as far as communication devices and autonomy abilities are concerned comparing to its predecessor RCX. Moreover, NXT is accompanied by a great variety of sensors the use of which makes possible the creation of a larger amount of more

complex activities. Finally, the recently created program development environments for the NXT brick facilitates the easier use of robots and also the understanding of more complex programming concepts such as the subroutines.

References

1. Asada, M., D'Andrea, R., Birk, A., Kitano, H., & Veloso, M., (2000). Robotics in Edutainment. Proceedings of the 2000 IEEE International Conference on Robotics and Automation, San Francisco, 795-800.
2. Beisser, S. R. (Jan 2006). An Examination of Gender Differences in Elementary Constructionist Classrooms Using Lego/Logo Instruction. *Computers in the Schools*, 22, 7-19.
3. Chandana, P., Hafner, V., & Bongard, J. (2000). Teaching new artificial intelligent using constructionist edutainment robots.
4. Dagdilelis, V., Sartatzemi, M. & Kagani, K. (2005). Teaching (with) Robots in Secondary Schools: some new and not-so-new Pedagogical problems. In ICALT'05 - Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies.
5. Daniel, C., & Cliburn, D. C. (2006). Experiences with LEGO MINDSTORMS throughout the Undergraduate Computer Science Curriculum. CA 36th ASEE/IEEE Frontiers in Education Conference, San Diego, 1-6.
6. Green, T., Hoc, J. M., Samurcay, R., & Gilmore, D. (1990), *Psychology of programming*, San Diego: Academic Press.
7. Hirst, A. J., Johnson, J., Petre, M., Price, B. A., & Richards, M. (2003). What is the best programming environment/language for teaching robotics using Lego Mindstorms?. *Artif Life Robotics*, 7, 124-131.
8. Hussain, S., Lindh, J., & Shukur, G. (2006). The effect of LEGO Training on Pupils' School Performance in Mathematics, Problem Solving Ability and Attitude: Swedish Data. *Educational Technology & Society*, 9(3), 182-194.
9. Jarvinen, E. & Hiltunen, J. (2000), Automation technology in elementary technology education, *Journal of industrial teacher education*, 37(4), 51-76.
10. Kaskalis, H. T., Dagdilelis, V., Evangelidis, G., & Margaritis, G. K. (2001). "Implementing Applications on Small Robots for Educational Purposes: Programming the LEGO Mindstorms," Proceedings of the 5th Hellenic – European Conference on Computer Mathematics & its Applications (HERCMA 2001), 337–341.
11. LEGO Dacta A/S. Study of Educational Impact of the LEGO Dacta Materials - INFOESCUELA – MED. Retrieved June, 15, 2007 from <http://www.lego.com/education/download/infoescuella.pdf>
12. Lund, H. H., & Nielsen, J. (2002). An Edutainment Robotics Survey. In Proceedings of the Third International Symposium on Human and Artificial Intelligence Systems: The Dynamic Systems Approach for Embodiment and Sociality, Fukui.
13. Marlow, E. (2001). Cooperative Learning versus Competition: Which Is Better?. U.S Department of Education, Opinion Papers, 1-12.
14. McNally, M., Goldweber, M., Fgin, B., & Klassner, F. (2006). Do Lego Mindstorms Robots have Future in CS Education?. *ACM SIGCSE Bulletin*, Volume 38, Issue 1, 61-62.
15. Noguez J., Huesca L. G., & Sucar E. (2007). Shared Learning Experiences In A Contest Environment Within A Mobile Robotics Virtual Laboratory.
16. Papert S. (1993). *Mindstorms: Children, Computers, and Powerful Ideas*, Second edition by Basic Books.

17. Karantarou, A., Tahos, N., Aliisis, D. (2005). Introduction to the basic principles and structures of programming using LEGO Mindstorms robot constructions. Proceedings of the 3rd Pan-Hellenic conference "Informatics didactics", University of Peloponnisos, Korinthos.
18. Komis, B. (2004). Introduction to educational software of informatics and communication technologies, Athens, New Technologies
19. Komis, B. (2005). Introduction in informatics didactics, Athens, Klidarithmos.
20. Komis, B. (2005). Pedagogical computer activities for Pre-school and first school age. University publications, second edition, Patra.
21. Solomonidou, X. (2002). Cooperative learning using informatics technologies in education: Experiences from elementary schools of Thessaly. 3rd ETPE conference proceedings, Aegean university, Rodos, Kastaniotis publications.
22. Ksinogalos, S. (2003). Programming teaching scenarios in secondary schools. In M. Iosifidou & N. Tzimopoulos, 2nd Pan-Hellenic informatics education technologies instructional personnel conference proceedings 'Use of informatics and communication technologies practicing in education', volume A', p. 783-795, New technologies publications.

Sketch for a Scientific Foundation for Constructionism: with a note of some difficulties

Mike Doyle

mikedoyle@logios.org

Abstract. TERECOP situates itself within the constructionist philosophy of pedagogy. This paper outlines a thesis that seeks to place constructionism in a Neo-Darwinian scientific framework. It is suggested that our capability to construct is a successful, data processing, evolutionary adaptation that is unique to our species. Instructionism, conversely, it is suggested, is associated with ancestral adaptations for group living, specifically language and memory. Reaction to certain ICT developments is considered from the viewpoint of this two-adaptation model. It is concluded that the model does offer the potential both to provide a scientific foundation for the constructionist approach and also offer a possible explanation of the tenacity of the instructionist approach.

Keywords: technology, constructionism, instructionism, language, evolution

1 Introduction

The TERECOP robotics education project of the EU Comenius Programme situates itself within a theoretical approach based on Piaget's constructivism as enhanced by Papert's constructionism set in a Vygotskian social context [1]. The authors noted that, a quarter of a century of ICT-based robotics notwithstanding, robotics had made few inroads into the school curriculum. As with mainstream ICT, robotics remains a tool for supporting other ends, such as science, mathematics and teamwork.

They contrast the constructionist approach with instructionist pedagogy. This latter is normally language and memory based, and focuses on the 'value' teachers add to pupils; e.g. see the English education system [2]. Both language and memory have been subject to extensive study. Construction is relatively little researched and poorly understood. Indeed, the academic default is that language is the catalyst of creativity: our lack of understanding of how we 'do' technology and science notwithstanding.

1.1 Two Cultures

Robotics, in common with science and mathematics, suffers from the Two Cultures [3] syndrome: an educated individual is expected to converse on Shakespeare but not on entropy. There is tension between word and graphic as information carriers [4]. In school, language has higher status than engineering. This suggests that the reluctance

of education to incorporate robotics into the curriculum has a cause deeper than a lack of suitable training materials.

The aim of this paper is to give constructionism a firmer foundation by subjecting the constructionist/instructionist dichotomy to evolutionary, Darwinian, scrutiny. It is hoped this will begin to explain the relatively unregarded status of engineering in a technologically dependent culture controlled by an academic-administrative elite.

2 It's not there! What's not there?

By standing on the shoulders of giants, we see further through their eyes: but we find great difficulty in thinking their unthought thoughts. We tend to clone the unknown from existing knowledge:

Something puzzling happened 50,000 to 100,000 years ago. The fossil evidence is patchy, but it seems that hominids suddenly developed brains that, in terms of size, were very much like ours. Yet this apparent growth spurt was not reflected immediately by any great cultural changes. That came 50,000 years later, when a whole variety of artefacts – tools and musical instruments and cave drawings – suddenly came on the scene.

Something must have happened between the physical changes in the brain and the cultural expression of such changes. Most linguists now agree that the something was the development of language. I am sure that our ancestors had been communicating for a long time (half a million years or so) before they became linguistically competent, so perhaps there is something in language itself that led to this acceleration of cultural complexity.

Or could it have been the other way round? Could cultural changes have brought about the development of language?

The importance of gossip, Maynard Smith [24:257]

2.2 Incremental change

Two words: *suddenly* and *language*, above, raise interesting questions.

- At speciation, the suite of characteristics that later distinguishes a species will only have reached the stage where interbreeding with the root population ceases to be desirable. There is no guarantee, indeed it is unlikely, that the incremental gene-driven process of phenotypic change will be complete.
- Technology, the creator of culture, does not magically appear. It progresses in precision and complexity over the generations, constrained by the amount of mental work and physical resource that are made available, i.e. by limitations enshrined in the second principle of thermodynamics.
- A neural mechanism to link language to technology is required. No such link is demonstrated, or even proposed. It is presumed.

2.3 The missing knowledge

Introductory texts on psychology, e.g. Atkinson & Hilgard [5]: have a chapter on language, on perception, on social relationships, but nothing about our ability to draw and to construct. So we have no scientific foundation upon which to build pedagogy.

- We are ignorant of how human beings do technology.
- We do not know how children begin to able to draw.
- We remain prisoners of philosophies woven with words.

The cultural default is: because we have language we can make things – or vice-versa.

3 On being human

Over the centuries, various attributes have been used to define our uniqueness relative to all other species. Tool-use was an early identifier – but other species use tools and construct entities: the tools used by chimpanzees have been reckoned commensurate with the tool-kit of Tasmanians [6]; and termite colonies and the nests of birds are complex built artefacts. The current favoured identifier is language [7 8 9 10]. Again, other species, particularly primates, appear to have language capacity: bonobos have learned symbol systems. Similarly, the life-style of humanity has many ape parallels [11]. On the other hand, an attempt to teach a language-using bonobo to draw a line between two dots failed [12]. So, drawing might just be a valid index of difference.

3.1 Human evolution

The Homo lineage stretches back to our divergence from the chimpanzees over five million years ago. Homo sapiens sapiens is about 150,000 years old. Species related to our lineage are: Homo habilis, associated with simple stone tools; Homo erectus, with a sophisticated tool assemblage (including the characteristic bifacial ‘hand-axe’), who used fire and ranged across Africa and Asia; Homo neanderthalensis ranged cold Europe whilst anatomically modern humans inhabited warmer Africa, meeting in the Levant 100,000 years ago and coexisting in ice-age Europe until around 30,000 years ago [13]. Both had similar toolkits. Both had brains as large as, or larger, than ours.

3.2 Speech

The first indicators of speech are seen in the first member of this lineage, around two million years ago. Speech anatomy is more pronounced in H erectus a million years later. The Neanderthals and their more slender African contemporaries had the full suite of anatomical modifications found in modern humans. This suggests that their common ancestor spoke articulately around half a million years ago. Hence, we are not the first species to speak. Indeed, language (Saussure’s parole) may have been a highly developed evolutionary adaptation well before our speciation event occurred.

Why speak? Why might language, once evolved, be adaptive? The answer appears to lie in our highly unusual social lifestyle.

Lifestyle. The human lifestyle is biologically unusual and intrinsically unstable. The genetic Darwinian model [14 15] fits most lifestyles including parasitism; symbiosis; daughters forgoing reproduction to help raise siblings; and herd living. However, it neither explains human capacity for large group social living, cooperating with genetically unrelated people, and designing and making artefacts, nor cooperation between genes in cell nuclei. Evolutionary psychology [16] offers an explanation.

Reciprocal altruism. If I have an overabundance of resources at the present, it pays me to share the excess with you, provided that I can be sure that you will reciprocate when the situation is reversed. Similarly, it pays you to honour your contract with me. A reciprocally altruistic lifestyle is an evolutionary stable strategy for the individual in an economic climate of unevenly distributed resources. In adaptation terms, there is a mutual increase in the likelihood of grandchildren of reproductive age. Prisoner's Dilemma, a game theory model, demonstrates how a small guaranteed mutual gain is assured by working together; but that a defector can scoop the jackpot on any given occasion, so non-cooperation is always a tempting option. The reciprocally altruistic lifestyle, put simply, is trade. Our lifestyle is characterised by trade and negotiation.

Making it work. Reciprocity needs work. In a naïvely cooperative population, the (inevitable) evolution of non-co-operators soon results in the co-operators becoming extinct. For reciprocal altruism to be evolutionarily stable, the cooperators need a means of controlling non-co-operative behaviour (defection or freeloading). For sentient organisms, like us, there are three prerequisites:

1. the ability to recognise oneself and other individuals;
2. a good memory for past events; and
3. a mechanism for sanctioning defectors.

The first two are functions of the nervous system that emerged with higher primates.

Defection control. A 'tit-for-tat' algorithm [17] is sufficient to assure reciprocity. The rule is: Cooperate on first meeting; thereafter reciprocate only if the other does. When implemented in a (computer) model, non-co-operators decline to a small proportion of the population – commensurate with the 'cooperate on first meeting' loophole.

Negotiation. In a population of real people, the first two prerequisites turn this simple algorithm into an affective mechanism of great subtlety. It is necessary to remember what favours you did to whom and when, and vice-versa. In the complex lifestyle of higher primates, including our Homo predecessors, negotiation and re-assurance of fidelity go hand in hand. We see the naissance of this in the grooming behaviour of primates. Dunbar [18] argues that the purpose of language is gossip. Anthropologists and linguists find that language is used precisely for such purposes. It is not used for technical discussions [19]. The most elegant and economical explanation of language evolution is in the pivotal role that negotiation has in a reciprocally altruistic society.

Language diversity. Thus, it seems that speech evolved as a facilitative mechanism for a reciprocally altruistic lifestyle. Language diversity [20] – the panoply of lexical, grammatical, phonetic, prosodic, pragmatic, and personal characteristics – may be seen as the outcome of an 'arms race', where speech is variously used to: differentiate

between and integrate within tribal groups, detect non-co-operators, test for reliability, cement interpersonal relationships, persuade, gain confidence, cheat and exploit.

3.3 Whence creativity?

It is difficult to see how our capacity for creativity could have evolved from this language capability. There is nothing within language that suggests any connection to technology. The adaptation itself is entirely contained within the phenotype, which is modified to facilitate language use. Neurologically, language is primarily located in areas of the brain that evolved well before our prefrontal cortex expansion. All languages in all cultures are equally powerful and expressive, unlike the technologies of different cultures. The importance of language to our lifestyle, its story-telling complexity entailing intensions, time, place and events, plus its capacity for infinite combination, is compelling, at face value: but speech has no precursor for technology.

Extended phenotype. Let me be very precise about what a precursor of technology might be. The extended phenotype [15] goes beyond genetic phenotypic hijacking and symbiosis: parts of the material world are also annexed. The phenotype is 'extended' into the environment to the advantage of the organism. A cadis-fly larva's house is made from grains of sand; a wasp's paper comb and birds' nests use environmentally available material. But these artefacts are no less an evolved adaptation than are the webbed feet of a duck. Some primates, notably chimpanzees, do exhibit to a small extent the learned use of tools, such as hammer and anvil stones to crack nuts. Yet, a language-using bonobo was unable to learn to draw a line between two dots.

Technology. We require a precursor to the behaviour of a species that, in its 150K year existence, progressed to study the origins of the universe and its own psyche. A technological precursor would be an extension of the chimpanzee learned-tool-use to a level where design and development are seen. We seek a veritable phase-transition. When the tool assemblages of all prior species, including the Neanderthals, are examined, the most notable characteristic is their stability. The bi-facial hand-axe of *H erectus* remained unchanged throughout its range for over a million years. The most notable change in stone tool construction was the use of small geometric components. The earliest date given for artefacts that show evidence of design and component-based construction is about 250,000 years ago [21]. This is a quarter of a million years after full speech anatomy development, and approaches our point of speciation.

Geometrics. We require an evolutionary process that led to a speciation event and which accelerated once the gene pool ceased to be diluted by interbreeding with the extant population. As this workshop revolves around products of the LOGO® Group, let us consider a brick. The LEGO brick is a cuboid with cylinders atop designed for assembly. This links back to ancestral geometric flints and component built tools. But, whence came the geometric forms? They are hardly present in the organic world.

What evolved? Geometric forms, the Platonic view aside, can only originate within our brains. The question is: How? The answer must lie with neurological changes that

took place, and may still be taking place [22 23], in our cortex. The prefrontal cortex of human beings is over a quarter of the whole brain and is massively connected to other parts of the brain, including the oldest [8 24]. It is where planning, personality and consciousness reside. Activity levels here are associated with mental conditions such as autism, schizophrenia, bipolar disorder, and artistic flair [25]. The aspect that I wish to focus upon is none of these. It is more ‘What’s in a Square?’ [26].

Square-diamond. The square is a fascinating shape. Cubed, it builds. It changes its name when rotated. This last is rather odd, because we have object-constancy firmly built into our perceptual system: My cat remains recognisably my cat from whichever viewpoint I see him, or part of him. A square, on the other hand, rotated by a quarter turn, becomes a diamond. This effect pops up in the mathematics classroom, where children have difficulty accepting that a square pointy way up is the same as a square on edge. Adults, when shown each orientation in isolation will name them differently, sometimes consciously correcting themselves when they recall their school geometry.



If we were able to explain why object constancy breaks down in this case, we might take a first step on the path to understanding technology: because the capability to break down the whole into parts is prerequisite for component-construction.

Data? The source of data within the brain on features of objects such as: colour, line, tone, etc. is the mammalian cortex. For instance, the visual cortex has neurones that specialise in processing lines of varied angle [27]. Those handling diagonal lines are different from those dealing with the horizontal and vertical. Let ‘geometry’ be a prefrontal cortex creation, sourcing data from the visual system. The two orientations are derived from different data, so ‘are’ (and are named as) different objects.

Our adaptation? Is it feasible that the prefrontal cortex might be parasitic on the rest of the brain? Neurones that do not receive input die. Many are pruned in normal development. However, it is in the ‘selfish interest’ of the neurone gene to multiply its representation in the community of cells that make the phenotype. The phenotype will only accept a greater proportion of a specific cell type if adaptive advantage ensues. A known role of prefrontal neurones is to analyse the world and reconstruct it explicitly. (E.g. people with Asperger’s syndrome use a part of the prefrontal cortex to analyse and reconstruct the rules of social behaviour that come so naturally to others.) There is no direct evidence that the brain pre-frontally constructs novel mental entities from internal data. The circumstantial evidence, however, is significant. We do isolate the atoms of shape, colour, and sound. We do reconstruct the world on a massive scale, as Heidegger [28] observed with concern. Agriculture, clothing, and housing (setting art, science and writing aside) are a sufficient demonstration of the adaptive advantage

that the capability to construct has given us. And, art (colour and line), music (sounds and sequence), dance (movements and space) and syntonicity (Logo) offer support.

4 Instructionism

Given that the human brain works as proposed, then our natural learning style will be constructive. Children's capacity to draw and construct starts to develop in school from the age of four and continues to about the age of ten, coinciding with the onset of pre-puberty. (Our constructive prefrontal lobes continue to develop well beyond our mid-twenties.) During this phase, when the prefrontal cortex begins to be active, children become highly imaginative, construct impossible worlds, and create fictitious friends. Why, then, is verbal instruction and rote learning so valued? The answer is: language. Language and memory underpin our lifestyle and society; language evolved for trade and gossip before we became a distinct species. This communication currency emerges before our constructive capacity begins to develop. It is established by the age of four. Language is available when children start school learning. Given the emphasis in society on early literacy and numeracy, and the susceptibility of number and reading to language-based method, it is unsurprisingly the default option.

4.1 Talk, memory and threatening technology

The pedagogue's craft uses the story and spoken sums. The fairy-story ties emerging imagination to a child's gossip capability. Mental arithmetic rides on the memory that supports reciprocal altruism. It would seem natural to base early education on the fully developed language capability rather than on the emerging constructive capacity. However, it may be argued that by anchoring early education to language there is a danger that children will be deflected from developing humanity's unique and more recently evolved capabilities. Unfortunately, education is under philosophical rather than scientific control. And language is the tool of philosophers and politicians. So, it should not be surprising that attempts to constrain the use of language and increase the constructive content of schooling have stalled. Comenius [29], Montessori [30], Feuerstein [25], and Logo [31] are examples of innovation that survive now only in niches. The lie is given by the elementary classroom walls, covered with children's drawings and designs, which are testament to what the children are really learning.

Technology has an uncomfortable relationship with teaching, which I would like to illustrate tangentially with an example of a technology that threatens speech primacy.

Mental arithmetic. Geometry exists on the back of the straightedge and compass. The 'simple' step of representing the abacus in writing transformed mathematics, and led to the stored program digital computer¹. Yet such is the thrall of language and the

¹ The first electronic stored program digital computer successfully to run a program did so in Manchester, England on 21st June 1948 just after 11am. The Turing Machine / Lambda Calculus is its symbolic (graphic, imaginative) counterpart.

memory skills that underpin our lifestyle, that skill in mental arithmetic is an academic touchstone. But the four-function electronic calculator is not my example.

Reading aloud. In school, the skills of reading and writing are assigned the highest priority. Literacy development entails a set of complex mental operations. The sound stream of speech is segmented into minimal meaning units. All the personal and prosodic information is stripped out. Segmentation and abstraction completed, the information is mapped to a set of graphic symbols. Speech drifts and accents vary, so mapping is not 1:1 [31 32 34]. For English: the Romanisation has no letter for schwa, so several vowel graphemes substitute. English spelling was standardized when its speech sounds were different. Consequently, text became a quasi-independent system.

Text-to-speech synthesis [35] renders text audible. But clients want a human voice, preferably with a high-status regional accent. The data for this is not in text. Letter to phoneme rules [36] do produce intelligibility. Naturalness, however, entails extracting elements from speaker recordings; mapping the lexicon to a pronouncing dictionary; and generating prosody by rule. This destroys the lexical information and stitches in the elisions, assimilations, sex, social status and prosody of someone's speech.

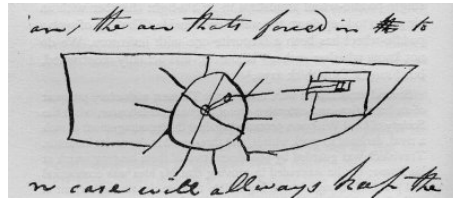
Such TTS has no pedagogical value. A phonic system is wanted, one with text as referent – the robotic sound of writing [37]. It could be built, but it has been rejected.

4.2 Technicity

At the heart of education lies graphics, not only as the means of noting knowledge, representing music or electric circuits but also the means of studying language itself. It becomes obvious (in the mathematical sense) that we have evolved a capability beyond the language adaptation. I have used variations on 'construction' to connote humanity's unique and recently evolved adaptation. However, *technicity* (coined by Heidegger [38] as translated by Dreyfus) is the better term for denoting the capability. Playwrights, novelists, poets and philosophers use the technology of writing to work words to their purpose; and linguists use writing to make speech available to scientific scrutiny. The description of language in notation proves the superiority of the graphic.

4.3 Constraints on Construction

We may now consider why constructionist approaches [40] have limited acceptance. I have suggested that language (a primitive evolved characteristic) coupled with a good memory may be an inhibitor. Whilst educators emphasise the directive role of (inner) speech as a learning facilitator, caution is needed. Recall that the Renaissance medical books juxtaposed anatomical drawings from the Leonardo school with Galenic text. The graphic contradicted the word, but the latter continued to be believed. Engineers do discuss – but with a pencil in hand ready to sketch, as the illustration below shows.



Sketch for a steam boat by Richard Trevithick, 1806

And mathematicians, who talk of the “language of mathematics” have blackboards to cover in graphical notation rather than a bank of tape-recorders to talk into.

Words. The word is unreliable: the currency of politics. Consider the pitfalls of questionnaire design [39]. It is also our primary system of communication. Given that education is a social enterprise, language (including mentalese) will, by default, predominate. The consequence is that where conflicts between text and construction arise, there will be a tendency for text to be credited over construction.

Language is not scientific; words are used loosely: ‘language of mathematics’ is a figure of speech, not an assertion that mathematics is a flavour of Sausure’s ‘langue’. Yet, this usage may mislead us into believing that it is truly a language.

This inexactitude is exacerbated by the nature of technology, which creates novelty for which words may not exist, with a consequent inhibition of articulate description.

Décalage. Technicity is recently evolved and hence may not be evenly represented in the population. Given that creative construction relies on connections to older parts of the brain for its data, the information available to individuals may vary significantly – a potential source of personal talent and expertise. There is possibly some support for this in work associated with ability measurement: Elliott [41] reports that Piagetian conservation tasks failed to scale, a consequence of inconsistent horizontal décalage.

Cost. Materiel is, of course, the major inhibiting factor. The materials, cf. science and cookery, are considerably more costly than those of traditional text-based instruction. Because assessment of educational progress is based on instructional techniques it is impossible to demonstrate superiority of outcome for construction over instruction. Hence, the economics of construction do not appear to be favourable. (This will alter as ICT costs decrease.) Note: this parallels the historical resource-dependency of technological development, which results in expenditure being targeted to ‘key’ areas.

Craft. However, the real constraint on the development of constructionist approaches is the weight of pedagogical craft and tradition: listen and recall; which has been supplemented, since Gutenberg in the 15th century, by the text-book and written assessment. The pedagogy of ICT is being developed in a few, innovative, locales; whilst the technology itself is largely being assimilated to extant instructional method.

5 Discussion

This sketch outlines a possible scientific basis for constructionism. The model posits two adaptations.

- Firstly, it is suggested that we inherit language and social organisation from a predecessor species; teamwork facilitated by the primitive hominine language adaptation lead initially to a reciprocally altruistic lifestyle, which later extended to large-group cooperative action and trade, in technology-based environments.
- The second, our species' unique adaptation, which I term technicity, is a derived characteristic stemming from neurological expansion in the pre-frontal cortex. This enables us to reconstruct deconstructed sense data creatively. It is the fount of technology and science, and the mechanism of construction.

Conversation. Vigotsky [42] suggests that we internalize speech after it develops to give ourselves an internal language (Pinker's mentalese [43]), with which we can hold conversations with ourselves and thereby think. No mechanism it proposed for this process. The model proposed here does propose a mechanism. Because 'technicity' is able to reconstruct almost any mental data, it may be the mechanism for Vigotsky's internalization of language. This implies disjunction of thought and language. It suggests that our cognitive processes co-opt and recreate speech as an instrument of communication, with others and ourselves. This is consistent with our capacity to create sign language and writing. But it also implies that language operates on two levels: a 'gossip' level [18] and a mode of communication of ideas. Crosstalk between these two may occur and might contribute to misunderstanding and misconception.

Cooperation. Human history since the transition to agriculture evidences a shift from competition to cooperation coupled with a trend towards role differentiation and trade specialisation. The social focus of Vigotsky is, therefore, best considered in a context of the evolutionary stable strategy of reciprocal altruism as implemented by a creative species. Cooperation, particularly in the field of technology, becomes a celebration of diverse talent, each developed to the maximum, contributing to a communal project.

Pedagogy. The psychology of teaching and learning is a highly confused panoply of competing theories [45]. Whilst extrapolations from animal learning behaviour have been widely applied, competing viewpoints, such as constructive/ionism remain in the realm of philosophy rather than science. Hence, the absence of a scientific alternative makes it unsurprising that pedagogy defaults to face-valid memory and language.

Transition. Whilst there is never discontinuity, phase transitions (the consequences of which are unpredictable from within the preceding phase) are common in nature. Transitions have been proposed within the biological realm [45]. Modern human thought (Piaget's constructivism expressed in Papert's constructionism) is clearly one.

TERECOP. Whilst it is accepted that this rough sketch and note will contribute only peripherally to the TERECOP workshop it may have practical offerings such as:

- suggesting that the process of construction is a uniquely human attribute and thereby exercises our highest intellectual capability (technicity).
- situating speech and memory in pre-human history and thereby signalling a need for caution in the use of discussion in learning: Consider carefully the objectives of language use because it may work to constrain creative thought.
- recognising that cooperation may have a genetic basis and that capacity for working cooperatively might well vary between individuals.
- accepting that physical construction is resource intensive and will therefore be economically constrained. Consider how ICT graphical media might achieve the constrictive objective with a lower resource demand.

In conclusion. I use the term *technicity* for our recently evolved creative capability, beneath which lies anatomically modern humanity's recall and recognition skills and language capability: adaptations prerequisite for a reciprocally altruistic lifestyle.

If this speculative thesis, further developed and researched, proves to be fruitful, it may provide a framework for re-constructing the school curriculum to offer a better balance between speech and technicity. At minimum, the two-adaptation model offers a framework for re-conceptualisation. If the analysis turns out to be a good model of reality, it should be possible to prove the constructivist/constructionist method to be the more powerful – provided appropriate measures of educational outcome are used. It also helps explain the Two Culture phenomenon. In the context of the TERECOP project, it is hoped that it will be a positive contribution to its theoretical foundation.

References

1. Alimisis D, Moro M, Arleguia J, Pina A, Frangu S, Papanikolaou K, 2007 Robotics and Constructionism in Education: the TERECOP project, IN I Kalas, Proceedings of the 11th EuroLogo Conference, Bratislava, 19-24 August 2007 (accessed online at <http://www.terecop.eu/downloads/P-Alamisis.pdf>, 6th June 2008).
2. Ofsted RAISEonline, <http://www.raiseonline.org> (accessed July 7th 2008)
3. Snow C P The Two Cultures Cambridge: Cambridge University Press 1993
4. Hogben L From Cave Painting to Comic Strip, London: Max Parish & Co. Ltd. 1949
5. Smith E E Atkinson R L Hilgard E R Atkinson & Hilgard's Introduction to Psychology London : Wadworth/Thompson Learning 2003
6. Diamond J Guns Germs and Steel London: Vintage 1998
7. Diamond J The Rise and Fall of the Third Chimpanzee London: Vintage 1992
8. Pinker S How the Mind Works London: Penguin Books Ltd. 1997
9. Deacon T The Symbolic Species: The co-evolution of language and the human brain. London: Penguin books 1998
10. Dunbar R The Human Story London: Faber and Faber. 2004
11. Goodall J Chimpanzees of the Gombe: Patterns of Behaviour Cambridge MA: Harvard University Press 1986
12. Matsuzawa T Ed Primate Origins of Human Cognition and Behaviour. Tokyo: Springer 2001
13. Lewin R Principles of Human Evolution Malden, Mass : Blackwell Scientific 1998
14. Dawkins R The Selfish Gene 2rev ed Oxford: Oxford Paperbacks 1989

15. Dawkins R *The Extended Phenotype* 2rev ed Oxford: Oxford Paperbacks 1999
16. Barrett L, Dunbar R, Lycett J *Human Evolutionary Psychology* London: Palgrave 2001
17. Axelrod R *The Evolution of Cooperation* London: Penguin 1990
18. Dunbar R *Grooming, Gossip and the Evolution of Language* 2004
19. Burling R *The Talking Ape* Oxford : Oxford University Press 2005
20. Nettle D *The Evolution of Linguistic Diversity* London : University of London 1996
21. McBrearty S Brooks A *The Revolution that Wasn't* *Journal of Human Evolution* 38: 453-563 2000
22. Mekel-Bobrov N, Gilbert S L, Evans P D, Vallender E J, Anderson J R, Hudson R R , Tishkoff S A, Lahn B T *Ongoing Adaptive Evolution of ASPM, a Brain Size Determinant in Homo sapiens* *Science* 309 5741 1720–1722 2005
23. Pollard K S, Salama S R, Lambert N, Lambot M-A, Coppens S, Pedersen J S, Katzman S, King B, Onodera C, Siepel A, Kern A D, Dehay C, Igel H, Ares M Jr, Vanderhaeghen P, Haussler D *An RNA gene expressed during cortical development evolved rapidly in humans* *Nature* 443, 167-172 2006
24. Carter R *Mapping the Mind* London: Phoenix 2000
25. Nettle D *Strong Imagination: madness, creativity and human nature* Oxford: Oxford University Press 2002
26. Feuerstein R, Falik LH, Feuerstein R S, Rand Y *The Dynamic Assessment of Cognitive Modifiability: the learning propensity assessment device: theory, instruments and techniques*. Jerusalem: The ICELP Press 2002
27. Hubel D H *Eye, Brain and Vision* New York: Scientific American Library
28. Heidegger M *The Question concerning Technology and other Essays* (tr W Lovitt) London: Harper and Row 1977 1995
29. Murphy D J *Comenius: a critical reassessment of his life and work* Blackrock Co Dublin : Irish Academic Press 1995
30. Lillard A S *Montessori: the science behind the genius* Oxford: OUP 2005
31. LCSI *Logo Philosophy and Implementation* Quebec, Montreal : Logo Computer Systems Inc 1999
32. Downing J A *Evaluating the Initial Teaching Alphabet* London : Cassel 1967
33. Adams M J *Beginning to Read: thinking and learning about print* Cambridge : MIT Press 1997
34. Johnstone R S *Accelerated Reading and Spelling with Synthetic Phonics* Edinburgh : Scottish Executive, Education Department 2003
35. Taylor P *Text-to-Speech Synthesis* Cambridge: Calbridge University Press (draft accessed on the net 2007) 2008
36. Elovitz H S, Johnson R W, McHugh A, Shore J E *Automatic Translation of English Text to Phonetics by Means of Letter to Sound Rules* NRL Report 7948 Washington DC: Naval Research Laboratory 1976
37. Doyle M P *The Computer as an Educational Medium* Unpublished M Phil Thesis, Manchester University 1986
38. Dreyfus H personal communication 2003
39. Oppenheim A N *Questionnaire Design, Interviewing and Attitude Measurement* London : Continuum 2003
40. Kafai Y, Resnick M *Constructionism in Practice* Mahwah, NJ : Lawrence Earlbaum Associates 1996
41. Elliott C D *British Ability Scales: Introductory Handbook* Windsor, Berks : NFER-Nelson
42. Vigotsky L S *Thought and Language* Cambridge, Mass : The MIT Press 1962
43. Pinker S *The Language Instinct* London : Penguin 1995
44. Smith J M and Szanthy E *The Origins of Life* Oxford : Oxford University Press 2000
45. Hoy A W et al *Psychology in Education* Harlow : Pearson Longman 2008

It Takes a Village... to do Science Education

Paolo Fiorini¹, Stefano Galvan¹, Loretta Giuliari², Laura Pighi³

¹ University of Verona, Computer Science Department, Verona, Italy

² Istituto Comprensivo Don Milani, San Pietro di Lavagno, Verona, Italy

³ Società Dante Alighieri, Verona, Italy

Abstract. This paper describes the efforts undertaken by a small community of concerned teachers to boost science education in the school district of Verona (Italy) by promoting constructivism with the help of various configurations of robotic devices. These efforts have been going on for the last eight years, slowly gaining momentum and impact. However, the most striking difficulties have been with the education environment, rather than the student themselves. We report on the development of curricula for Middle and High Schools using the LEGO kits (the Kineplay and Eddy projects), on our efforts to involve in these activities teachers at various grades, and in particular on the sensibilization of the education administration, of the families, and of different city organizations, thus showing that science education must truly be a community effort.

1 Introduction

In many Civilized Societies, media exposure and loud talk have become a sign of professional competence, and the need of hard work and in depth understanding have become useless and irrelevant for large sectors of the public opinion. This is very evident in Italy, where it is extremely difficult to reverse this established opinion and propose an education model aiming at restoring scientific competence, creativity, and curiosity in the students. In fact, the difficulties are not only in the need of developing new educational formats, since it is not possible to propose educational models of the past century in the world of Internet and cellular phones, but also in shaking up a disappointed educational staff, in getting the attention of the public administrators busy with politicking, and of the entrepreneurial world for whom schools and academia have become almost irrelevant. Science education in Italy should be a truly global effort requiring to address also the following problems:

1. Elementary Schools: Propose new teaching formats that could excite students about science projects.
2. Middle Schools: Overcome the current situation of directing gifted children towards humanistic studies.
3. High Schools: Develop multi-disciplinary projects that could foster team spirit together with scientific excellence.
4. Universities: Exit from the ivory tower mentality and address the specific societal needs of technical innovation.
5. Adult Education: Providing solid means to update the background of technical professionals.

Thus a village may not even be sufficient to make a dent in this huge problem.

In the last few years, robotics has been proposed in Universities and High Schools as an innovative tool to teach scientific subjects. Scientific education is greatly improved when classroom teaching is supported by adequate laboratory courses and experiments following the *inquiry based learning* pedagogical approach. However, since the cost of laboratory equipment is an important issue, this approach has been seldom implemented until low cost robotic devices have allowed developing cost effective laboratory practices. Moreover, adequate teaching material should support the technological instruments, such as specific syllabi, introductory textbooks, evaluation instruments and so on. This supporting material is currently not available, thus leaving to the teachers a great deal of additional work. Lastly it would be advisable to have good communication and coordination between the various grades of education, and among institutions, to share educational material and to focus on ambitious goals that can be reached with difficulty by single institutions.

In this paper, we present the experience of several groups of dedicated teachers at various grade levels in using robotics to teach scientific subjects, including robotics itself. We briefly summarize the results of our sparse experiments from Elementary Schools, to academic teaching and adult education, and show the need, still unanswered, of synergy and coordination among institutions and within each course to transfer more excitement to the students about science and its importance in the society. We will start by describing the best developed tools, i.e. those developed for High School and academic activities, focusing on Kineplay, the learning environment developed using the LEGO[®] Mindstorm[™] [9], and Eddy (Educational Device: Do it Yourself!), a low cost educational mobile robot [5]. Then, we describe the activities in Middle and Elementary Schools that have been carried out at Istituto Comprensivo Don Milani, in San Pietro di Lavagno (Verona) using the standard LEGO Mindstorm tool. Adult education and advanced subjects to technical High Schools have been funded by the Veneto Region, as a result of intensive lobbying efforts. Thus, in the last Section we address the need of creating a large support base in the public and in the various stakeholders, to raise awareness about the poor scientific competence of our students and its implications for the future. This efforts have been undertaken by the Verona branch of the Dante Alighieri Society, whose representatives have embarked in a series of scientific lectures to schools and institutions to illustrate this problem and present possible solutions.

2 Past work on robotic education

The multidisciplinary nature of robotics makes it a natural tool for science and engineering education at many levels. Robotics has been shown to be a superb tool for hands-on experimental learning, not only of robotics itself, but of general topics in Science, Technology, Engineering, and Mathematics (STEM). From a pedagogic point of view, robotic hands-on experiments follow the constructivism learning paradigm. These

ideas go back to Piaget's pedagogical work, but take also into account the intuitions of Vygotskij on the proximal developmental area in which children acquire their knowledge primarily through social interaction, and Bauer's view of the importance of shared experience with teachers and other students. Constructivism was further refined by Papert in his paper [14], where he filled the gap between active learning and technology thus laying the foundation of the use of computers and mechanisms for education. Thus the ideal learning model should include a well balanced mixture of hands on experience, supported by the appropriate technology and mediated by teacher's account of past experience and explanation of theoretical background.

Depending on the students' grade, robotics can be the goal or the means of education. The former approach is followed in specifics courses at universities, while the latter is more related to K-12 education. Traditional and hands-on approaches to robotics teaching have been explored in several workshops [6] [17] and conference special sessions. In [12] the author describes the urgency of providing K-12 teachers new instruments and materials for their courses. However, the focus has been mostly on higher education, with only a few experiments reported on K-12 teaching. Even for High School and academic teaching, it is hard to find good tools to support laboratory activities. Furthermore, no material is available in Italian. In [15] an interesting virtual laboratory for kinematics is presented, but it is no longer available on-line. In [16] a computational construction kit is presented that encourages users to experiment and play with a collection of sensor, logic and actuator blocks, exposing them to a variety of advanced concepts including kinematics, feedback and distributed control. Finally, a recent initiative *RoboticsCourseWare.org* is collecting and organizing robotics courses from leading Universities in an open source, copyright free form, to give teacher worldwide enough material to develop courses specific to their needs. However, no such initiative is available for High and Middle School teachers, who are perhaps the ones most needing training and support.

Whether addressing the needs of higher education or those of Elementary Schools pupils, it is important to give students many simple robots that are cheap, safe, easy to use and in some cases even prone to be broken to let students explore all the implications of their actions. Unfortunately, since the cost of a robotic laboratory is high, inexpensive robotic devices must be developed either from scratch or using the available construction kits. A particular interest is on using low cost commercial platforms, i.e. adding sensors and boards to the iRobot device [13] and adding parts to standard LEGO Mindstorm kits as described in [17].

It must also be noted that while the material on robotics education at various grades and competence levels is rather abundant, very little is available on using the robotic kits to teach general scientific subjects. This need has motivated part of the work described in the following Sections, where we present the steps undertaken to establish general science education curricula using robots.

3 High School Tools: Kineplay and Eddy

Kineplay is the name of the curriculum that we have developed to teach elementary concepts of fixed manipulators using LEGO Mindstorm sets to High School students

Traditional robotics classes cover concepts such as rigid body transformations, forward and inverse kinematics, velocities and Jacobian of linkages, mechanical design aspects and programming of robots. Although many of these notions are complex, basic kinematics is rather simple, especially if it is explained with the aid of laboratory sessions. To set up the laboratory exercises we overcame two misconceptions about the LEGO robotic kit. First, that it is not a serious tool for school courses, and second that it is only suitable for teaching simple concept of mobile robots. On the contrary, the LEGO kit allows to design and build a fully operational fixed robot, a task can be hardly done with other laboratory equipment of the same price range.

In this course, we apply the constructivism paradigm to the way kinematics concepts are taught. We provide a quick overview of the basic concepts in the frontal lectures, and then we let the students carry out the laboratory experience by interacting with the tutors to clarify the supposedly known mathematical foundations, such as geometry, matrix algebra and trigonometry. The frontal lectures are done partly in the High School, to refresh the basic mathematic and geometry concepts, and partly at the University, to introduce the kinematics tools. Usually these two parts are organized into two sections of 10 hours each. Then 12 hours are devoted to laboratory practice, to apply these concepts to building and operating a robot made with LEGO bricks.

We are still using the old RCX version of the LEGO kit, because the new NXT series is less flexible in building kinematic structures and forces the students into a set of pre-determined mechanical configurations.

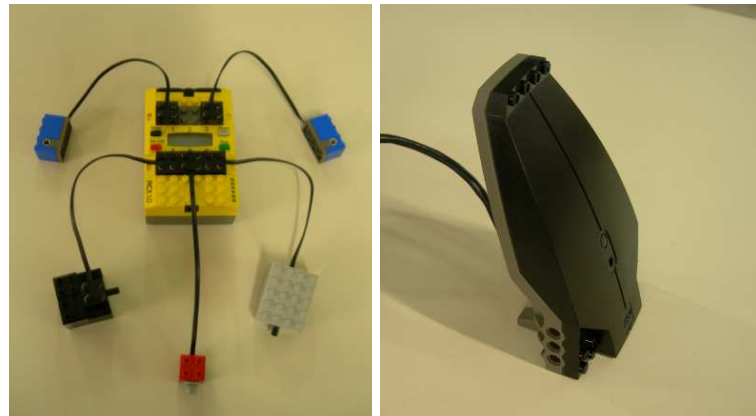


Fig. 1. RCX with actuators and sensors and infrared tower for communication

3.1 The Kineplay Environment

To enhance the flexibility of the LEGO kit, we used *brickOs*, a firmware directly installed on the LEGO controller RCX. Thanks to this firmware the RCX can be programmed directly in C or C++ and the only limitation is given by the amount of memory available in the system. The compilation of the programs on a PC is easy because the RCX processor is a standard Hitachi 8300, for which several cross compilers are available. To complete the laboratory set up, we have installed six low end PC running the cross compiler and equipped with the USB support for the LEGO infrared tower. PCs are equipped with Slackware Linux Kernel 2.6, which has a built-in driver for the infrared link. Figure 1 shows the standard RCX LEGO processor and the infrared communication tower. We run programs directly on the RCX brick so that students can better understand the problems related to embedded system design. To simplify software development and to let students focus on the robotics problems, we developed a basic software infrastructure subdivided into three main parts: drivers, communication, and manipulator control.



Fig. 2. Some of our students at work and one of the robots built by the students.

The laboratory is organized into three phases. The initial phase is the manipulator design that allows students to get acquainted in a fun way to various aspects of technical design, team work, and time constraints. None of the concepts related to mechanical design were introduced during the frontal lectures, and students learn first hand the importance of mass balance and static stability. Figure 2 shows a group of students at work and one of the robots built by the students.

The second phase of the laboratory is the kinematic analysis. Students use different approaches to solve the inverse kinematic problem. Some of them use the standard approach discussed during theory lectures and follow step by step the examples given. Other students more confident with the computation of matrix transformations develop more advanced solutions.

During implementation, students have only to insert functional and structural parameters in the robot program. They do not have to do any real programming, because the background in computer science of students

from different High Schools would have required too much time for this phase.

Once the robot has been fabricated and programmed, the students can verify the correctness of their implementation by displaying on the RCX block the Cartesian value of the end effector position. Motion resolution is basically comparable to the size of a LEGO brick that becomes the position measurement unit.

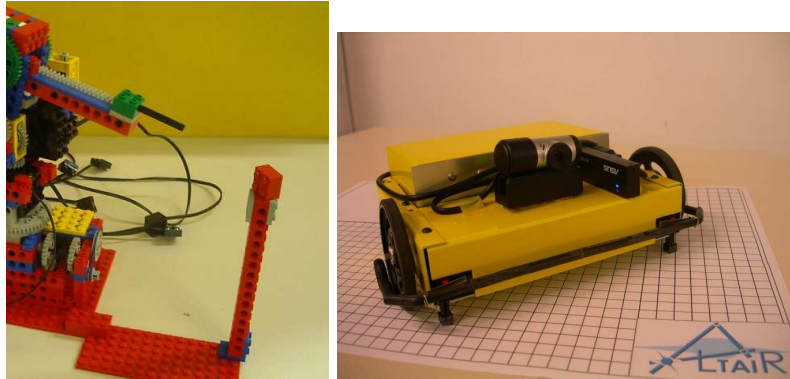


Fig. 3. The left figure shows the "Drop the Brick!" final test, and the right side shows the Eddy Robot.

The final phase consists of throwing away a LEGO brick from a tower whose coordinates are given in advance to the students, as shown in Figure 3. We called this task "Drop the Brick!". We also put some static obstacle on the manipulator path introducing complexity in the task. Since the beginning, this approach to teach robotics has shown many positive results. Students are very enthusiastic, they learn to work in team in the design phase dividing tasks among themselves and scheduling their work. We repeated the Kineplay experience several times: High School students are recruited by the Tandem project, a collaboration between the University of Verona and High Schools, and also with our own students. More than 200 students from 10 schools have already attended these robotics courses, with very satisfactory results from the students' personal point of view. It is of course very difficult to assess whether this course has any direct impact on the student interest in sciences and on their future academic career. Attending students belong to the last two years of High School and no long term measurements were taken. Furthermore, the data on their academic career are not available to us.

3.2 Eddy

The Kineplay experience was positive for us, but we realize that when the robotic course was over, there was no motivation for the students to

continue working on robots, and no long term project could be started with the High Schools. Thus, we decided to build a small mobile robot that schools can use for year-long projects, without the limitations imposed by the LEGO kits.

A few companies have started producing small robots for education but it is difficult to find robots that are cheap enough to be affordable for High Schools and highly configurable to adapt to specific educational goals. Khepera [11] robots are well-known, modular and robust but their cost is not affordable for schools. Those considerations held also for many commercial and research product [7,8]. Others valuable devices are related to a specific application field [10] or more oriented toward an evolutionary approach, where robots are used as a network of semi-intelligent sensors, such as [1] and [2].

To overcome these problems we developed Eddy, a robotic platform for education, that students and teachers can build together shown in Figure 3. Our aim is to provide an inexpensive mobile platform with highly customizable sensor capability. We follow the Open Hardware Paradigm [5], and provide all schematics and source code to let students build, enhance and use their own robot [4]. The overall cost of the robot parts is about 300 Euro, which is affordable to most Italian High Schools.

Eddy is a small robot; however, it is not just a micro-controller that may act as a “proxy” for the sensors, like most of the economic systems, such as Fischertechnik, Basic STAMP, or Scribbler. By using a fairly standard CPU with a stripped-down version of a GNU/Linux distribution, the only limits are the device support (on *kernel side*) and the amount of memory available for applications. With this system, in few hours it was possible to develop a very simple software (running on Eddy) to control the robot with a standard Bluetooth USB device and a Nintendo[®] *wiimote*, using one of the many open source Linux libraries already available [3].

Following the encouraging results we obtained with Kineplay we are working on the development of easy software tools for Eddy, to make the users concentrate on the learning aspects, rather than dealing with software and hardware development.

4 Middle and Elementary Schools: A Two Prong Approach

At Universities and High Schools it is possible (after a long search) to find teachers who take upon themselves to learn new teaching tools and apply them in the classroom. However, in Middle and Elementary School, teachers are seriously worried about their ability to learn new tools and to be outsmarted by their students. Thus it is not possible to address only the students needs, but special attention must be given to train the teachers to the new tools.

In this context, the learning objective for the students is to stimulate their *active thinking*, i.e. the ability to find and build their own know-how by trail and error, acquiring new information when needed and experimenting until an appropriate solution is found. The goal is towards

new and emerging technologies, so robotics in itself is important, but space must be given also to energy and environmental issues, just to mention two. Furthermore, the comparison with the solutions proposed by the other students, and the evaluation of the different performance will provide the evaluation of the work, better than any grade assigned by the teacher. In this case the teacher is not asked to provide "the" solution to a problem, rather to act as a facilitator, helping and stimulating the student to find a solution and learning him or herself when, as it is often the case, the student is faster than the teacher.

However, this is the main difficulty of bringing new learning tools, and in particular robotics, to Elementary and Middle Schools, i.e. teachers are afraid of looking bad in front of their students. To overcome this problem, we organized a year long program to give teachers the self confidence and the technical knowledge required. The first step was a short course at the University of Verona, in which about 20 teacher from various Middle and Elementary Schools of the province of Verona were taught the basic concepts of robotics and of using the LEGO NXT kit. The lectures first aimed at introducing teachers to the various aspects of robotics and to their future impact and current relevance. Then, the focus switched to providing teachers with the practical knowledge of using the LEGO NXT sets, by executing the basic examples in the kit. The second phase of the program consisted of a series of self guided meeting, in which the teacher applied these basic concepts to develop learning units on various scientific subjects using a LEGO laboratory. Specifically, the objectives of these meetings were:

- Attract the attention of the education establishment towards the importance of science and technology in everyday life.
- Help growing the scientific and technological culture of the students, by means of higher quality teachings.
- Stimulate the practical understanding of mathematical concepts.
- Develop learning models following the social constructivism paradigm.
- Start a virtuous circle by which students become the builders of their own knowledge.
- Address the issue of intelligent machines: from design to fabrication.
- Rethink the curricula design, and develop new laboratory learning units.
- Develop the new teaching model of *teacher-researcher*, who is able to acquire new competences and able to promote innovation in teaching.
- Start robotic classes in the Elementary School, to establish a science and technology learning path, from earlier grades.
- Develop, within each school, robotics laboratories.

Currently, eight schools are involved in this program, with about 20 teachers developing scientific learning units based on robotic tools. The program received the support of the Education Administration of Verona, and a small financial support to cover teachers' expenses. The result of this work will form the core of a curriculum that will be distributed on the Internet for other schools to use. However, some difficulties are emerging with respect to the autonomy of the teachers. In fact, after carefully following the course and acquiring the tools and material to

add new teaching material to the curriculum, there has been a slow down in activities and some lack of creativity, which we hope will be soon overcome.

5 Lobbying for Education

Of course, without enough "political" support, initiative such as the one described above cannot have a wide diffusion and cannot be adopted by a large enough number of schools to make an impact. Therefore, the Verona branch of the Dante Alighieri Society took upon itself to carry out the lobbying efforts. The Society representatives contacted the Administration of the Verona School District, possible donors, and institutions interested in improving science education to coordinate a city wide effort to disseminate the educational experiments described above.

Many other lobbying efforts were started, in particular by technical High Schools, desiring to improve the quality of their offering. A first result of these efforts was the establishment of a Robotics District in the city of Verona (sponsored by the Veneto Region), that supported the creation of after-hour lectures on robotics to students and to adults as well. These lectures were organized in courses ranging from Control Theory, to Robotics, to CAD, to advanced computer programming. The courses allowed the participant to achieve a good understanding of these advanced subjects and to receive a certificate of participation after a final examination.

The results of the new science and robotics activities will be demonstrated with a year long series of events, involving several schools of the Verona district. A number of teachers will volunteer to bring experiments and new lectures to various schools and to mentor both teachers and students who will start year-long science projects. The projects will be first presented at the beginning of the school year in a workshop opened to students and teachers, and demonstrated at the end of the year in a science festival coordinated with the Museum of Natural Sciences of Rovereto. Since 2001, The Museum of Rovereto organizes the science festival *Discovery on Film*, showcasing various aspects of technology, and demonstrating students projects that have been carried out during the previous months. We plan to organize a similar festival in Verona, which will also help attract industry and local institutions to our educational efforts.

6 Conclusions

In this paper we describe the approach taken in the School District of the city of Verona to attempt at increasing the student competence in science subjects. We started by developing curricula for High Schools by teaching robotics, taking advantage of the appeal of this subject on the students. However, it was rather evident that robotic devices could also be useful tools for teaching other subjects and at other student grades. Thus, encouraged by the interest of teachers and their results

with students, we started adapting the tools and lectures for teachers as well as students of a number of different schools.

We developed two tools for teaching robotics at the High School level, which have now been used with more than 200 students, with excellent results. We have started a small teaching program for Middle and Elementary School teachers, which has now spurred the development of science curricula specifically dedicated to the needs of lower grade students. Finally, realizing the importance of a global effort to impact science status in the society at large, we started lobbying various institutions and local administrations to try raising their awareness and interest to scientific culture in our society.

Whether robotics is the goal or robotics is the medium used to teach other subjects, it is important to have the correct evaluation instruments to verify that the students learn what we want them to learn. While it is easy to assess the enthusiasm of the students and their efforts to correctly finish the activity or win a contest, it is more complex to verify that the notions they learn will endure after the course. At the moment we think that this is the big issue about robotics *in* and *for* education together with teacher training (especially in K-12 courses), and the availability of ready-to-use robotics tools and textbooks.

References

1. The e-puck robot. <http://www.e-puck.org>.
2. The jasmine micro-robot series. Available at: <http://ipvs.informatik.uni-stuttgart.de/BV/swarmrobot/tikiwiki-1.9.2/tiki-index.php>.
3. A simple wiimote library for linux. Available at: <http://libwiimote.sourceforge.net/>.
4. Altair website. Eddy. Available at: <http://metropolis.sci.univr.it/eddy>.
5. L. Bertelli, F. Bovo, L. Grespan, S. Galvan, and P. Fiorini. Eddy: an open hardware robot for education. In *AMIRE 2007 Conference, 4th International Symposium on Autonomous Minirobots for Research and Edutainment*, Buenos Aires, Argentina, October 2007.
6. A. Casals and A. Grau, editors. *1st Workshop on Robotics education and training*. Euron, Weingarten, Germany, 21 July 2001.
7. EPFL. Alice. Available at: <http://asl.epfl.ch/robots.php>.
8. Fraunhofer. Volksbot. Available at: <http://www.ais.fraunhofer.de/BE/volksbot>.
9. S. Galvan, D. Botturi, A. Castellani, and P. Fiorini. Innovative robotics teaching using lego sets. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 721–726, 15-19, 2006.
10. A. Golovinsky, M. Yim, Z. Ying, C. Eldershaw, and D. Duff. Polybot and polykinetic/spl trade/ system: a modular robotic platform for education. *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, 2:1381–1386 Vol.2, 26-May 1, 2004.

11. K-Team. Hemisson, koala ii and khephera ii. Available at: <http://www.k-team.com>.
12. M. J. Mataric. Robotics education for all ages. In *AAAI Spring Symposium on Accessible, Hands-on AI and Robotics Education*, Palo Alto, CA, March 2004.
13. M. J. Mataric, N. Koenig, and D. Feil-Seifer. Materials for enabling hands-on robotics and stem education. In *AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education , Hands-on AI and Robotics Education*, Stanford, CA, March 2007.
14. S. M. Papert. *Mindstorms: Children, Computers, and Powerful Ideas, Second ed.* Basic Books, New York, 1999.
15. M.F. Robinette and R. Manseur. Robot-draw, an internet-based visualization tool for robotics education. *Education, IEEE Transactions on*, 44(1):29–34, Feb 2001.
16. E. Schweikardt and M. D. Gross. roblocks: a robotic construction kit for mathematics and science education. In *ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces*, pages 72–75, New York, NY, USA, 2006. ACM.
17. H. Zhang, W. Zheng, S. Chen, J. Zhang, W. Wang, and G. Zong. Flexible educational robotic system for a practical course. *Integration Technology, 2007. ICIT '07. IEEE International Conference on*, pages 691–696, March 2007.

Representative examples of implementing educational robotics in school based on the constructivist approach

Stassini Frangou¹, Kyparissia Papanikolaou², Liliane Aravecchia³, Luc Montel³, Silviu Ionita⁴, Javier Arlegui⁵, Alfredo Pina⁵, Emanuele Menegatti⁶, Michele Moro⁶, Nello Fava⁷, Stefano Monfalcon⁷, Irene Pagello⁸

- (1) Dept. of Philosophy and History of Science, Interdisciplinary Program of Graduate Studies in Basic and Applied Cognitive Science, Greece, stassini.frangou@sch.gr
- (2) Dept. of Education, School of Pedagogical & Technological Education, Athens, Greece, spap@di.uoa.gr,
- (3) Institut Universitaire de Formation des maîtres, Dept. Education Technologique et Sciences et Technologies Industrielles, Université de Provence Liliane, France, {l.aravecchia, l.montel}@aix-mrs.iufm.fr
- (4) Dept. of Electronics and Computers, University of Pitesti, Romania, ionis@upit.ro
- (5) Public University of Navarra, Pamplona, Spain, {arlegui, pina}@unavarra.es
- (6) Dept. of Information Engineering, University of Padova, Padova, Italy, {emg, mike@dei.unipd.it}
- (7) Town Museum of Rovereto, Italy, museo@museocivico.rovereto.tn.it
- (8) IT+Robotics srl, Vicenza, Italy, irene.pagello@it-robotics.it

Abstract. Educational Robotics (ER) is a powerful technology which combines both constructing and programming a robot model. As such it can address teaching objectives from a wide range of disciplines from computer science and technology to design, mathematics and science education. Additionally ER has strong experimental characteristics which can effectively support innovative constructivist approaches to teaching and learning. In this paper we focus on the design of robotics enhanced activities emphasizing the main constructivist principles adopted. Secondly we illustrate these aspects through some representative examples.

Keywords: educational robotics, constructivism, constructionism, secondary education, Lego Mindstorms, project based learning, educational technology

1 Introduction

Educational Robotic (ER) systems consist of building material and software facilities which allow the construction and the programming of various robots from smart cars to chimney cleaners. Robots have sensors and machines like motors. They collect data from their environment and use them as parameters. An important feature of this technology is that it can be very simple to use for constructing a model and programming it, while users can create extremely sophisticated applications. So it can be used equally effectively by primary and university students. Moreover may ER

support a wide range of different explorations. It can be described as ‘having low floor, high ceiling and wide walls’ [1].

First research projects of ER technology are going back to '80. Then, there were robotic turtles which could be programmed with Logo. In our days many robotic systems are proposed for school use. An interested system is the NXT version of LEGO robotics which is supported by a graphical programming interface for developing robotic applications.

Activities with ER can serve learning objectives from a wide range of disciplines from technology and design to mathematics and science education. They are hands-on activities with important experimentation features. From this point of view ER creates an active, cooperative learning environment which emphasises on students' participation. So incorporating robotic technologies in school curriculum can enrich teaching practices with great impact in addressing teaching objectives from different disciplines with an innovative way.

Moreover developments in cognitive psychology, cognitive science and the education field support the idea that learning is a process heavily influenced by learners' previous experience. Learning is considered as an active process through which new meaning is constructed by learners. This approach to learning which is common to many theoretical and experimental works in many disciplines is now known as the constructivist approach.

The aim of this paper is to explore important aspects of robotic applications at schools that make them appropriate for designing learning activities based on constructivist principles. In section 2 we describe the main characteristics of teaching and learning within the constructivist approach and we discuss their implications on the design of robotic enhanced activities. In section 3 we present a methodology for developing such activities and we illustrate our proposal with six examples created for and used in the teachers' training seminars organized in the context of the TeReCoP project. The paper ends with concluding remarks concerning the learning opportunities promoted by such robotic enhanced activities.

2 Implementing Educational Robotics in the classroom

ER technology can be considered as an educational tool. Research in Greece, Italy, Spain, France, Romania, Czech Republic shows a small number of implementations in real classroom environment of ER technology in primary and secondary schools and in tertiary education. What is really interesting is the great number of robotic research projects which can be listed in all levels of education [2]. Although these applications vary concerning their objectives and methodology, most of them adopt a constructivist perspective emphasizing on collaborative and student centered learning activities. So as a first step we should look closely in some theoretical issues of constructivism.

Constructivism is a theory about teaching and learning with roots in philosophy, psychology, sociology and education. According to constructivism learning is “a self-regulated process of resolving inner cognitive conflicts that often become apparent through concrete experience, collaborative discourse and reflection” [3]. The central

idea of Constructivism is that human learning is *constructed*. Learners build new knowledge upon the foundation of previous one. This view of learning presupposes that knowledge is an individual construction which corresponds to physical world. What is important is learner's currently believes. No matter if they are correct or incorrect, despite having the same learning experience with somebody else, each learner constructs individual meanings [4].

Two important notions orbit around the idea of constructed knowledge [5]. The first is that learners construct new understandings using what they already know. Learners confront their understanding in the light of what they encounter in the new learning situation. If what learners encounter is inconsistent with their current understanding, their understanding can change to accommodate new experience. So learning may involve some minor conceptual reorganization or major conceptual change. The second notion is that learning is active rather than passive and depends upon learners taking responsibility to learn.

Constructivism, despite the criticism about its coherence, has important implications for teaching that should be carefully considered when designing instruction [4]. Learning is based on prior knowledge so learning environment should exploit students' current ideas in relation with newly introduced information. New knowledge is actively built so students experimentations are important element of the teaching process. Students may need different experiences to advance to different levels of understanding, so activities which encourage multiple representations of concepts and relations are suitable. Students should apply their current understandings in new situations in order to build new knowledge, so open ended tasks should be incorporated in learning process. This constructivist view of learning also influences the role of teachers. The main task that teachers are assumed to perform, according to constructivists, is no longer the transmission of knowledge, but the facilitation and coaching of learning [6].

Constructionism proposed by Papert and his colleges at MIT, is aligned with constructivism in the case of learning with computer technology and ER technologies. In Paper's words: "*It is easy enough to formulate simple catchy versions of the idea of constructivism; for example of it as 'learning-by-making'*" [7].

The constructionist approach involves learners building knowledge and meaning through the construction of something external or shareable [7]. Furthermore, such a process also provides a motivating context for students to learn the subject matter and content and test their knowledge. Just as maintained by Puntambakar and Kolodner [8] that when students are engaged in multiple cycles of designing, evaluating, and redesigning, they also have the opportunity to confront their understanding and misunderstandings of concepts. Effective design projects involving ER according to Resnick and Ocko [9] are the:

- Design projects that engage kids as active participants, giving them a greater sense of control and responsibility for the learning process.
- Design projects that encourage creative problem-solving.
- Design projects that are interdisciplinary, bringing together ideas from art, technology, math, and sciences.
- Design projects that help kids learn to put themselves in the minds of others, since they need to consider how others will use the things they create.
- Design projects that provide opportunities for reflection and collaboration.

- Design projects that set up a positive-feedback loop of learning: when kids design things, they get new ideas, leading them to design new things, from which they get even more ideas, leading them to design yet more things, and so on.

Based on and expanding the above mentioned ideas, we conclude on several principles about the design of robotic enhanced activities and their implementation in real classrooms: (a) collaborative activities should be undertaken by students working in groups and in plenary as knowledge is the result of a carefully organized discussion and collaboration, (b) learning activities should be experimental, practical and explorative as knowledge is achieved through a set of tasks which reveal students' current beliefs, (c) learning activities should cultivate students' metacognitive skills like reflection, self regulation and self assessment

3 Representative Examples

An appropriate method for organizing students' activity in ER is project-based learning. Project-based learning (PBL) emphasizes learning activities that are long-term, interdisciplinary, student-centered, and integrated with real world issues and practices. PBL focuses on relevant and useful tasks for students by establishing connections to life outside the classroom, addressing real world concerns, and developing real world skills. PBL cultivates a variety of skills including the ability to monitoring their work, cooperate with others, make thoughtful decisions, take initiatives and solve complex problems.

Designing and implementing robotic-enhanced projects could be a very demanding teaching and learning activity. The methodology we propose for organising ER activities consists of the following five stages: The first stage is the engagement stage in which teacher and students explore a general issue and they set the problem that their project is going to address. At the second stage, the exploration stage, all necessary new knowledge, skills and tools are introduced through practical activities and experimentations. The third stage, the investigation stage, consists of open ended investigations based on questions related to the initial problem. At the fourth stage, the creation stage, students, in small groups, synthesize and propose solutions to the initial problem. Finally at the fifth stage, the evaluation stage, each group presents their work and receives feedback from their colleagues and the teacher. Although this methodology is suggested here for ER projects, it can be utilized for organizing any lesson (teaching period).

Based on the above methodology the six pilot teachers' training courses on ER were developed in the context of the TERECOP project. At this section we will describe four representative examples which have been used for training purposes during the courses and two projects created by trainees as an outcome of their work in the course. Some of these examples have been implemented in classroom environment and some will be implemented during next year. So, at this point we are not able to present evaluation data from the implementation of the projects in real classrooms.

3.1 The BusRoute (Greece)

The BusRoute is a project for introducing educational robotics to students of age 12 to 14. It addresses objectives of Mathematics, Science, Technology, and Computer Science. After completing the project students will be able: to describe the basic characteristics of a robot (Technology); to design and construct a moving vehicle (wheels, axles, motors) (Technology); to use suitable software and programming structures in order to perform specific tasks (Computer Science); to calculate physical quantities affecting the design and operation of a vehicle, (Physics, Mathematics). Skills which students may use or develop in order to complete their tasks are: problem solving, experimenting, argumenting, evaluating arguments, organizing, monitoring their work/progress, and cooperating. Meanwhile they form a positive attitude toward robotic technology. The project can be completed in 12 teaching periods (45min). A suggested teaching sequence according to the model presented above is the following.

Engagement stage: Students are introduced to the theme of this project: “traffic within a town”. Photos and videos are used to trigger students’ interest and initiate discussions in small groups and in plenary. A scenario (a robotic bus which could operate in the centre of a town) is used to present the initial problem. Then students are asked to present their own experiences and believes in order to define, in detail, the final problem that they are going to investigate.

Exploration stage: Students are introduced to the basic functions of the construction materials and basic programming techniques: construction of a bus which can move to all directions, design and test a program which moves the bus forward –backwards, design and test a program which turns the bus, design and test a program which moves the bus on a square, design and test a program which moves the car on a predefined path, control the bus through a touch sensor, control the bus through a light sensor. Students are performing the tasks following specific instructions (provided in appropriate worksheets), they are gradually introduced to experimentation, and they are encouraged to observe, evaluate and generalize on important aspects of the newly presented information. The final outcome is the construction of a robotic bus which can move around, turn and controlled through its sensors.

Investigation stage: The general problem, as it was formed in the engagement stage, is analysed in smaller questions. Examples of questions could be: ‘How the robotic bus parks and how it starts off at the terminal?’, ‘How it will move on a pre–defined track?’, ‘How to deal with situations of danger or an obstruction?’, ‘How it will stop at the bus stop and wait for passengers?’, ‘How could it serve disabled people?’, etc. Each group, in this case, is working on a different question. At the end of their investigation they present their solution to the rest of the class. The work of each group in this stage is completed independently and students should monitor their own progress. Diaries are kept by students in order to promote self-monitoring. Students are asked to propose and test ideas, complete and evaluate their tasks. The task is open-ended and the proposed solution is acceptable as far as it is effective. In this stage the teacher’s role is to create the appropriate learning environment and to encourage participation of and contribution from all the members of the class. Part of this stage is the agreement upon the evaluation criteria of the final solution.

Creation stage: At this stage students are asked to synthesize the proposed solutions and to create a complete answer to the initial problem. They prepare presentations of their work. Students are participating with ideas, argue, negotiate and justify their choices.

Evaluation stage: Each team is asked to present their project and participate in the discussion. They are asked to evaluate their own work and the work of other groups. The teacher gives feedback to the students.

3.2 Robotics challenge (France)

This project was designed and implemented in a classroom by three students-teachers (Technology Teachers) of the French “Teachers Training Institute”. It is based on the following challenge: A robot has to go from A to B either through a labyrinth with colored walls (white when the path turns left and black when it turns right) or following a black line on the floor. This is an activity for pupils aged 12-13, in the part of their technology course treating of “computer aided piloting”.

The target skills are part of the French Technology curriculum. After the end of this project students are expected to be able to:

- Identify the different parts of the robot ;
- Identify and justify the sensors and actuators used ;
- Represent the various stages of the movement by observation of the robot ;
- Modify an existing program according to the specifications given ;
- Adapt the system to a new situation.

The project is to be completed in 5 hours.

Engagement stage: Pupils watch a video on robotics, followed by a discussion. The robotics challenge is then presented.

Investigation stage: Pupils analyse the route the robot will have to follow from A to B and decide on a strategy to program the robot.

Creation stage: Pupils modify the existing robot by implementing the sensors and the program chosen according to their defined strategy

Evaluation stage: The different projects from each group of pupils are analysed and compared by the class, and a synthesis is made by the teacher and the pupils.

The results of the implementation of this project were presented in a professional report as part of the evaluation of the students as future teachers.

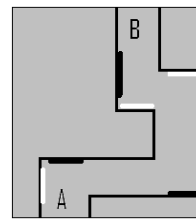


Fig.1. The labyrinth

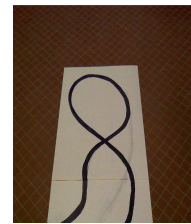


Fig.2. Following the black line

3.3 Automated camera (France)

The firm ERM sells an automated production line called “ERMAFLEX” that fills, packages and packs flasks of different types. In order to present its machine to future

clients, the firm wants to make a video of the course followed by a flask along the production line. In order to follow the progress of the flask, a robot with an onboard camera will be used.

This project was designed for pupils of age 16, in their first year of professional college in the field of "Maintenance of Industrial Plants".

The learning objectives of this activity are linked to kinematics. The aim is to have the pupils define basic notions such as trajectories (indifferent, rectilinear and circular) and movements (translation and rotation).

Progress of the teaching sequence: the project was planned over 4 hours, during one day (2 hours in the morning and 2 in the afternoon).

Engagement stage: The teacher presents the problem to be solved to the pupils (they have seen the production line in function before), as well as the Lego NXT kit and programming software. The next hour is spent by the pupils to build the robot with the help of an assembly guideline.

Investigation and Creation stage: The pupils have to retrace the course of the production line "ERMAFLEX" with their robot.

Evaluation stage: The different results from each group of pupils are analysed and shared by the class and a synthesis is done by the teacher and the pupils.

This project has been implemented by two students-teachers of the French "Teachers training institute" in their classroom and was compared to a more classic lesson treating the same subject. The results of the comparison of the two different teaching methods (with or without the help of educational robotics) was presented by the student in a professional report as part of their evaluation as teachers trainees.

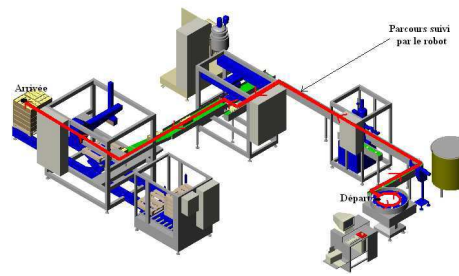


Fig.3: Product line

3.4 Locating and tracking (Romania)

Taking further the idea of describing the phenomena in a suitable natural manner, the robotics become a powerful educational technology. Basically, the robot is a physical model of a living being. Usually, a robot is built to perform some tasks in human like-manner. A lot of things can be discovered and explained using appropriate robotic materials and programs. In our previously reported work [10] we presented the way in which the approach specific to robots intersects fundamental domains and which kind of problems can be approached in the area of fundamental sciences in connection with the specific issues of robotics. Trying to solve any real life problem involves a sum of knowledge from different areas.

Our example is built on one of the most human abilities of the robots: locating and tracking of the objects in their proximity. The estimated time for this project is 6-8 hours. The initial problem is: 'The subject searches for the object. If it is sensed the subject is locating it. The subject decides to track the object in certain condition (for

instance, if this moves on and it is close enough)'. The pedagogical approach in this problem starts with the engagement stage when the teacher exposes the problem, for instance: 'A living being is looking for something. What does the living being has to do?'. The students are quickly involved in the exploration stage and a holistic approach is firstly expected in terms of different disciplines: biology, physics, mathematics, programming, radar technology, etc. The interdisciplinary vision is used to describe the global behavior of the living being. In the investigation stage all these aspects are ordered in terms of the smaller question derived from the general problem. Different groups of students analyze the particular processes, for instance: scanning and sensing the objects, reacting when the object is moving, the strategy of tracking, etc.

The creation stage challenges the students to provide their own solutions and to imagine the functional structures answering to the initial problem. Despite of the fact the proposed subject seems to be simply at a glance it can generate a lot of interesting alternatives for a final solution. For example, different solutions for vision can be chosen, different kinds of displacement could be imagined (continuous, stepping or skipping, etc.), and different strategies of tracking could be programmed too.

Finally, the evaluation stage is a very attractive activity when the students present and argue their solutions and are open to receive feedback from the teacher and from their colleagues. Frequently, exciting ideas and perspectives of development arise in the evaluation stage.

3.5 The cat, the mouse and the master (Greece)

'The cat, the mouse and the master' is a project for introducing basic programming structures of the Lego MINDSTORMS Education NXT programming environment. It was designed and implemented in the Greek teachers' training course. In a previous session, issues on using the Lego MINDSTORMS material, sensors, and on making robotic constructions have been introduced. The estimated time for this project is 6 hours. The scenario refers to a cat moving around looking for mice and changing behavior when meeting its master. A simple robotic construction simulates a cat, whilst the mice are black areas on a flat mock-up. Trainees worked in groups and the project deployed in five stages.

Engagement stage: Initially the mock up is put on the ground, and the groups are invited to make their construction work on it, and adapt it accordingly putting on the appropriate sensors and program it in order to simulate a cat able to identify mice on the mock-up as well as its master when she touches it!

Exploration stage: Trainees are introduced in basic programming statements and structures. Groups undertake three activities that gradually stimulate trainees to explore basic programming statements and structures of varying difficulty and complexity. Each activity poses a specific problem that trainees undertake to solve:

- At first they should make the cat able to run after the mouse and stop when it reaches a black area (the mouse!). To this end the robotic construction should be extended to include the appropriate sensor for example a light sensor, whilst it should be programmed using functions, the loop structure, and blocks.

- Then the cat should be able to stop for a while and make a sound when its master touches it. To this end, the robotic construction should be extended to include the appropriate sensor for example a touch sensor, and the program controlling the robot should be extended to include the condition structure, and statements like Display, Sound, Wait For.
- Lastly, the cat should search for mice in an extended area by moving on a spiral path. Math block and variables are introduced through this sub-activity.

On each activity appropriate worksheets with instructions and information about specific statements and structures of the Lego MINDSTORMS Education NXT programming environment are provided, aiming to enable groups working autonomously.

Investigation stage: The general problem is analysed in specific questions. Each group investigates alternative approaches aiming to develop a comprehensive strategy for the ‘cat’ behaviour. For example, questions that were investigated were about the different strategies that a cat might use in searching for mice, ‘How will the cat stop if it doesn’t meet a mouse? Is this a matter of the mock-up design or the specific construction?’, ‘How the cat will react to different types of obstacles? How does the cat recognize its master?’, ‘What might be a mouse? What if the mouse was a moving construction?’. Moreover, evaluation criteria for the final product are discussed and determined.

Creation stage: Each group adapts the robotic construction(s) and develops the appropriate program for guiding the behaviour of the mice (in case the mouse is also a robotic construction) based on the strategy developed at the Investigation stage.

Evaluation stage: Final products are presented and discussed in plenary session. All alternative solutions are examined and evaluated based on a synthesis of the criteria proposed by each group at the Investigation stage.

3.6 Getting data from the environment: the data logger (Italy-Spain)

When the main objective of a project-based activity is to discover or verify a general law that controls a phenomenon, or to make some statistics on the experiment, one usually needs to collect lot of data from the real world. The manual acquisition of experimental data, though interesting from an educational point of view, is subjected to unavoidable inaccuracies that can compromise the following analysis.

The NXT firmware permits us to use sensors not only for robot controlling purposes but also to get samples from such inputs and to store them onto an internal file, subsequently uploaded to a PC for post-elaborations. One of the basic examples we suggested in the course curriculum, presented for the first time during

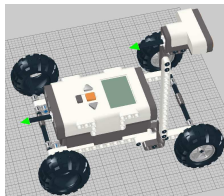


Fig.4. The car

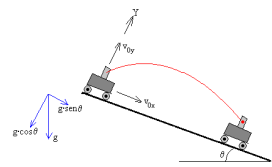


Fig.5. The slope and the acceleration

the training course that took place at Rovereto (Italy), was the so called ‘data logger’ (DL). The goal of this project is the students to study the uniformly accelerated motion and to deduce its fundamental quadratic law between space and time. The estimated time for this project is 3-4 hours. Through the engagement stage students discuss about how to ‘ride a bicycle down a sloping road’.

Because the NXT servo-motors are speed-controlled devices, we decided to use the natural gravity acceleration in order to apply a constant force to a vehicle: therefore during the exploration stage students working with the teacher built a very simple car on four wheels without motors, equipped with a sonar sensor to get space data, leaving the car to move freely on a slope with a constant inclination (Fig. 4 and 5).

The program periodically samples the sonar sensor output about the distance between the vehicle and a fix object, i.e. it sets a timer, opens the data file and then in a cycle waits the timer synchronization, reads the sample from the sonar and writes the time and the sample to the file. The cycle ends when the distance reaches a maximum (the end of the straight path of the car). The recorded ASCII file with the acquired data can be uploaded to the PC using a specific NXT-G function.

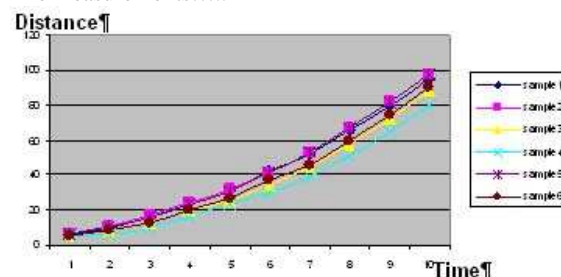
Students, through the investigation stage study the collected data and look for repetitive patterns. Students are promoted to edit the data with appropriate software, construct and study the corresponding distance – time tables and graphs. Also they make calculations and graphs of velocity. One of the most interesting knowledge that students should “discover” is that a physical phenomenon is only partially perfectly repeatable, due to noise errors and other physical inaccuracies (e.g. irregular friction, sensor precision, etc.). The plotting of the results of the repetition of the DL experiment can convince them (Fig. 6).

Optionally, students through the investigation and creation stage, may also investigate the impact that several factors like the wheels, the friction, the angle of slope, the loads, may have on the car motion. They may also study distance/time relation by using appropriate algebraic calculations.

During the evaluation stage the acquired data can be suitably displayed and used for a discussion among the students and the teacher:

- to agree with the evidence of the data with respect to the expected behaviour, trying to find reasonable justifications to possible deviances;

The measurements.....



The theory...

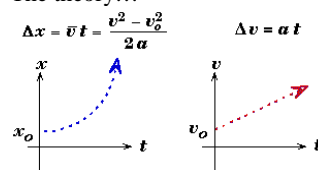


Fig. 6. A distance –time graph for accelerating motion

- to deduce laws, constraints, proofs and intuitions from the shared analysis;
- to make a deeper insight in the physical phenomenon under observation;
- to provide a new awareness which is the basic condition to build new knowledge with a constructivist teaching/learning approach.

The DL example can be used as a prototype to perform attractive, rather complex data acquisition experiments with one sensor and also with more than one sensor. In the latter case the reading of samples might be done as much synchronously as possible to permit correct correlations among the different sensor data. For instance one could study the correspondence between the rotation of a motor, measured through its internal sensor, and the motion of the whole vehicle, measured with the sonar in case of a linear motion, like in DL example, or with a gyroscope or a compass sensor in case of a rotational motion.

4 Conclusion

In this paper we presented examples of educational robotic activities designed within the constructivist approach of teaching and learning. Important aspects of these examples include the way they were organized as projects deployed in different stages, the underpinning teaching model adopted, and the investigating and exploratory tasks involved.

Students work with the target learning concepts undertaking broader projects to work with. Projects should be authentic and presented in a meaningful context. The way students' work is organised in 'working spaces' trigger the expression of students' ideas and the investigation of students' personal questions. The diversity of the learning outcomes of each task involved, aims at the personal engagement of each student in the learning process.

The sequence of tasks in each project promotes the gradual development of freedom in students' initiatives and students' expression. During each project a number of new skills / knowledge are cultivated. This is done mainly through activities that engage students in guided researches and experimentations (exploration stage). The experience gained from these tasks gives shape to new ideas. A further elaboration of ideas takes place during classroom discussions and teacher's intervention. Consolidation of ideas and self expression takes part during open ended tasks where students construct their own products (investigation stage). So the control of the learning process is gradually transferred from the teacher to the students. The problems posed by each activity are gradually transformed from close to open ended. Tasks are initially guided by the teacher but at the end they are controlled by students.

Finally, the social character of each interaction appears to be a very important factor in each project. The social environment is important for the development of individual understanding, for presenting final products and for getting feedback. So in each project cooperation between groups and between members of a group is promoted.

Our intention was to contribute to the dialog about innovative teaching practices within the framework of constructivism. We hope that we have illustrated some useful

examples and pointed out some interesting strategies that can be useful to other practitioners in the education field.

Acknowledgements. This paper was based on work done in the frame of the project “Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods” (TERECOP) funded by the European Programme Socrates/Comenius/Action 2.1, Agreement No 128959-CP-1-2006-1-GR-COMENIUS-C21 2006 – 2518 / 001 – 001 SO2.

References

1. Resnick, M., Silverman, B.: Some reflections on designing construction kits for kids. in Proceeding of the 2005 conference on Interaction design and children, (Boulder, Colorado, June 08-10-2005), pp.117-122 (2005)
2. Alimisis, D., Moro, M., Arlegui, J., Pina, A., Frangou, S., Papanikolaou, K.: Robotics & Constructivism in Education: the TERECOP project, In Ivan Kalas (ed.), Proceedings of the 11th European Logo Conference, 19 - 24 August 2007, Bratislava, Slovakia, Comenius University, ISBN 978-80-89186-20-4 (2007)
3. Brooks, J., Brooks M.: In search of understanding: The case for constructivist classrooms. Alexandria, Virginia: Association for Supervision and Curriculum Development (1993)
4. Drive, R., Bell, B.: Students’ Thinking and the Learning of Science, *School Science Review*, pp.443-456 (1986)
5. Hoover, W.A.: The Practice Implications of Constructivism, *SEDLetter*, Vol. IX, No 3 (1996)
6. Korthagen, F. A. J., Klaassen, C. A. C., Russell, T.: New learning in teacher education. In P. R. J. Simons, J. L. van der Linden, & T. Duffy (Eds.), *New learning*, pp. 243-259. Dordrecht: Kluwer Academic Publishers (2000)
7. Papert, S.: Situating constructionism. In I. Harel, & S. Papert (Eds.), *Constructionism*. Norwood, NJ Ablex Publishing (1991)
8. Puntambakar, S., Kolodner, J. L.: Toward implementing distributed scaffolding: Helping students learn science from design. *Journal of Research in Science Teaching*, 42(2), 185–217 (2005)
9. Resnick M., Ocko, S.: Lego /Logo Learning Through and About Design. In Papert S, Harel I. (ed.) *Constructionism*, pp.141-150. Ablex Publishing Corporation, US (1991)
10. Ionita, S., Ionita A.I.: Robotics between «object» and «educational tool». Steps towards a constructionist methodological approach. In Proceedings of EcoMedia International Conference, 23-24 Nov. 2007, Pitesti, Romania, pp.119-125, ISBN978-973-690-701-2 (2007)

CEABOT: Nationwide Little humanoid robots competition; rules, experiences and new challenges

Alberto Jardón¹, Paloma Zafra¹, Santiago Martínez de la Casa¹, Antonio Giménez¹

¹ Robotics Lab. Universidad Carlos III de Madrid. Avda de la Universidad 30.
28911 Leganés, Madrid. Spain

Abstract. Ceabot competition, first national competition of humanoid robots for degree and postgraduate courses students, is introduced. It is organized by the robotic part and automation Spanish committee (CEA-IFAC). Three editions already took place. Its aim is to encourage students to start with robotics, programming little humanoid robots that were constructed by themselves or adapted from a commercial kit. In the paper the main aspects of the organization, the rules and the competition are revised.

Keywords: Teaching with robotics, Didactic approaches, Humanoid robots.

1 Introduction

In this document we tried to compile and spread the gained experiences from former CEABOT competitions. The competition is annually presented in “Jornadas de Automática” sponsored by CEA-IFAC. The competition final goal is to encourage the automation and robotics teachers to try new students to participate in new editions of that competition.

2 The competition

The objective of the competition is promoting the participation of degree and postgraduate courses students for their starting at robotics, programming and control of walking robots.

Remote control units are not permitted during the execution of competition tests. The robot of the team must demonstrate its skills through accomplishing the various tests fully autonomously. Any intervention by team members during development of the tests is punished with a penalty, even a single touch with the hand to avoid a collapse or change of position to recover it from a strange movement. All hardware and software control should be included in the robot. The robots are going to inhibit communication with the outside world to avoid tele-operation. The behaviour of the robot must be programmed and based on sensory information available on board. The students should have chosen the sensors according to the test.

It consists of two or three tests, the first one in the past calls was a walking test. Robots must walk forward from the starting line until the finishing one, where they must go back to the starting line walking backwards. The second one a sumo fight.

This test is in a plane rectangle, levelled out, green, stiff field which measures are 2x2.5m. It is shown in the following draw, figure 1. In figure 2 the sumo court.

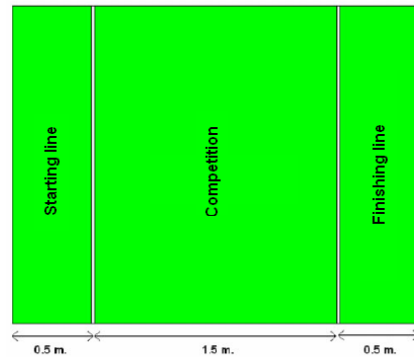


Figure 1. Field for the competition

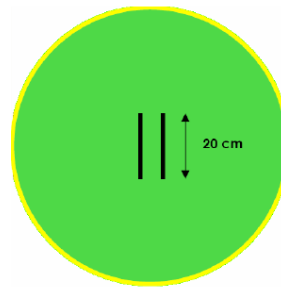


Figure 2. Ring court for the sumo fight

2.1 The rules and teams

The organizing team is the one who writes the rules for the competition. Every year the rules have to be changed, revised and published. For further information on the different tasks the robots must overcome check the competition rules. The rules of this competition are based on the ones from the Federation of International Robot-soccer Association (FIRA), and the competition is based on Humanoid Robot World Cup Soccer Tournament (Hurosoft) with small robots at [1].

Each team is allowed to have maximum one robot. One team is made up of up to five students. A student cannot be in more than one team. It is recommended having teams of two or three people because there is no limit for number of teams. But if there are too many teams participating, the jury would set up a qualification round to make sure only the best ones are taking part in the tournament.

For score points the robot has to make its opponent fall down or expel it from the court. The one who makes more points wins an assault. There are three of them.

3. Students participation

To be able to participate in the competition, each team has to prepare the robot to afford the heats.

At first, the robot has to be built. After this, the sensors have to be chosen and added. Then the software programming, that is the part where most time is spent, has to be done. It consists of generating trajectories and designing primitives for the movements.

The robot's weight, dimension and anthropomorphic characteristics are described in the rules. How the walls, the marks and colours must be has to be written there as well. The teams, the jury, and everything about the heats and scoring have to be revised every year. A penalty can be defined in case anyone does not play with the rules, even the exclusion of a team from the competition may be declared.

4. About the organization

The competition is an extra activity in the “Jornadas de Automática” organized every year by the Public Universities of Spain that are at GT-ROB. The Host University organizes the whole event. GT-ROB and the teacher of last year’s Winner University support them. The development committee consists of people selected by the last year’s winner.

The Host must be in charge of the construction of the courts and the development committee of its verification and examination.

The participant teams always have to be accompanied by a teacher who is responsible for the students. The registration is for free and the sensors, the robot and every material are paid by the respective University. This limits the number of teams per University.

To encourage the participation in the competition it is important to find a technologic company which is interested in sponsoring the prizes.

5. Beyond contest

The robotics as a teaching tool wins over more followers every day, from secondary education to university courses across the degrees, masters, etc. Not only engineering departments are using robotic platforms but it is increasingly used in other fields. This suggests that the potential of robots in education is developing. Advantage should be taken on the variety of robotics and low cost robot kits. (Weinberg et al. 2003).

The robots provide the students a natural way to develop skills in integrating systems, the current functioning of the devices, critical thinking and independent resolution of problems, working in teams and multidisciplinary approach. Many teachers are interested in these low-cost reconfigurable and/or mobile platforms for teaching purposes or research. Universities and high school centres are using these kits as platforms to test ambient intelligence, programming control systems, mechatronics etc. Numerous projects have dealt with the impact of robotics in education ([2], [3]). This demonstrates that the motivation to learn is substantially increased, when it gives students a practical way to implement the theoretical foundations, building and programming robots to solve certain evidence or real problems.

The interaction between the students and real robots provides them with enough experience to understand and solve real problems, developing new capacities to identify and propose viable technical and economical solutions. Compared to other disciplines, robotics is still an emerging area, which combines aspects of mechanical engineering, electronics and programming applied in a particular device. This multidisciplinary approach and the synergies that help them to work in teams of students from different engineering directions make Robots an excellent teaching platform.



Figure 3., CEABOT07.

Figure 3 shows the competition of 2007. The experience gained by students makes them able to focus on drafting a report about the work done and explaining how the, operating circuits, sensors, etc. work.

6 Conclusions and Future Work

After the experience that has been made by the organizers of the contest in its first three editions, some possible conclusions and initiatives can be drawn. To increase the competitions acceptance and impact among the students, it has taken time to set and readjust the rules. Furthermore, enough time needs to be given to the students to be prepared for the tests. It has to enhance disclosure and advertisements to increase participation. We have to involve teachers who hold seminars about subjects like robotics, programming and its automatic bid to incorporate ending of career projects opportunity to participate. In turn departments must be able to finance the purchase of kits and consumables. This is facilitated through the use of robots built by students themselves and low cost commercial kits.

Thanks

We want to specially thank, for the efforts made to all the UC3M students that collaborate in the last three contest edition on behalf of the team Roboticslab.. Thanks to the CEA-GTRob and CEA_IFAC organizations for supporting the initiative contest each call.

References

- [1] Web site, accedida Marzo 08:
<http://www.fira.net/soccer/hurosot/overview.html>
- [2] M Ceccarelli, "Robotic Teachers' Assistants", IEEE Robotics & Automation Magazine, vol. 10, no. 3, pp.37 - 45. September 2003
- [3] L. Greenwald, J. Kopena, "Mobile Robot Labs", IEEE Robotics & Automation Magazine, vol. 10, no. 2, pp.25 - 32. June 2003

Algorithm, Pseudo-Code and Lego Mindstorms Programming

Anthi Karatrantou¹, Chris Panagiotakopoulos²

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

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

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).

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

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.

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'*.

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.

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'*.

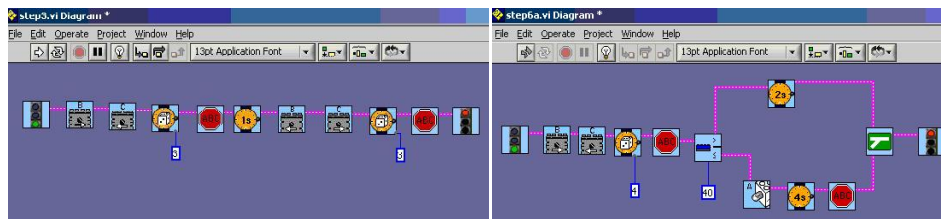


Fig. 2. Characteristic programs in Robolab of the subjects' work (from steps 3 and 6)

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

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.

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. Fassouloupoulos, 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)

Semantic and epistemological continuity in educational robots' programming languages

Emanuele Micheli, Marco Avidano, Fiorella Operto¹

¹ Scuola di Robotica,
P.O. Box 4124, p.zza Monastero, 4, 16149 Genova, Italy
{micheli, avidano, operto}@scuoladirobotica.it

Abstract. The object of this paper is to analyze some new open-source software for the programming of educational robotic kits which can accompany the student from pre-school to high school. The authors propose the development of a learning environment which operates on two levels: the physical level, with the planning and construction of the robot; and the abstract level which is linked to the programming. In our experience of educational robotics, the personalization of the robotic artefact is an important factor in order to achieve success. There are few possible types of personalization for the program and the current trend is that of standardizing the language. The approach that we propose is that of a language that can easily be personalised. We are working on designing and uploading a “converter icon-code” on the Lego NXT robotic kit which could be used by students aged from 5-6 years old, to those in high school.

Keywords: educational robotics, open source, icon language

1 Introduction

The increasing availability of robotic kits used for educational robotics from pre school to high school, demonstrates the interest in and the usefulness of these technological teaching methods, both in curriculum subjects and to increase the students' technical and scientific abilities.

The problem that the authors have noticed during several years of national and European projects in Educational Robotics is that there exists a gap and a discrepancy between the substance of that which is communicated and learnt through educational robotics, and the different pieces of software that the robots themselves use.

This issue concerns one of the fundamental aspects developed by S. Papert as it is at the origin of the students' increased learning abilities, and therefore of the artificial learning environment. According to this view, the acquisition of knowledge is no longer conceived via the unique way learning and gaining knowledge as thought in the traditional learning school, but rather there are as many ways of doing this as there are expressive capabilities among the students, working with the given medium. Papert called this environment microworlds: whilst prior to school years everyone develops linguistically within their own cultural environment without particular

difficulties, however, in a formal situation, not everyone is capable of learning new skills [1]. In this former case, 'learning' differs from the natural way understanding.

The advantages related to the continuing the learning process by using our natural way of understanding are, to some extent, reduced in the field of educational robotics due to the lack of software capable of being used by students from pre school to high school ages. This is the case even though there is a common code for all information technology which is made up of the general algorithms which are at the base of all types of programming environment. The creation of algorithms makes it possible for the student to solve scientific problems (mathematical, physical, logical, technological) in the best way possible.

Educational Robotics allows the students to actively and enthusiastically apply themselves when solving scientific problems. Students become better at solving any scientific problem thanks to the programming in robots. But following international projects Robodidactics and Roberta [9], many teachers have noted the difficulty that using different language to solve the same problems presents. The crux is that there is a risk that the student will be tied by the technical specificities of the language used and will not be able to find a pattern in the more complex languages.

The language change presents a stumbling block for the students, and the challenge is to render this transition as linear and logical as possible. What is needed is a transition which allows the student to understand that behind all forms of language found in software there exists a common algorithm. If it were possible to find a constant technique used by the students in their curriculum, this technique would allow them to understand the origin of the algorithm and not just allow them to master the language.

2 Robopal, Lego WeDo e Roberta

Due to the school's requirements, teachers often focus on the language used and do not place enough emphasis on the importance of the creation of a general algorithm, which can be developed into a 'human' language understandable by all. This problem has also been highlighted by the University of Amsterdam's projects [2]. The same university has developed an iconic software which is capable of translating the icons chosen by the student into a Java script. This characteristic, which is also highlighted in the European Robodidactics project, has improved the students' abilities of deduction and their ability to not be limited by the language used. The software used and developed by the university of Amsterdam is ROBOPAL. In the iconic software used by Lego there are no 'translation' programs, but these kits are the only ones capable, as observed in the Roberta [4] and [Robot@Scuola](#) [5] projects, are the ones capable of being used by the student in both pre school and high school. The software is not, on the other hand, the same for all ages, as it is too complicated for pre school children and too simple for high school students. At the moment, converters which convert directly from iconic to code language do not exist for the Lego kits. The University of Amsterdam is developing software capable of completing this conversion.

Lego itself has, in its market projects, highlighted this continuity problem by introducing a new kit (WeDo) and a new language which is more easily understood by pre school children (Robolab 2.9).

The authors have worked on a normative pathway and a study on the importance of having knowledge of the different forms of programming language. They point out the importance of identifying a continuous path during the transition between the various languages which allow for the programming of robots. The first step needed in order to understand, and to be aware of the existence of, different languages, is the creation of a personal, personalised language.

In Roberta project, the personalisation of the robotic artefact has allowed the female students to develop the robot more quickly, and to face scientific technological issues with more interest, passion and enthusiasm. Today, there are still no didactic normative paths which provide for the development of personalised program languages.

In the pathway that we present, the first part of the introduction to robotics and to programming is distinguished by the possibility for the children to personalise software commands found in the robot, thus rendering the language used unique and personal to them.

After this stage, pre school children will be able to compare the different solutions found and will be able to share the different languages developed. This will enable them to appreciate the need for a common, standard language, which they will develop in early on in high school.

In high school, the pathway plans for a critical phase: the transformation of iconic language into code language through the use of new software which is inspired by Robopal but which is compatible with Lego. The pathway which we will offer will make use of software which is compatible with the Lego NXT kit, so as to allow the transition from iconic language to that of C++, which is widely used in Italian technical institutions. This software has not yet been developed.

2.1. Personalizing the language

Out of authors' experiences, it was noted that difficulties were encountered when using the latest programming software for the NXT LEGO kit with children from primary schools[3], whilst there was much less difficulty with the previous product linked to the RCX. This first observation raises the need to produce new programming software for the NXT LEGO kit, software which is capable of adapting to the skills of the user. The Staff at School of Robotics, therefore, is working to meet this objective. The first step in the creation of new software will be to modify the icons of the NXT software with the "My Bloc" function [Boogaarts et al., 2006].

From this point a program will be produced which will be capable of being managed on a free, open-source operating system capable of linking up an online community, and which can easily be shared and personalized [7]. Indeed, the software will be able to be modified on two different levels: on a high level, where which the teachers will be able to modify the source code and on a low level where there will be a personalization which is simpler at a graphic and macro level.

The students will be able to personalize their own icons, getting them to correspond to their own language, and create macro actions. In this way, the program will become a personalized product to be shared with others. The teachers will easily be able to create blocks of commands capable of meeting the teaching needs and share these new blocks with other teachers. Immediately after the sharing there will be a convergence towards the standard iconic language. In high schools, the program must allow a progressive transition towards the discovery of the lines of the code and therefore each iconic instruction will be translated into some lines of the code which are easily identifiable by the student.

This concept is certainly not new in open-source software, however what is new is the application of an environment which is totally modifiable and capable of programming educational robots. There are numerous free and open source experimentations, such as Alice[8], which encourages the use of programming in a virtual environment. Programming a robot enables the student to understand the concepts of acquisition of reading data which in a simulation occurs less evidently. Thanks to the robotic implementation, the actions of programming will have consequences in the real world. This acting on various levels (abstract, physical) enables the involvement of the so-called diverse intelligences capable of being recognized during the various phases within an educational robotics project[7]. The creation of a personal programming language which then converges in an official iconic language, will in turn converge into a code enabling students in the coming years to discover diverse programming systems (iconic and coded) accompanied by an instrument which will guarantee the continuity of the discovery as well as ensuring that the wish to discover will continue. Indeed, youngsters frequently distance themselves from software when they believe that they have exhausted its potential; a multi-form software which is modular and personalizable will enable this waste to be avoided.

3 A software and a methodology applicable from pre-school to high school

The modelled software - which is the subject of this paper - does not concern only the physical level, but also the theoretical one. We have designed our software to be employed as a continuous educational tool from pre-school to high schools. At the same time, we have also taken into account that the methodology has to change from one level to the other (from primary to secondary level). In fact, Lego Engineering is working on a similar project, that is to design a single software program to be employed on Lego Minstorms robotics kit from primary and secondary levels, to the graduate levels[11]. In 2006, following the release of the robotics kit NXT (the revised version of the former RCX), Lego introduced new software - extensively based on LabVIEW – which was correctly considered to be the logical consequence of their experiences(fig.1).

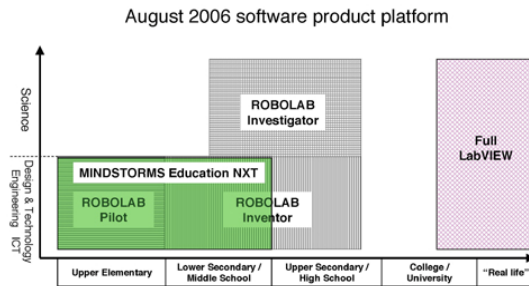


Fig. 1. Lego software production following the release of the NXT [11]

On this subject, there is an interesting projection, appearing on Lego Engineering's website, which shows that they have planned to design a single software to be used from primary to junior high school, while they have devised a different and more articulated one for senior high school students. Furthermore, with the coming release onto the market of the WeDo robotic kit - expected by January 2009 - it looks like Lego had also planned to enter the market of educational robotic kits for primary school (age 7-11)[12](fig.2).

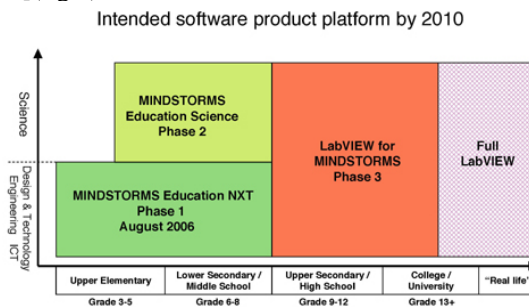


Fig. 2. The forecasted Lego's software's production by 2010 [11].

A team of engineers and programmers of the School of Robotics - the association to which we are affiliated - has worked out a new solution - to be applied on NXT robotic kit - which seems novel compared to Lego educational product just mentioned, WeDo. We are working on a solution which focuses greater attention on the starting up of the "students" (that is, the pre-schools) in educational robotics through the designing and uploading onto the NXT kit a "converter icon-code" which could be used by students from 5-6 years of age to high school. Here the concept in point is continuity of learning and reasoning (fig. 3.).

This solution fits perfectly within the path devised by Lego: it represents an educational improvement, which also has its own philosophy.

Here below a table of a likely educational progression, where educational robotics have been employed as a tool for teaching programming languages.

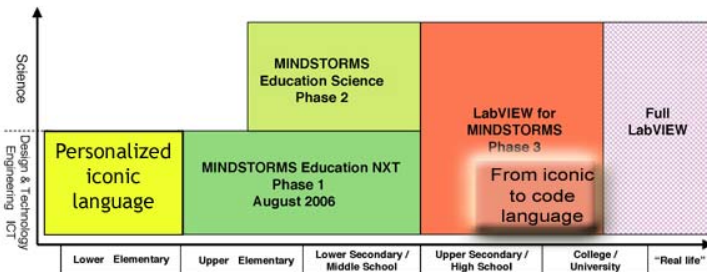


Fig. 3. Author's proposal: a "converter icon-code" which could be used by students from 5-6 years of age to high school's.

3.1 From 5 to 8 years: introduction to programming

In order to introduce the concept of programming a robot, a true and real human simulation will be proposed. This method used in the laboratories organized by the School of Robotics represents a relatively unobtrusive method for helping children to understand programming. The first stage is only oral, each child must give vocal commands to their classmate (who simulates a robot). The second stage is linked to drawing the oral commands. In this way the children create real and true icons which the teacher can use in the program thanks to a simple scanning of the drawings. In so doing the children can see their own works controlling the robot, built earlier by the teacher. In this environment we thus assist in a personalization which allows a simplification to the introduction of a standard, common language as a programming language may be. Furthermore the children can see their own works controlling the robot and thus associate drawing with a subsequent action (of the real robot). The teacher will be able to share on the online platform the icons of the children and discover those of other students. The teacher can thus discover the multiplicity of the language corresponding to a common action. At this stage the programming becomes confused with the narration. Both verbal and graphic narration capable of describing the actions of a robot.

3.2 From 9 to 10 years of age

In this range of age, the teacher should promote more the students' activity of assembling the kit than the programming. The kids will combine together the NXT robotic kit on the basis of the standard models proposed by Lego Manuals. Next, the students could personalize these models, like in the case of *Roberta*, the European project devoted to the promotion of robotics among girls (in which there are personalized models of robots done by girls).

At this school level, we advise the teacher to employ the iconic language to programming the kit. The teacher could draw on ideas from the libraries developed by School of Robotics, or by his/her associates(Fig.4). In the first instance, these libraries

will be set up using icons similar to Robolab's, which have been already used successfully in many primary school cases[13]. In this way, the teacher adapts the version of Lego NXT Education to the needs and specificities of his/her students. The programs written in this context have to be simple, with little use of the information from robotic sensors.

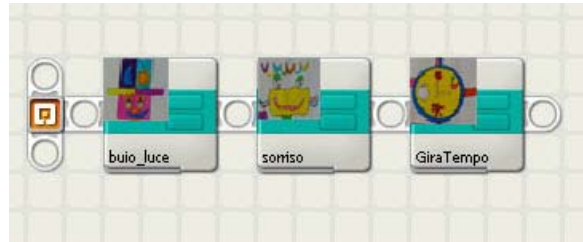


Fig. 4 Personalization of NXT-G's iconic language

3.3 11-13 years of age

In this phase, kids have already learned and managed the programming logic, and also a simplified version of the flux diagrams. Now they should be invited to program their kit using the information from sensors. Here the teacher can introduce constructively (with the hands-on method) the concept of action-reaction which in the previous years was only hinted at, but not formalized. In this phase the teaching shifts from a student-centered software (which was designed by the teacher) to a standard language which is the iconic language proposed by Lego, with no distinctive feature. At the end of this phase of program learning, the teacher will invite the students to re-process and re-design the programs, and also the robots' assembling. He/she will adapt this further step to the features of his/her class of students. Then the teacher will introduce the concepts of subroutine, and of the macro to be retrieved. The students will personalize their robots and also the language program, for instance, drawing new icons. The teacher will suggest the students to overcome programming by trial-and-error, previously designing their program on paper, drawing the program with self imagined flux diagrams, and then designing the software on their pc.

3.4 14-17 years of age

At this phase, the teacher will invite the students to formalize the program previously written on paper with the help of simple flux diagrams, or algorithms. In fact, it is important for the students to start writing algorithms abandoning the iconic language and using words, which is the first step towards learning program codes.

At this point our converter can be usefully used – a shifter from iconic to code lines. With this, the student should acknowledge that, modifying the icon's control

parameters, the line code changes accordingly. In so doing, the student will easily learn to shifting from the iconic to the code programs.

Teachers and students could upload their products on the platform Robot@Scuola (organized and managed by Scuola di Robotica) to share their instructions with other students all over Italy. Following a phase of training, shifting from iconic to code languages, the students will easily and definitely get on to the code programming.

There exists a similar project to that proposed which regards the european project “Robodidactics”, that the authors have participated in, which provides for the use of the ROBOPAL software, developed by the University of Amsterdam. The software which is compatible with the Robotech robotic kits which contain a MUVIUM microchip is capable of managing the conversion from Robopal’s iconic language to that of a Java code. Today, a similar converter for NXT-G software is being developed.

4 Online programming: sharing experiences

The only road for growth is that of comparison. We learning by copying. Every mind, every intelligence, in order to be able to better develop and express its own capacities needs to be nourished by a fertile environment. The sharing of experiences and the comparison of different thoughts are essential elements for pushing each of us to reach our maximum potential: the level which may be attained can potentially go well beyond what can be predicted by even an in depth analysis of the capacities of the single subject. Each of us has different capacities for synthesis, analysis, study of the elements of departure and the routes which can be taken: “complete people” who are capable of reaching the maximum level in each part are very rare. A free and open environment without communication barriers or barriers to the sharing of ideas and information is essential for reaching our maximum potential and giving each of us the possibility to express our own capacities and potentialities better: by using the method of comparison, collective results which are greatly superior to the simple sum of the results achievable by the individual separate components can be obtained.

Clearly, the best result is obtainable by using a direct comparison of the parts: by taking advantage of the internet’s potential and of the communication tools made available by the net, such as forums, chat rooms and blogs, it is possible to obtain great results with minimum cost.[14].

The software which is proposed to be developed will unite the positive elements of each aspect of the network with regard to the communication and sharing of ideas: the projects created by each individual school or student will be available to share with the entire community in order to obtain comments and suggestions and in order to serve as a stimulus for both the creator (“I want to show what I am capable of doing”) and the visitors (“if he did that, I want to do better”) to do their best in a live environment which allows for sharing and competing.

The software will allow for the creation of projects and will provide a simple and immediate way of sharing them: frequently the sharing of projects is hindered by the difficulty of publication, where the additional effort of making the project presentable

in online blocks works only at the start, thus impeding the growth of high quality projects. Making the sharing of the project immediate stimulates the communication and the sharing of ideas: each person will be given the possibility to express their own capacities in the best way possible. This will apply both to students with strong imaginations and initiatives who propose innovative ideas and new objectives, as well as to students with less imagination but nevertheless highly capable of resolving problems and questions relative to the development of the project.

In summary, the key elements of the software will be: an open environment in which the students will be able to present their own projects and ideas, and to give and receive comments for improving these easily.

5 Personalizing the programming: involving girls

The personalization of the students' own programs, just like the personalization of the single robot, makes the products conceived by them unique. Thanks to the experience of the project "Roberta" in which the involvement of the girls in the study of the scientific-technological materials is strongly supported through the use of the robotic kits, it was noticed how fundamental the emotive aspect- the link between the artefact and the student- is. At the construction stage of the robot indicated in the "Roberta" manuals, the personalization of it is a formalized stage.

Numerous studies [15] demonstrate how the girls suffer a strong separation at the programming stage. A few microworld (EX Robotic Microworlds) projects provide for a possible personalization (for example of the character), but we have not found projects allowing for the personalization of the iconic code. We believe that this personalization can help the girls and in general all students to see the program as their own product and not as a series of instructions in a list. Obviously, this educational step must not induce the belief that a myriad of "personal" software is wanted, but must ensure that the personalization of the software enables the student to better understand the subsequent necessities of standardizing the programming languages.

6 Future prospects

The software project and the teaching routes relative to the programming proposed in this article have not yet been experimented with. As soon as schools resume, some primary, mid-school and high school teachers will be involved in various of the stages. The first stage of the project foresees the conversion of the NXT icons into those of the old robolab system. This conversion will enable the teaching staff to learn how to personalize the software in order to make it clear for their own students. In the meantime, the School of Robotics will work on the creation of the conversion software.

References

1. Papert, S.: "The children's Machine: Rethinking School in the Age of the Computer"(1993)
2. Hartmann S., Wiesner H., Wiesner-Steiner, A., Robotics and Gender: The Use of Robotics for the Empowerment of Girls in the Classroom, in Gender Designs IT, Springer Verlag, 2007
3. van Lith, P., Teaching Robotics in Primary and Secondary schools, Proceedings, ComLab International Conference 2007, Computerised laboratory in science and technology education, November 30 - December 1, 2007 Radovljica, Slovenia
4. Reichel, M., Wiesner-Steiner, A., "Gender Inscriptions in Robotic Courses" DCML (2006)
5. Web Platform on Educational Robotics in Italy -Robot@Scuola, <http://www.scuoladirobotica.it/retemiur>
6. Boogaarts, M., Daudelin, J.A., Davis,B.L.,Kelly,J., Levy,D., Morris,L., Rhodes,F., Rhodes,R.,Scholz,M.P., Smith,C.R., Torok,R.: The Lego Mindstorms NXT idea book. NO Starch Press, (2007).
7. Carrington, D., and Kim, S.K. Teaching software design with open source software. ASEE/IEE Frontiers in Education Conference, (Nov. 2003).
8. Gross, P., Powers, K.: "Evaluating Assessments of Novice Programming Environments" ICER'05, October 1-2,Seattle, Washington (2005)
9. Roberta, Girls discover robots, <http://www.iais.fraunhofer.de/3845.html>
10. Dewar., R., Schonberg, E. :“ Computer Science Education: Where Are the Software Engineers of Tomorrow?” Cross Talk, (2008)
11. O'Hara, Keith J., Kay, Jennifer, S. "Investigating open source software and educational robotics", Journal of Computing Sciences in Colleges, 2003
12. Lego Engineering, <http://www.legoengineering.com>
13. Press release on Lego WeDo, http://www.lego.com/education/download/FinalWeDoLaunchPressRelease063008_2.pdf
14. Erwin, B., Cyr, M. and Rogers C.: "LEGO Engineer and RoboLab: Teaching Engineering with LabVIEW from Kindergarten to Graduate School" (1999)
15. Bianchetti,P. Giannini,L., Mazzei, D., Merlo, D., Micheli, E., Vello, B.: "Rob&ide – Storia di Robot e Androide" Didamatica , Cesena, (2007)
16. Adams, J.C.:“Alice, Middle Schoolers & The Imaginary Worlds Camps” SIGCSE'07, March 7-10, Covington, Kentucky, USA (2007)

A Piedmont SchoolNet for a K-12 Mini-Robots Programming Project: Experiences in Primary Schools

M. Stella De Michele¹, G. Barbara Demo², Simonetta Siega³

¹ I Circolo Didattico Settimo Torinese, v. Buonarroti 8,
10036 SettimoTorinese, Italy primocircolo@comune.settimo-torinese.to.it

² Dipartimento Informatica, University of Turin, c.so Svizzera 185,
10149 Turin, Italy barbara@di.unito.it

³ Istituto Comprensivo "Fogazzaro,, di Baveno (VB), via Brera 12,
28831 Baveno - Verbania, Italy simo.si@alice.it

Abstract. PIONEER (PiedmOnt NEt for Educational Robotics) is a schoolnet for K-12 "Educational use of robotics" project originated and carried out by primary school teachers and headmasters. Its goal is to promote Papert's constructionism in a cooperative environment for setting up a model of mini-robot programming experiences in support to the standard curricula covered in school years K-12. Here we concentrate on primary school activities where educational aspects concerned by using small robots fill a long list. In this list there is of course mathematics, but also education to affectivity, creativity, communication, geography, and other. Experiences from our project are described.

Keywords: cross-disciplinary activities, inquiry based teaching technique, pupil centered teaching.

1. Introduction.

In July 2007 a group of Italian primary and secondary school headmasters signed the agreement "Net for the educational use of robotics" aiming to make use of mini-robot programming to carry out activities of mutual interest in their schools. The project is also called PIONEER (PiedmOnt NEt for Educational Robotics) since the concerned schools are scattered through the Piedmont region. The First Teaching District of Beinasco (Turin), with its headmaster V. Termini, was chosen as the leading institute, and the teacher S. Siega as the educational manager. The net also relied on the cooperation of G. Marciandò, who led the *Robotica Laboratory* of the Regional Institute for Researches in Education (IRRE), and of G.B. Demo from the Dipartimento of Informatica of the University of Turin.

PIONEER aimed at promoting Papert's constructionism in a cooperative environment for setting up a model of mini-robot programming experiences in support to the standard curricula covered during the K-12 school years [1]. All the educators who are members of the net had already been involved in ICT projects in different times and kinds of activities. In particular, most of them had been cooperating with G. Marciandò in his *Robotica Laboratory* activities promoted by

Piedmont IRRE. This institute was going to change its mission in summer 2008. Thus the idea of connecting several schools in a network had administrative and financial reasons, but also, and most importantly, educational goals primarily originated from teachers working in the field. They selected a schoolnet organization in order to gather experiences from different institutions and to create both a shared pedagogical environment and a common professional guidance. This conceptual change in school organization was deemed very important particularly in a situation where the administrative rules and the educational guidelines are often changed. The common environment is likely to provide a greater stability.

The educational researchers grouped in the net had already shared, in their previous activities, the belief that they can fruitfully take advantage of their common cultural background based on psychology and pedagogy [2, 3] to meet the current technology challenges. This mingling between tradition and innovation has given rise to a project for an original educational methodology where technology is used in order to offer children the pleasure to learn every subject "beyond the pencil and the book" [1]. In the drafts of a PIONEER Technical Group meeting we read that the net aims at "developing, documenting, evaluating and disseminating K-12 mini-robot-based educational activities that must be concrete, feasible and strongly affecting the children daily curriculum, following Marcianò's idea of robotics as a learning environment" [4]. Teachers also wished an experience exposing pupils to the method during several years of their education. Thus a K-12 project was decided where robots should be used with continuity rather than in occasional laboratory hours. Though some junior and senior secondary schools are also involved, most PIONEER experiences up to now concern kindergarten and primary schools, probably because primary school teachers are most accustomed to cross-disciplinary activities, and because innovative methods of teaching standard subjects are considered more successful if applied from the very beginning of the children school life.

As said above, several members had already been involved in activities connected with mini-robot programming before the net was set up. To give an idea of these early experiences, in Section 2 the teacher S. Siega sketches activities in a fourth-grade class in Baveno primary school during the year 2003/2004 when a single Lego RCX robot was used. These can be considered the first net experiences because S. Siega currently is the PIONEER pedagogical manager. Sections 3 and 4 concern recent activities. In Section 3 M. S. De Michele describes her 2007/2008 experiences in a second-grade class with the Bee-Bot, by the TTS-group, programmable by pressing buttons on its back. Several teachers in PIONEER schools have used the Bee-Bot. For lack of space we sketch here only De Michele's activity, which is interesting because she was novice to programmable robots. Her experience can be useful to teachers envisaging to approach robotics with their pupils of the lower grades, and can inspire confidence that good results are achievable when pupils and teachers learn together. Section 4 is a short overview of recent activities where students write programs. From about the beginning, PIONEER schools have used different types of robots and programming languages. Among programming languages used to program the RCX Lego robot, Siega and her schoolchildren in 2004 began to use the NQC (Not Quite C) textual language, proposed by D. Baum [5]. Most pupils found using iconic languages less clear than using the textual NQC, particularly when icons have to be connected in a behavior description. As for the teachers, they observed that using the

same textual format both in programming and in natural language reading and writing allows interesting exchanges between the linguistic competences and those needed to conceive and develop robot programs [6], [7]. Thus G. Marciandò, wishing to have a children-oriented, easy-to-use robot programming language, defined the textual language NQCBaby, which is a Logo-like language following the mini-language approach [8]. NQCBaby is briefly described in Section 5, where also a short description is given of the software tools developed around it for a better use by pupils and teachers.

Future directions of PIONEER work are given in the conclusive Section 6.

2. Early experiences: from 2003/2004 to the net.

As we have written in the Introduction, the teacher S. Siega is the current pedagogical coordinator of the network of Piedmont schools involved in the educational use of robotics. Since 2003 she began to program one RCX Lego Mindstorm in a fourth grade primary class, after having worked with her pupils using Microworld software and the Logo language. The pupils criticized both the RCX manual, which presents a too limited variety of examples, and the programming language, which was found to be not enough user-friendly. Pupils also said that the "robot" concept should apply not only to an object built using Lego bricks, but to any programmable, autonomous and mobile object. Due to this observation the awareness arose that by using different kits a larger number of children, belonging to different ranges of age, could be involved in robot activities. This is the important result that the schoolnet today can be proud of having achieved.

After the 2003/2004 single-class experience, G. Marciandò proposed the project "Educational use of Robotics" for the three school years 2004-2007. Three schools agreed with his plan: Siega's Istituto Comprensivo of Baveno, the Direzione didattica of Tortona and the Istituto tecnico of Novara. The latter is a senior secondary school. The project has made possible to study and, above all, to test the idea that robotics in school should be regarded as a subject pertaining not as much to the "new technologies" area, rather to the "new possible teaching methods" in a school-laboratory, i.e., a school environment where to "learn how to learn".

The first experiences were often initiated almost by chance, but they were quickly consolidated owing to the children's greatly positive response. Scientific measures of possible recognition and validation of educational applications have been proposed and documented [9]. In the meanwhile, the NQCBaby language was developed as a new instrument specifically designed for an educational use of robots in the school.

After three years, the natural evolution of the IRRE project was the creation of the network of Piedmont schools to which this paper refers, because of the spreading of good practices produced in nearby schools. The network shares in its work the realization of what S. Papert wrote: "The child programs the computer and, in doing so, both acquires a sense of mastery over a piece of the most modern and powerful technology and establishes an intimate contact with some of the deepest ideas from science, from mathematics, and from the art of intellectual model building. ... Programming a computer means nothing more or less than communicating to it in a

language that it and the human user both *understand*. And learning languages is one of the things children do best", from the Introduction of [1].

The use of different languages enables schoolchildren to communicate with different robots. If a pupil likes better to use icons, she/he may use them rather than a textual language: what matters is the concept of programming. Children enter commands to a robot and then check if the robot performs the intended action. The immediate feedback allows them to understand if they have done a good job OR IF they have made an ERROR. In this case they can correct and change the action of the robot immediately!

Practicing a method of learning by doing is a peculiarity of the PIONEER network of schools. This allows pupils to understand what they are doing rather than to learn mostly by heart. "When a student learns something in school, the most important thing is not the content, but the method of learning, which can be applied again in the future"¹.

3 First programming activities using the Bee-Bot.

The Bee-Bot, produced by the TTS group, is a big bee that can be programmed by pressing buttons on its back for moving forward, backward, turning left, right, starting to move or deleting previous commands. As we have written, several teachers in PIONEER schools have carried out activities with the Bee-Bot. Here we recall fragments from the report that M. Stella De Michele wrote to document the activities that she, new to robots, carried out with her second grade schoolchildren during the last (2007-2008) school year. M. Stella is specialized in teaching humanities, but in 2007 she promptly agreed to become in charge of the robots experiences in her school and to use the Bee-Bot with her seven-year-old second-grade pupils, so as to start learning with them how to program mini-robots and how to use them for standard curriculum teaching.

"I think it necessary that school confronts with the technology to which children are exposed in everyday life. I had used computers for some years with my classes, but I was curious to use an object that can move around following your description of a path, given either by writing a textual description or by pressing buttons as in the Bee-Bot case.

Our story with robots began when schoolchildren found one Bee-Bot on our classroom windowsill. We tried to understand why this bee, different from those we are used to, was there. Possibly she had got lost because of the pollution and had come into our classroom to rest. The bee was greeted, given a nickname (Maya), and the children introduced themselves. They soon found out that by pressing the buttons on its back they could teach it how to move on the floor (i.e. in a two dimensional space): going straight or turning left or right exactly of a **quarter of a cake** (second grade pupils have not yet dealt with angles and their measures). We discovered that the bee could **stroll around** the classroom by pressing more buttons in a sequence and then the **go** button. When a child asked whether we could make the bee go from one child to another, i.e., from a starting point to an end point, some of the classmates

¹ [16], page. 3.

observed that buttons should not be pushed randomly, as they had been doing when they wanted the bee go strolling on the floor.

Making the Bee-Bot go from one child to another requires children to take decisions: first we must decide where to go from where, i.e. design a path connecting two points. Different children may suggest different paths. We take some of them into consideration, and for each path we decide which buttons to press and how many times. Then we verify if the Bee-Bot moves the way we want. If it does not, that means that we have given the bee the wrong teaching, and in order to change its behaviour we have to modify, by successive adjustments, the sequence of buttons to be pressed. If we want to teach the bee a wholly new behavior, we have to take some time in *planning* exactly what we want the Bee-Bot to do.

We have to be precise and discover how far the bee moves at each step and so on. Thus we introduced the concept of **measure**: if Maya has moved for a while, how can we tell how far she went? How do we measure the distance covered? First we used several non-conventional tools, then we chose the ruler, because it is a common tool and gives a number for the quantity of space covered at each step. To determine how far the bee goes with a given number of button pressings, one child suggests the arithmetical operation of adding (the length of one step to the previous ones), another suggests multiplying the number of steps times the space covered by the single step). Thus the teacher recalls that both are right because product is defined by means of the sum, and a child shouts: «Teacher, is this robotics or math? ». Children drew the paths on their exercise books with squared sheets, and at this point the introduction of the Cartesian plane, suggested by some of them, turned out to be perfectly natural."

After the experience of one year we are not proposing here a generalization. The above activity report is an excerpt of a class journal, which we will use to compare and discuss our experience with the ones of other PIONEER colleagues with lower-grade classes. Though we have not yet performed a specific evaluation of children's achievements, we can compare the abilities acquired by them with those of all the other pupils in the same age we had in over twenty years of teaching. We notice that, by using a Bee-Bot, lower-grade pupils develop skills for:

- counting and logical thinking;
- solving topological problems;
- accessing problem-solving education;
- getting used to an *inquiry-based learning (and teaching) technique* even in activities, as those described above, perceived as close to mathematics. This is an uncommon experience in lower grades [10].

In addition, we perceive that pupils have a playful approach to robotics and begin to understand what programming a robot is. We are planning in our school an evaluation session adapted from the one described by Kurebayashi for older students [11].

It is important to point out that the above activities naturally involved several educational aspects other than the more obvious ones concerning mathematics. For example, we considered different reasons why the bee had come into our classroom. The environment pollution was considered an acceptable reason, and children all together wrote the "Bee-Bot Story". Moreover, different forms of pollution, causes, consequences and remedies were discussed: thus some environment-preserving education has been covered. Pupils introduced themselves to the bee, gave it their welcome while holding it in their hands, gave it a name, involved it in their school life

by showing concern for the new "thing": this is Education to affectivity and to diversity. For each robot session we had a discussion time followed by a self-activity where every child wrote down a few lines on what we had done. Children learned by doing activities with a concrete object and teachers learned with them.

4 Primary school programming languages

The current PIONEER methodology includes the use of four different kinds of robot kits, with different features and functions that allow different kinds of learning: the Bee-Bot, the Scribbler by Parallax, RCX and NXT by Lego. Children can use five programming languages, according to their skills but also depending on the robot kit that is being used. Also, through a long-lasting cooperation with B. Demo of the University of Turin, a compiler for the NQCBaby language is available with a very simple and user-friendly interface that children have immediately accepted. Pupils describe the desired robot behaviors in NQCBaby programs that are translated into NQC [5]. Thus they keep a competent use of the language primitives and are enabled to learn.

After four years of experiments, enrichments and modifications of the methodology, the schools involved in the PIONEER project may claim that the educational use of robotics, in favorable circumstances, allows kids to attain powerful skills for their cognitive development. Schools with longer time experience have been able to observe that students involved in robotics activities for six school years, i.e. from their primary school second grade to the junior secondary third grade, are able to solve meaningful problems and write the related programs with robots equipped with sensors and actuators.

So, in the last year it has been possible to experiment both in the kindergarten and in the lower grades of the primary school the *Bee-Bot*, the bee-shaped robot, a programmable machine that involves children in the use of the first computer procedures, as was explained in the previous section. After the Bee-Bot, it is possible to work with the *Scribbler*, the blue turtle (also called "the messy robot") that aims at simulating what the children program in Logo with Microworlds.

In the upper grades of the primary school, the Lego Mindstorms bricks allow to use various languages (both iconic and textual), to implement paths with several types of sensors, and to find different meaningful solutions to given problems. To conclude, in junior secondary school, activities using the most recent Lego NXT robot, more complex and refined in its components, meets the different needs of teenagers, without forgetting the application of the PIONEER project methodology aiming at the student cognitive development rather than at promoting coding skills. During 2007/2008, in Baveno school, four different types of mini-robots have been used; they have been programmed by means of six different languages, depending on pupils' grades and previous experiences. Such numbers show the growth of experiences with robot use in Baveno school during about five years from first activities. Students educated through robots in the schools of the network come out having an idea of the ubiquitous technology not as a black box or a magic, rather as a world they can control because they understand it.

5. A textual programming language and related software tools

An integrated development environment (IDE) and a compiler of the programming language NQCBaby into the NQC language for the RCX robot are currently available to schools, while a compiler of NQCBaby for the NXT robot is being developed by students of the University of Turin, Dipartimento di Informatica [10]. A platform-independent method is a PIONEER future aim for providing a single child-oriented textual language, to be used for programming all different robot types. This language is based on the NQCBaby language, therefore based on the native children tongue and, following the Logo philosophy, with primitives coming from the children language. As a matter of fact, our approach consists in allowing children to use easier languages, rather than building tools to make easier the existing languages, such as the "wood icons" for the iconic programming language proposed in [12]. The PIONEER methodology defines an NQCBaby gradual introduction to schoolchildren with language enrichments from children at beginning-to-write level that use NQCBaby0 to NQCBaby6 level, usually for junior secondary school. NQCBaby0 is the kernel of the language. It is the textual form of the button commands on the Bee-Bot back.

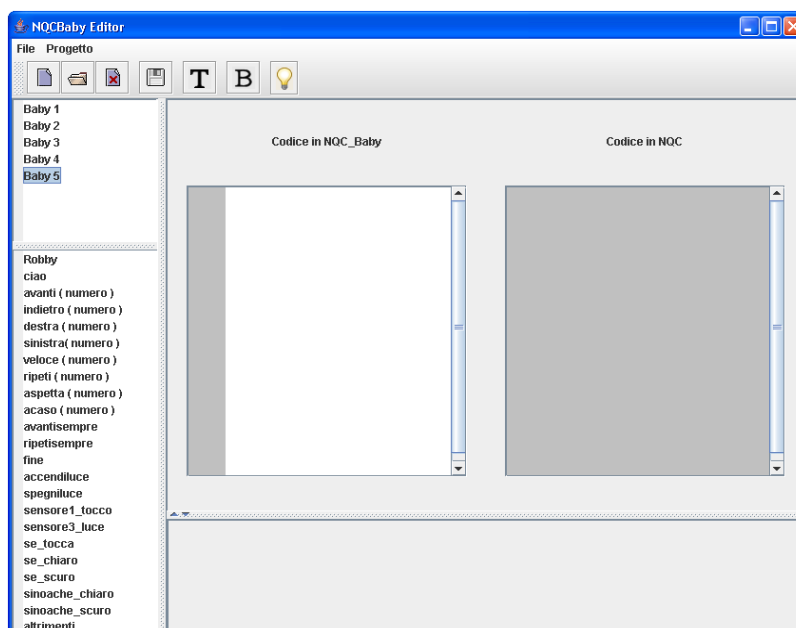


Figure 1. PIONEER IDE window

Children write their NQCBaby programs using the Integrated Development Environment (IDE) interface shown in Figure 1. The "white board" in the center of the window is where children write their NQCBaby code. On the top left side, we have the toolbar where the button T is used for translating the NQCBaby code. Errors are reported at the bottom with the code line. Language levels are written on top of

the left column indicated as Baby1, Baby2 and so on. Each successive level encapsulates the previous ones and deals with a different robot needing/allowing new primitives or new hardware components, sensors or actuators. Ordered introductions of new components, for example sensors, and related primitives for using them in robot-programmed behaviors go along with the progress of schoolchildren's logical and linguistic abilities [7] [10]. In this way, robot programming fits the learning achievements and becomes an original tool that contributes to strengthening the advances in standard linguistic and logical curricula. The language grows with children, with their school education and with what they can/want to do with their different robots.

Following the mini-language approach, NQCBaby is not a complete language, because our purpose is not that children become skilled professional programmers, rather that they have the opportunity to use concrete robots for doing **concrete programming**, i.e. for solving problems by using the basic yet complete structures of algorithmics, as from Jacopini-Böhm theorem [13], [14].

When the RCX robot is used, NQCBaby is translated into NQC. When an NXT robot is used, NQCBaby is translated into the NXC (Not eXactly C) language by means of a compiler under development. For NXT the last extension of the language provides primitives that better fit the NXC language, target of the translation. In Figures 2 and 3 two NQCBaby examples are shown, in an English translation for the sake of comprehension.

<pre> Hi Robbi speed(3) forward(100) speed(7) backward(100) repeat(3) right(90) left(90) end repeat(2) backward(10) forward(20) end thanks-bye </pre>	<pre> task main() { SetPower(OUT_A+OUT_C,3); OnFwd(OUT_A+OUT_C); Wait(100); SetPower(OUT_A+OUT_C,7); OnRev(OUT_A+OUT_C); Wait(100); repeat(3) { OnFwd(OUT_A);OnRev(OUT_C); Wait(90); OnFwd(OUT_C);OnRev(OUT_A); Wait(90); Off(OUT_A+OUT_C); } repeat(2) { OnRev(OUT_A+OUT_C);Wait(10); OnFwd(OUT_A+OUT_C);Wait(20); Off(OUT_A+OUT_C); } Off(OUT_A+OUT_C); } </pre>
---	--

Figure 2. First NQCBaby example

The NQCBaby program shown in Figure 2 describes a robot strolling around: it might be a program where pupils check primitives of the language without a specific goal. The left-hand column is NQCBaby translated into English, the right-hand column is the same code translated into the NQC language.

A second program in NQCbaby is shown in Figure 3. We find in it the function **flip-coin** that in both the NQC and NXC languages corresponds to a call of the function **random**. The program describes the behavior of a robot that goes forward for

a while then chooses to turn left or right depending on the result of flipping a coin. The NXC version of the program is on the right column. By comparing the NBCBaby and the target code versions of programs here shown, we have examples of what we mean by saying that NQCBaby is a children-oriented rather than robot-oriented language.

<pre> Hi Susi repeat-always speed(75) forward(500) if (flip-coin = heads) right(1); else // it's cross left(1); end; end-repeat; thanks-bye </pre>	<pre> task main() { while(true) { OnFwd(OUT_AC, 75); Wait(500); if (Random() >= 0) { OnRev(OUT_C, 75); } else { OnRev(OUT_A, 75); } Wait(360); } } </pre>
--	--

Figure 3. Randomly going left or right

6. Conclusions

Experiences here described began with one teacher and a small number of pupils. Nowadays, the project counts about 100 teachers in 17 different primary schools for about 1000 schoolchildren from the age of 5-6 to 13. Future activities will concern evaluating the competences acquired by these already fairly large number of students. Moreover, teachers in the net will continue developing the methodology but also using it as an everyday teaching tool in several disciplines, which is one of the peculiar goals of the project. An effort is also toward extending the number of junior secondary schools involved, in order to follow the students that have programmed robots in primary school as they progress in their education life. The homogeneousness and the common support of the pedagogical methods while carrying out robot activities, though the geographical distribution and the different types of schools involved, is another peculiar aspect of our project.

Besides all the cross-disciplinary innovative activities that students will experience with robot programming, other important results specifically concern digital literacy. PIONEER pupils learn how to write in a formal language, what an integrated development environment tool is, and how to use the one we implemented specifically for this project. By using different translators for different robots, they acquire the general concept of a translator, and of its error-finding action. We can definitely say that their digital competences are to those of pupils only using an Office suite or a similar one, what the musical technique of piano players is to the one of stereo music listeners, following the *Pianos Not Stereos* paper by M. Resnick, Bruckman and Martin [15].

Other activities in primary schools will concern inquiry-based teaching techniques that also look possible in scientific subjects, particularly in mathematics. Some hints have been given in Section 3. This would be quite a positive change with respect to often currently used teaching techniques that present mathematics as a mechanical exercise, particularly in primary schools, but unfortunately also in secondary schools where, for example, solving problems of Euclidean geometry is disappearing.

Acknowledgments. Many thanks to all the members of the PIONEER project and to the schoolchildren who learned robot programming with us. Thanks to the University of Turin students who designed and implemented software tools here mentioned.

References

1. Papert, S.: *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York (1980).
2. Vygotskij L.S., *Thought and language*, Ed. Universitaria G. Barbèra, Florence (1934).
3. Vegetti M.S., *Man psychology: for a science of a socio-historic education*. In: Liverta Sempio O., Vygotskij, Piaget, Bruner. *Concepts for the Development*, Raffaello Cortina Ed., Milan. (1998).
4. Marciandò, G.: Robotics as a learning environment, In: *Didamatica Conference 2007*, 22--30, Cesena (2007).
5. Baum, D.: NQC language, <http://bricxcc.sourceforge.net/nqc>
6. Marciandò, G.: Robotics languages for schools, In: *Didamatica Conference 2006*, 185--197, Cagliari (2006).
7. Demo, G.B., Marciandò, G., *Contributing to the Development of Linguistic and Logical Abilities through Robotics*, in 11th European Logo Conference, pp. 46, Comenius University Press, Bratislava (abstracts only, full papers at <http://www.eurologo2007.org/proceedings>) (2007).
8. Brusilovsky, P., Calabrese, E., Hvorecky, J., Kouchnirenko, A., and Miller, P. *Mini-languages: A Way to Learn Programming Principles*. *Education and Information Technologies* 2 (1), pp. 65-83 (1997).
9. Marciandò, G., Siega S., *Informatics as a language*, in *Proc, Didamatica 2005*, Potenza (2005).
10. Demo, G.B.: *Programming Robots in Primary Schools Deserves a Renewed Attention*, In: *Proc. First World Summit Knowledge Society*, Athens, 24-28 September, (2008)
11. S. Kurebayashi, S. Kanemune, T. Kamada, Y. Kuno, *The Effect of Learning Programming with Autonomous Robots for Elementary School Students*, in 11th European Logo Conference, pp. 46, Comenius University Press, Bratislava (abstracts only, full papers at <http://www.eurologo2007.org/proceedings>), (2007).
12. Horn, M. S., Jacob, R. J. K.: *Tangible programming in the classroom with Tern*, In: *CHI '07 Conference On Human factors in computing systems*, 965-1970, San Jose (2007).
13. Böhm, C., Jacopini, G.: *Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules*, *J. Communications of the ACM*, 9, 5, 367--371 (1966).
14. Demo, G.B., Marciandò, G., Siega, S.: *Concrete Programming using Small Robots in Primary Schools*, In: *8th IEEE International Conference on Advanced Learning Technologies*, pp. 301--302. IEEE Press, New York (2008).
15. Resnick, M., Bruckman, A., Martin, F.: *Pianos Not Stereos: Creating Computational Construction Kits*. *J. Interactions*, 3, 6 (1996).
16. J. Kopciowsky, *The learning process according to Feuerstein*, La Scuola, Brescia (2002).

Teachers as designers of robotics-enhanced projects: the TERECOP course in Greece

Kyparisia Papanikolaou¹, Stassini Frangou², Dimitris Alimisis¹

¹Dept. of Education, School of Pedagogical & Technological Education, Greece

² Dept. of Philosophy and History of Science, Interdisciplinary Program of Graduate Studies in Basic and Applied Cognitive Science, Greece
spap@di.uoa.gr, stassini.frangou@sch.gr, alimisis@otenet.gr

Abstract. In this paper the training course about educational robotics implemented in Greece in the context of the TERECOP project, will be presented and discussed. During the course, trainees worked in a constructionist learning environment and they were actively engaged in activities working in teams with peers. Trainees initially worked as students to familiarize with materials and the programming environment, then they worked as teachers to reflect on a methodology for designing robotics-enhanced activities and the pedagogical implications of working with programmable robotic constructions in the classroom, and finally as designers constructing their own project. To enhance the sense of community and promote collaboration during and beyond the face to face meetings, an e-class was also maintained.

1 Introduction

Research on the implementation of innovations shows that it is not easy to change teachers' behaviour [3]. When designing a teacher training course it is useful to remember the educator's axiom "teachers teach as they are taught, not as they are told to teach". Thus, constructivist professional development sessions should better be based on learning activities that teachers should be able to use in their own classrooms. It is not enough for trainers to describe new ways of teaching and expect teachers to translate from talk to action; it is more effective to engage teachers in activities that will lead to new actions in classrooms.

During the 2nd year of the 'Teacher Education on Robotics-Enhanced Constructivist Pedagogical Methods' (TERECOP) project (European Programme Socrates/Comenius/Action 2.1, Training of School Education Staff) [2], six training courses on educational robotics were implemented at the corresponding European countries of the eight institutions that participate in the project. The curriculum of the course and the training methodology were designed during the first year of the project. In particular, the training methodology is constructivist in the sense that focuses on learning experiences to enable trainees to build their own understanding of the technological and pedagogical perspectives of educational robotics. As far as the implementation of the courses is concerned, we adopted a combination of face to face

meetings with online learning to enhance communication and collaboration among the course participants. However, each national team decided on specific aspects of the training context such as the schedule, the trainees' profile, the activities used through the course.

Especially, the training course implemented in Greece was held at the premises of the School of Pedagogical and Technological Education (ASPETE) in Athens, and organized in 5 face to face meetings of six teaching periods each (5x6=30 teaching periods in total) during 3 Fridays/Saturdays afternoons. In this course participated 4 trainers and 23 trainees who were teachers in service (4 teachers of primary education and 11 of secondary education) and candidate teachers. During the course, trainees worked in a constructionist learning environment since they were actively engaged in activities, working in teams with peers. To enhance the sense of community and promote collaboration through the course an e-class was also maintained. The final products of the trainees, some of them are briefly presented in Section 4, certify the potential of the proposed training methodology and implementation.



Fig. 1. Teachers as trainees

In this paper the training course implemented in Greece will be presented and discussed. In Section 2 the training course, its scope and aims, as well as the way it was scheduled is described. In Section 3 the e-class and the way it was organized and used through the course is discussed. Then in Section 4 the trainees' products are presented. The paper ends with concluding remarks briefly discussing the preliminary evaluation results based on the trainees' products and comments.

2 Training Course: context, contents, and structure

During the training course, trainees undertook multiple roles. They initially worked as students to familiarize themselves with materials and the programming environment, then they worked as teachers to reflect on the methodology for designing robotics-enhanced activities used in TERECoP and on the pedagogical implications of working with programmable robotic constructions in the classroom, and finally as designers constructing their own project.

In particular, the training course was organised in five (5) meetings that each one lasted for six (6) teaching periods of 45 minutes. The course curriculum was organised in the following six (6) sessions each one focusing on a specific theme:

- **Building a 'didactic contract'**: introduction to the course and the theoretical background aiming to agree on a "didactic contract".
- **Theoretical framework** for designing robotics-enhanced projects.

- **Focusing on construction:** Robotics as a learning object focusing on materials.
- **Focusing on programming:** Robotics as a learning object focusing on the programming environment.
- **Focusing on a methodology for designing robotics-enhanced activities:** Designing robotics-enhanced projects/activities based on the methodology used in TERECOP.
- **Trainees' projects presentation and evaluation:** course evaluation was based on questionnaires and interviews.

Below the scope and aims of each session, as well as the materials prepared and used, the activities that trainees undertook and their products, are presented.

Building a 'didactic contract'. In this session the focus is on 'breaking the ice' and constructing a 'didactic contract' between trainees and trainers. Initially the trainers and trainees introduce themselves discussing about their expectations from the course and agreeing on a 'didactic contract'. In particular, the trainers presented shortly themselves and then invited the trainees to talk in groups of 4-5 persons and each one to introduce him/herself in 2-3 minutes to the group. Trainees were asked to provide personal/professional information, to express individual learning needs and goals, expectations and possible learning difficulties. Lastly one representative from each group, shortly introduced the members of her group to the plenary. Trainees and trainers were also invited to post a message in a relevant topic at the discussion forum of the e-class shortly introducing themselves.

Then, one of the trainers presented the overall aim, the specific objectives of the course, the content, and the training methodology. The trainees were invited to express their own expectations, opinions, suggestions and ideas first in their groups and then in the plenary through a representative. Trainers and trainees discuss and decide on the 'didactic contract'. The session finished with an agreement between the trainers and trainees on the above mentioned issues and on arrangements necessary for the smooth running of the course. Finally, this 'didactic contract' was uploaded in the documents area of the e-class.

Theoretical Framework. At this session the focus is on the theoretical background of designing robotics-enhanced learning activities. Trainees undertook specific activities involving critical thinking about constructivist and constructionist principles and the role of educational robotics. Initially the trainees worked in groups of 3-4 members and each group studied a specific section of the paper 'Constructivist Learning Using Simulation and Programming Environments' [6]. Then, the groups submit an abstract, explaining what they found more important to their particular reading, at the discussion forum of the e-class in a relevant topic that was visible to the whole class. This way all the trainees shared their readings and opinions. The representative of each group presented briefly the abstract to the whole class and the trainers commented on the presentations. Then the trainer presented the basic principles of constructionist learning emphasizing on the use of educational robotics as a leaning tool.

Finally, the trainees completed their diary which was organized around the following questions: (a) What was the best that happened to you today through the course? (b) What was the worst that happened to you today through the course?

The diaries were uploaded at the private document area of each group in the e-class. Trainees were also invited to comment on their experience of the first training day submitting a message in the relevant topic at the public discussion forum.

Closing this session, the papers entitled 'Piaget's constructivism, Papert's constructionism. What's the difference?' [1] and 'Rethinking Learning in the Digital Age' [4] were proposed for further reading (they were available in the public document area of the e-class).

Focusing on construction. This session focuses on the introduction of the materials included in the Lego Mindstorms Education NXT kit, and robots' assembly. It was organized in two sections. During the first one, trainees got organised in groups of 3 or 4 members. The basic criterion for selecting a group was that its members should be able to cooperate through the face-to-face meetings but also during Easter holidays in order to develop their own project. Thus, the group formation was decided by the trainees themselves. One Lego Mindstorms Education NXT kit was given to each group and trainees worked in groups to identify sensors, motors and construction parts like blocks, axles etc. in their kit. A trainer made a brief introduction to NXT functions and then the groups were promoted to experiment with the touch sensor, light sensor and servomotor in order to become familiarized with sensors and their parameters. At the end of this section, a discussion about the technical characteristics of each sensor took place in plenary.

During the second section trainees constructed a car robot with two motors. To this end, they used instructions included in the official guide. They were also proposed to open the Lego digital designer and use it as an additional guide for the construction of the car robot. Lastly, a discussion-evaluation of their experience through the construction of the robot-car took place. The trainers and trainees agreed on a set of criteria for evaluating robotic constructions.

Focusing on programming. The third session focuses on the programming environment and the development of virtual models that guide robots with varying configurations, i.e. motors' activation using basic programming blocks within the NXT-G software, robots' assembly in different configurations and development of meaningful programs to control them.

This session was organized in three sections. At the first section, the trainees working in groups undertook specific introductory activities to the programming environment of Lego Mindstorms Education NXT. The initial project was to design a programme that moves a robot along the sides of a square. To this end, an appropriate worksheet was given that included specific instructions. Then, the trainees developed their first program and investigated the relation between power of motor and speed of the car robot constructed in the previous session. The factors which influence the final speed of the car robot were discussed in plenary. Then they were asked to investigate left and right turns with both 'move' and 'motor' blocks and finally they developed their own blocks for left turn of 90° and right turn of 90°. Each group uploaded the blocks developed through this activity at the private documents' area of the group in the e-class. Finally the groups were asked to make their robot move on a square path (final programs were uploaded). Their programs included blocks like 'move', 'motor', 'record', 'loop', whilst they also defined their own blocks.

During the second section, the trainees worked in groups with the project ‘The cat, the mouse and the master’ introducing basic programming structures and statements of the Lego Mindstorms Education NXT programming environment. Initially a mock up with black spots was put on the ground simulating the area where the cat is moving - each black spot corresponds to a mouse! -. The groups should adapt their robotic construction in order to make it work on the mock up as a cat running after a mouse. Three activities that gradually introduce trainees to different programming concepts of varying difficulty and complexity were proposed. Each activity sets a specific challenge-problem to the trainees:

- at first they should make the cat run after the mouse and stop when it reaches a black area (the mouse!) using a light sensor, the loop block, and developing their own blocks,
- then the cat’s behaviour should be ‘extended’ to be able to stop for a while and make a sound when the master touches her. To this end, the cat robot should be extended to include a touch sensor. Trainees should also extend the program using condition blocks, and blocks like Display, Sound, Wait For,
- finally they should use variables in order to make the cat move on a spiral path.

On each activity appropriate worksheets containing instructions and information about specific blocks of the Lego Mindstorms Education NXT programming environment, were provided, aiming to enable groups work autonomously.

In the discussion followed, many different ideas were proposed about the behaviour of the cat on the mock up, leading to alternative programming solutions.

Finally, in the third section, the data logging functionality was introduced. The particular activity that trainees worked with was about collecting time and distance data from a moving robot and developing graphical representations of the corresponding data that give information about the motion of the robot.

Focusing on a methodology for designing robotics-enhanced activities. This session focused on pedagogical issues arising when designing robotics-enhanced projects for students. This session organised in four sections. Trainees initially reflect on the methodology used in TERECoP for developing robotics-enhanced projects for students. Then they have a real experience working with a real project designed based on this methodology, they discuss their experience and conclude to evaluation criteria for well-designed robotics-enhanced projects. Finally they make their own proposal using the methodology to design a project outline.

Theoretical framework for designing robotics-enhanced projects. A theoretical introduction about project-based learning was made by a trainer. Then the methodology for developing robotics-enhanced projects for students proposed by TERECoP was presented. The particular methodology consists of five stages [5]: engagement, exploration, investigation, creation and evaluation. The particular stages were introduced through a real, fully developed project ‘The Bus Route’.

Trainees, working in groups as ‘teachers’, study how the project ‘The Bus Route’ is structured in stages and they analyze each stage of the project according to the type of activities involved. Each group undertakes a particular stage, study the corresponding material like the project description and the available worksheets, and comments on the teaching strategies, the role of the teacher, and the students’ tasks

involved. Then the groups present their ideas and opinions and in collaboration with the trainers result in a synthesis. The final product of this work is uploaded in e-class. *Working with a real project.* Trainees work in real conditions as 'students' with the investigation stage of 'The Bus Route' project. The scenario of this project was presented and analyzed in smaller problems/questions. Each group investigated a problem/question and suggested a solution. All solutions were presented and discussed in plenary and uploaded in e-class. Advantages of organizing cooperative activities were also discussed.

Evaluation for well-designed projects. In groups and then in plenary trainees discuss and decide on criteria for evaluating robotics-enhanced projects for students within the constructivist approach. This work resulted in a rubric including the main criteria discussed and the level of performance expected for several levels of quality.

Designing a new project. Trainees work in groups to propose an idea for a project suitable for their students. To support this process, several electronic resources (sites on the Internet) with innovative ideas about robotic constructions had been published at the discussion forum of e-class from trainers and trainees during the previous week. Finally, the groups give an abstract description of the project they intend to develop and submit it to the public forum at the e-class.

Trainees' projects presentation and evaluation. Between this and the previous session, a period of three weeks has intervened. Through this time the groups had one kit at their disposal in order to develop a new project based on the proposed methodology. So, during the final session of the training course, trainees present their own projects and receive feedback from the class. The work of each group had already been uploaded on the e-class. In particular, each group presents their project (the construction, functionalities, and suggested teaching – learning activities). Then they receive feedback from a particular group of trainees (compulsory), the rest groups (voluntary), and the trainers. The evaluation process is based on the criteria agreed in the previous session.

Finally, trainees complete an evaluation questionnaire about the course (methodology, organisation, content, e-class, learning experience and integration of robotics in the school reality) and they participate in a semi-structured interview.

3 The e- workspace

In order to enhance class communication during and beyond the face to face meetings, we created an e-workspace that we maintained through the course. To this end, we used the open source e-class platform of the Network Operation Center (NOC) of the University of Athens (<http://eclass.gunet.gr>). The trainers created a 'virtual class' or e-class in order (a) to *provide* trainees with *resources* (course content, worksheets, presentations) and *support* such as timely information about the course content & scheduling, useful resources & links, on-time support through the public areas of 'announcements' and 'forums', (b) to *promote a sense of community* among the members of the class (trainers and trainees) providing opportunities for

communication/collaboration and resource sharing during and beyond the face to face meetings.

The e-class was organized to support communication and collaboration at two levels: at *class* and *group* level. To this end, we used *public areas* for all the members of the class with different rights for trainers and trainees like the ‘Announcements’ area that permits trainees to make announcements to the class, the ‘Documents’ area that allow the trainers to upload content whilst trainees only to download the available files, the ‘Agenda’ area that allow the trainers to describe the course structure with time and session information, the ‘Links’ area where the trainers may suggest interesting Internet sites to the trainees, the ‘Forums’ area (see Fig. 2) for discussing topics where trainers and trainees are allowed to create discussion topics and submit messages. Moreover, each group was provided with a *private area* for uploading files,. We also arranged *private areas for each group* where trainees could upload their products when working with activities (such as programs or texts, the group diary at the end of each session, the material of their own project), discuss topics, and exchange e-mails. This area was also accessible by trainers. In several cases, the trainees could share their group products if these were copied in the public area.

The screenshot shows the 'eClass' platform interface. The main content area is titled 'Περιοχή Συζητήσεων' (Public Forum Area). Below the title, there is a table with the following columns: 'Περιοχή Συζητήσεων', 'Θέματα', 'Αποστολές', and 'Τελευταία Αποστολή'. The table lists several forum topics related to the TERECoP training course, including sessions on education and entrepreneurship, and various administrative notices.

Περιοχή Συζητήσεων	Θέματα	Αποστολές	Τελευταία Αποστολή
Forums του προγράμματος επιμόρφωσης TERECoP			
1η συνάντηση: Εκπαίδευση και Ρομποτική	3	33	2008-04-17 13:33
2η συνάντηση: Γνωριμία με υλικό και λογισμικό Θέματα προς συζήτηση, απορίες, σχόλια	3	8	2008-04-17 19:02
3η συνάντηση: Σχεδιάζοντας συνθετικές εργασίες εκπ. ρομποτικής I Θέματα προς συζήτηση, απορίες, σχόλια	1	3	2008-04-19 12:04
4η συνάντηση: Σχεδιάζοντας συνθετικές εργασίες εκπ. ρομποτικής II Θέματα προς συζήτηση, απορίες, σχόλια	3	5	2008-05-14 12:33
Θέματα Επιμόρφωσης Θέσεις, απόψεις, ερωτήσεις, σχόλια, παρατηρήσεις για το πρόγραμμα επιμόρφωσης, την πλατφόρμα της ηλ. τάξης κ.λπ.	1	2	2008-04-13 11:38
Ας γνωριστούμε! Συστηθείτε στην κοινότητα του TERECoP!	1	21	2008-04-18 11:24
Συνδέσμοι - Χρήσιμες πηγές Ας μοιραστούμε ενδιαφέρουσες πηγές με θέμα την αξιοποίηση της ρομποτικής στην εκπαίδευση	1	2	2008-04-17 14:32

Fig. 2. A screenshot of the e-class of the training course. The public forum area is depicted organised in different topics.

During the course we used the *public areas as tools for administration purposes*, for example for providing the course content and worksheets before each session and

timely information about the course organization or each individual session, as well as the *public and private areas for teaching purposes promoting reflection and social interaction*. For example, we used the public forum to organize a ‘helpdesk’ where everyone could submit a problem or provide a solution, to stimulate trainees introduce themselves and share their expectations, to make trainees express themselves in specific discussion topics, share and reflect on their peers’ ideas, experiences, and perspectives - e.g. trainees at the end of each session submit a comment on their learning experience of the day or suggest interesting and useful links on the Internet whenever they locate it-.

4 Trainees’ Projects

During the course, trainees had to design their own projects based on the proposed methodology. Six of the seven groups of trainees developed and submitted interesting projects. All the groups worked with the Lego Mindstorms kit and programmed the robotic construction with the Lego Mindstorms Education NXT version 1.0. Below we provide brief presentations of the six projects.

Project 1: selector of recycled garbage. This group consisted of two mathematicians (a woman and a man) and two computer scientists (2 women). The man had strong experience on Lego Mindstorms, whilst the three women were beginners. In this project, students work in groups in a laboratory equipped with computers and some Lego Mindstorms kits. Students are invited to construct a simulation of a selector of recycled garbage able to identify the colour of different objects - normally garbage bags come in special colours (see Fig.3). The selector decides if the object is to be recycled or not based on its colour, and accordingly puts the object in the appropriate bin. The robot is equipped with two belts and a light (or colour) sensor. The sensor checks the colour of the objects and activates one of the two belts accordingly. Worksheets for school students were also produced by the trainees.



Fig. 3. Selector of recycled garbage

Project 2: autonomous irrigation system for water management. This group consisted of a mechanical engineer and a computer scientist (2 men), both having a basic knowledge level on Lego Mindstorms. Through this project students are invited to design and construct an autonomous

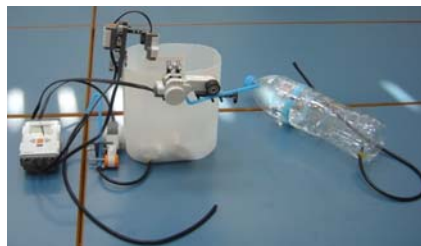


Fig. 4. Autonomous irrigation system for water management

irrigation system for water management. The basic functions of this system are: (a) fill up a tank and control of the water level, (b) control of watering from the tank during the night.

The main challenges set by this project concern (a) avoiding water loss while filling up a tank, i.e. the tank must not be overflowed, and (b) automatic provision of water from the tank when it is getting dark and the climate conditions favour watering. The characteristics of the system can be changed or enriched by students' ideas. Lego Mindstorms NXT kit, a plastic tank and watering pipes are used for the construction of the system (see Fig.4). The project is organized in 5 stages following the proposed project-based learning methodology. It aims, in addition to other objectives, to sensitise students about the rational management of water resources.

Project 3: Organizing seats in a theatre. This group consisted of a computer scientist (woman) and two physicians (men), all beginners. In this project, students are invited to construct and program a robot able to follow a predefined route in order to count the free seats in a theatre or cinema or ground, and inform the man in charge about the free seats of the whole place or a specific section (see Fig.5). Extending the project, this robotic construction might also check tickets and place the audience in the corresponding places.



Fig. 5. Organizing seats in a theatre

Project 4: Easy parking. This group consisted of a computer scientist (man) and an architect (woman), both having basic knowledge on Lego Mindstorms. In this project students are invited to construct a car-robot able to perform 'easy parking' on a mock up having several obstacles (see Fig.6). In particular, the robot should be able to identify blank spaces, avoid obstacles by turning left or right, stop, and park at free car parking places.

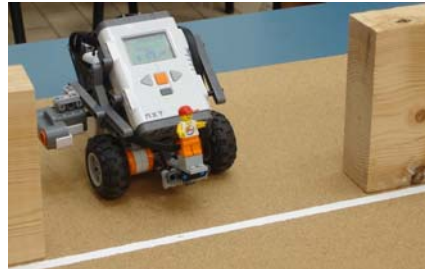


Fig. 6. Easy parking

Project 5: A moving car. This is an introductory project on robotics developed for primary education. In this project, pupils are gradually supported to cultivate basic construction and programming skills. Initially, pupils should construct a car robot and make it move forward,



Fig. 7. The catapult

backward and turn left or right. Then a challenge is set e.g. to move the car through a specific route and then move it freely in any path. This project can be expanded to a game with many challenges!

Project 6: The catapult. This group consisted of a mechanical engineer and two computer scientists, having basic knowledge on Lego Mindstorms. The project was designed for students of 15 and 16 years old. Students are invited to construct a robotic arm with one motor by following simple instructions (see Fig.7). Then they should program it to throw small balls in a basket (projectiles). In order to make it work effectively, students should conduct experiments with the parameters involved like the length of the robotic arm, the motor power, the projection angle, the horizontal distance etc. Experimental data are collected and represented in graphs using the appropriate software. Detailed examination of these graphs help students investigate relationships among the parameters involved. Finally students may continue playing a basketball game!

5 Evaluation and Discussion

In the training course implemented in Greece, a balanced whole of collaborative, learning- and teaching- focused approaches was adopted. The course evaluation was based on the trainees' products through the course and mainly on the projects they developed, the questionnaires filled by the trainees and the interviews organised. Preliminary results prove the potential of the training approach.

Trainees' projects were presented and discussed in the final session of the course. The trainees' projects followed the 5-stage methodology for designing robotics-enhanced projects that had been worked out during the training course. The description of the projects and the relevant materials (worksheets etc.) produced by the trainees indicate that the trainees efficiently adopted the proposed methodology. The trainees' projects address authentic problems from real life (projects 'recycling garbage', 'saving water resources', etc.) and engage students in problem solving through exploration and investigation activities that exploit sufficiently the potential of the educational robotics.

The trainees' answers and comments to the questionnaires and during the interviews, provided some evidence about the potential of the training course focusing on the training methodology, the content provided, the e-class, the learning experiences and the integration of robotics in the school reality.

Training methodology. Trainees recognised their active participation in all the sessions of the course and their creative involvement even in the theoretical parts introducing constructivist and constructionist principles and the methodology for designing robotics-enhanced projects. Several trainees emphasised that the educator's axiom 'teachers teach as they are taught, not as they are told to teach' was really respected in the course. They admitted that they had a real experience of constructivism ("It was for me a lesson of knowledge construction", "Constructivism was present all the time in the course", "This course was substantially different from the courses I have attended in the past").

Some comments focused on the synthesis of the groups: trainees doubt about the efficiency of the personal relations criterion for group formation purposes. Especially the group of the primary school teachers noted that “if a teacher of Informatics participated in our group, s/he would have helped us a lot...”. Other trainees emphasised that the cooperation of teachers coming from different disciplines (maths, science, informatics etc.) is necessary for the successful implementation of the projects in school settings given that the projects are normally interdisciplinary.

The communication and cooperation between trainees and trainers was appreciated by the trainees as very supportive and helpful (“we achieved a common language...”). However, they suggest that the duration of the course should be extended and the development of their own project –or most of it- should take place during the course.

Concerning the *educational content* they very much liked the activity-orientation. 75% of the trainees characterised it as very useful and the rest as useful. They also liked that they had a real case of a project (‘The Bus Route’) to analyse the different stages of the methodology. They suggested that more examples and activities for homework would be also useful.

Concerning the *e-workspace* most of the trainees evaluate the central role of the e-class during the face-to-face meetings and beyond them in enhancing social interaction and promoting a positive sense of community. They found the use of the web-based class as an interesting and useful experience that they will exploit in the future as teachers or trainers. They acknowledge the timely provision of information, course content, and support when necessary. They also acknowledge its contribution to an economic distribution of content, resources, and trainees’ products, as well as to knowledge and ideas sharing.

They mentioned that the discussion forum was mainly used for posting messages and not for real discussions since most discussions took place through the face to face communication. However, they expressed their reservations over using an e-class in real conditions as participation and administration are quite time consuming tasks.

Learning experiences and the integration of robotics in the school reality. Trainees acknowledged the potential of educational robotics as a teaching tool but also as a subject in different disciplines such as technology, informatics, and engineering.

A critical issue for integrating robotics-enhanced projects in the schools that was discussed, was how an interdisciplinary project may fit in the current school curriculum and schedule. Interesting ideas were proposed for integrating educational robotics in schools such as working interdisciplinary projects or research programs running out of the school schedule involving students from different levels e.g. engineers from vocational education working with high school students. Trainees seem also to worry about the management of big classes during the implementation of robotics-enhanced activities in school settings (“It will be difficult for one teacher to manage a school class of 30 students...”) and the cost of the necessary equipment.

Finally, trainees highly appreciated the opportunity to create their own project (“a serious gap would have been created, if I had not worked on a new project within my group”). They recognised that at the end of the course, they feel capable to implement the robotics technology in their school class (“I understood how to exploit these new ideas and technologies in my school class”).

Acknowledgements

The paper is based on work done in the frame of the project TERECOP co-funded by the European Commission - European Programme Socrates/Comenius/Action 2.1, Agreement No 128959-CP-1-2006-1-GR-COMENIUS-C21 2006 – 2518 / 001 – 001 SO2. This publication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

6 References

- Ackermann E.: Piaget's constructivism, Papert's constructionism: What's the difference? Future of Learning Group Publication (2001).
- Alimisis, D., Moro, M., Arlegui, J., Pina, A., Frangou, S., Papanikolaou, K.: Robotics & Constructivism in Education: the TERECOP project, Eurologo 2007, August 2007, Bratislava.
- Fullan, M.: The New Meaning of Educational Change (2nd edition). Toronto: The Ontario Institute for Studies in Education (1991).
- Resnick, M.: Rethinking Learning in the Digital Age. In: G. Kirkman (ed.): The Global Information Technology Report: Readiness for the Networked World, Oxford University Press (2002).
- TERECOP, WorkPackage 1: A methodology for designing robotics-enhanced constructivist learning for secondary school students (2007).
- Tsang, C.: Constructivist Learning Using Simulation and Programming Environments, MIE2002H Readings in Industrial Engineering I (2002).

Using a Programmable Toy at Preschool Age: Why and How?

Pekárová Janka

Dept. of Informatics Education, Faculty of Mathematics, Physics and Informatics,
Comenius University, Bratislava, Slovakia
pekarova@fmph.uniba.sk

Abstract. Robotic toys bring new dimension to role-play activities in kindergarten. Some preschool curricula clearly identify reasons for their inclusion. However, preschool teacher needs to revise usual teaching methods in order to use them. Offering a programmable toy or robotic-related activity doesn't mean immediate success in work with children. We document our research with concrete programmable device in a preschool classroom. Details of robotic-related sessions can help reader to design the quality game for preschool-age based on using a programmable toy.

Keywords: programmable toy, preschool, Bee-Bot

1 Introduction

What is a robot? 6-years old boy immediately responds: "It is controlled." Some friends of him also use to play with a toy robot at home. Children from preschool¹ classroom named Frogs also understand numerous purposes robot can have:

"I would like to have a robot to clean up my shoes."

"I would like it to tidy my bed."

"I want a robot which would hoe in the garden instead of me so that I could ride a bike."

However, nobody wishes to have a robot which "will ride a bike while you hoe in the garden."

We have recorded children's comprehension to robots within wider research of the ways how digital technologies can become integral part of preschool curriculum. We have accepted the broad definition of digital technologies as devices which provide *interactivity*, *response* or *communication* [3]. This definition encompasses walkie-talkies, metal detectors, remote-control cars as well as programmable toys or computer.

We explore

- *suitability* of concrete *digital tools* for preschool use,

¹ Preschool age means five to six-year-old children in our country.

- the phenomenon of *learning with digital technologies*, especially the issues how teacher should organize learning and how digital devices influence preschool routine.

We pay special attention to programmable toys that young children can access and control in a simple manner.

[9] suggest that control aspects should be included into early experiences with technologies because

- much of everyday technology is controllable;
- engaging in control activities obliges children to deal with and to construct simple 'programs';
- control technology activities may help children to develop more *general abilities to think and learn*.

If we plan to enrich preschool learning by the use of digital technologies, we may consider programmable toys also from another point of view. Digital technologies spread into many kindergartens nowadays and there is a widespread belief among educators and parents that children will require technological competencies to succeed in the workplace [6]. However, some kindergarten experts argue that digital technologies are inappropriate choice for young children's play (see [1]). Young children need to learn in concrete learning environment, to create hands-on experience with their surroundings (Beaty, 1984, in [10]). In this situation programmable toys appear to be a good choice because

- they are *tangible* technological devices and children can directly manipulate with them,
- they can stimulate problem-solving in real conditions of children's environment.

2 Programmable Toys in Curricula

Slovak national programme of education for children in kindergartens [2] doesn't explicitly require use of digital technologies in kindergarten. Despite this fact we can find older computers in many preschool settings at present and computers from IBM KidSmart Early Learning² initiative appear in larger towns by now. However, we have no evidence some kindergarten uses programmable toys in its curriculum.

The attitude to digital technologies for kindergarten children is more positive in other countries. British curriculum [7] recommends practitioners to use programmable toys to support learning. The curriculum introduces programmable toys as a good example for developing knowledge and understanding of the contemporary world. In the field of mathematical development, children should develop the ability to describe simple journey and instruct the programmable toy in order to develop position language and estimation [7]. Australian ICT Learning Innovation Centre, department of Queensland Government has even published special document (see [5]) containing valuable teachers' ideas for using specific programmable toy, Bee-Bot, in kindergarten and at primary school.

² <http://kidsmartearlylearning.org>

3 Robotics and Programmable Toys in Kindergarten Reality

Apart from Slovakia, we have discovered a few examples of good practice with programmable toys or robotics worldwide. [10] have observed children aged four to six during their interaction with Electronic Blocks building kit. Electronic Blocks have simple interface, based on Lego™ series for young children. Children have built and controlled remote-control cars and torches by connecting touch, light and movement blocks. By combining several blocks in correct order children have been able to design simple device and control it by sensors. They have developed short program sequences containing conditions (sensor input and output).

Robotics plays a vital role in the curriculum of Brazilian Escola Parque³. Kindergarten pupils have built models mostly from Lego™ parts. They have been able to design a new model according to their preferences. Some children have constructed complicated houses. They have spent much time by improving the model, but they haven't used any control elements. Other children have been more courageous, they have designed simple car models and connected small electromotors to them so that cars can move straightforward or stop. In case car didn't move, children would swap cables between battery poles. They have concentrated on designing a stable model. They have learnt basic notions about control in their work with motors. However, they will create programs for the models in computer much later, in three years period of time.

We have chosen different approach to programmable devices in the Frogs classroom in our research. We have preferred **programming aspect** of robotic toys to construction and design. That's why we have used a device which enables children to control it from the very first moments of play. In following part we will briefly introduce a programmable toy Bee-Bot™². Then we provide reader with further details about its use in concrete preschool setting.

4 Bee-Bot, the Programmable Toy

The programmable toy Bee-Bot⁴ was awarded as the most impressive hardware for kindergarten and lower primary school children on the world educational technology market BETT 2006. It uses Logo-related principle of controlling floor robot. It enables the child to program a journey on the square grid.

The design of a toy is adapted to a child user – the toy has a shape of a yellow bee with black stripes. (This design is not fixed. It can be slightly modified by the use of special plastic shells, on which child can stick paper antennae, woolen wings etc.) The toy has a small connector for a toy carriage or other moving device in its back part.

We can control the toy by a few colorful buttons. By pushing them the child enters a sequence of simple instructions for motion or rotation of a toy.

- Four orange buttons serve for a backward/forward motion and rotation to the left/right.

³ <http://www.escolaparque.g12.br/>

⁴ <http://www.bee-bot.co.uk/>

- The central button is a green GO button. It launches interpretation of the whole sequence of pushing buttons.

There are two more buttons, two blue buttons for erasing memory (CLEAR) and short break in executing commands (PAUSE) in the toy controlling part. User interface in fact copies interface of successful Pixie⁵ robot and adds child-friendly design to it.

The child can enter up to 40 instructions in one programmed sequence. User cannot modify the length of single step or size of angle rotation. These parameters are constant (which is comprehensible in relation to the target group of users), the toy moves in 15 cm in one step. Pushing rotation buttons means right angle rotation without changing toy's position.

The toy provides a simple feedback to the user. After completing the whole sequence of instructions its eyes will blink and the toy will hoot. Pushing the buttons in the mode of creating programmed sequence has also been accompanied by a silent peep sound.

Sounds can be disabled by a discreet switch in the bottom of the toy.

The way of controlling the toy is simple. Children get used to the green GO button very fast. This button is the only green button in the whole interface; moreover it is located in the central part of the toy. The slight problem appears by two blue CLEAR and PAUSE buttons. They have same colour and are placed symmetrically. The titles of the buttons can therefore be supplemented by a picture sticker for young children that cannot read, in order to distinguish between them easier.

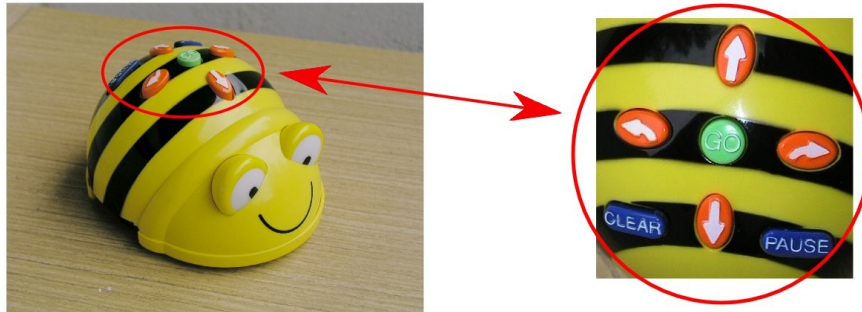


Fig. 1. Bee-Bot interface

In our qualitative research study we have observed particular problem with CLEAR button for several times. Before entering new instructions, the child shall clear the toy memory. Otherwise previous sequence of instructions is saved and by pushing buttons the child will simply add new commands in the end of program sequence. This default behavior of a toy makes sense for specific type of activity similar to *Bee-Bot knight* [4]. The activity *Bee-Bot knight* starts by locating the toy to the castle. From that starting point the toy begins its journey in several phases: on the first day it travels to collect the shield, next day to collect the shield and the sword and so on. The landscape on which the Bee-Bot playing fairy-tale character moves

⁵ www.swallow-systems.co.uk/pixie/pixie1.htm

doesn't change at all. After picking the knight accessories the teacher manually transfers the toy to the castle.

We can describe the script of similar activities by these features:

- the whole story consists of several connected phases,
- Bee-Bot is placed in the same location in the beginning of each phase,
- each phase contains repetition of all previous phases i.e. of the whole program sequence.

In case of mistake child has to enter whole sequence of commands from the very beginning one more time. If we consider different type of activities (for example moving Bee-Bot from one place to another one), saving previous instructions and the possibility to reuse them won't produce any extra effect.

On the other hand, CLEAR button develops idea of memory, saving instructions in as simple interface as possible.

The toy can be introduced in variety of age groups and school subjects (see [4], [5]): from early years to lower primary school children, for development of literacy, numeracy, natural sciences, history, geography, but also citizen or religion education.

The range of ideas for using Bee-Bot in numerous creative ways covers the basic and the only functionality of the robot – to plan the journey on the square grid map and to test the solution by executing whole sequence. Bee-Bot doesn't provide more ways how to control it. Related software product Focus on Bee-Bot⁶ simulates the behavior of the toy on screen. The software serves as an introduction to 2D and 3D computer screen representations ([9]) similarly to most on-screen control programs (Pip simulator, 2go, Jelly-bean Hunt).

The software and the physical toy are fully autonomous.

5 Robotic Activities in Kindergarten

Currently, our research team focuses on developing attractive activities and effective practice for learning with digital technologies at preschool age. We believe that robotics is one of the fields of computer science that has great potential for learning with technologies for young children.

We work on design and evaluation of various activities and tools for preschoolers. We use methods of participant observation and field notes to record and analyze positive and negative aspects of pedagogy of our sessions with preschool class. Our sample consists of 24 preschool children from partner kindergarten, with equal ratio of boys and girls.

We have conducted four sessions related to controlling programmable toy Bee-Bot. Each session has lasted from thirty minutes up to one hour.

We have been also trying to identify key ideas important for learning with control technology in preschool classroom. They can help preschool teacher who wants to effectively integrate control technology into classroom practice to avoid some mistakes we have made.

Table 1 outlines topics and methods used in each session.

⁶ www.focuseducational.com

Table 1. Topics and methods of robotic-related sessions

Topic	Method	Output
Tell me...what a robot is	Discussion	Individual, informal comprehension of the word 'robot'
First steps with Bee-Bot	Introductory presentation Group work in large groups	Comprehension of Bee-Bot's control buttons
Alarm clock alive	Motivation story Group work in small groups	Design of simple program sequence (forward and backward motion)
Birthday party	Motivation story Group work in large groups	Design of simple program sequence (forward, backward motion, simple rotations)

We had started most sessions together with all children by some introductory presentation or motivation story. Afterwards children were split into groups based on their choice.

In initial sessions we were looking for the successful script of the activity as well as to getting familiar to children whom we hadn't known before. The very first meeting *Tell me...what a robot* is consisted from informal discussion with children. We have brought out a few children ideas from this session in Introduction.

We supposed children would be charmed by a new toy. On the first session children immediately took possession of Bee-Bot with no respect to new technology. Still, they weren't able to discover the principle of robot's motion because they didn't notice it had repeated previous program sequence before executing new one.

However, we found out fast that the toy itself wouldn't provide strong motivation to sustain children's attention for longer time. We encountered serious problems on the session *First steps with Bee-Bot*. Why?

- We didn't provide children with concrete problem task. We didn't use any story mat for a Bee-Bot to move on. We tried to fix this problem by building a route from wooden building blocks. However, they didn't fully compensate original square grid. Children couldn't use accurate commands to move a robot to the end of the route, they just guessed.
- Problem of controlling the robot itself was interesting only for a few children.
- Number of children in each group was too high to offer each child enough opportunities to play with Bee-Bot more than once. Children soon became impatient and inattentive.

Session *First steps with Bee-Bot* has led us to conclusion that we need to completely rethink our approach to children. We attended preschool class two more times in order to observe teaching methods that class teacher used. We have soon

revealed that we should not rely on single activity with no choices. Then we have designed *Alarm clock alive* activity which has proved to be very successful.

5.1 Alarm clock alive

The activity has combined a powerful narrative with design elements – criteria also valid for designing popular digital games [11]. In the beginning we built map of the town from the square paper parts (castle, house, road) and asked children to place small toy figures on some parts of the town. We used the Bee-Bot as an alarm clock, we navigated it to some figure. After completing its journey, Bee-Bot hooted and awoke the figure.

We used funny badges of different colours to split children into equal groups. We worked with two groups and Bee-Bot in parallel, the other two groups worked on their own town plan. They could paint new buildings to it, colour black-and-white templates or draw some detail to existing buildings.

Children thought up different stories about the Bee-Bot, while playing. They used Bee-Bot as an alarm clock or a watchman walking through the town, taxi-driver who helped a friends to visit each other's house or a sheppard looking for lost sheep. Motivation to this activity was an intrinsic one; it arose from the story about the town. Children appreciated playing with Bee-Bot as we can see from their final remarks. Eleven children randomly chosen by class teacher (six boys, five girls) said:

- Three boys and one girl appreciated whole activity: „I enjoyed it all. “
- One girl and one boy reported playing with the bot: „I enjoyed how we played with the bee. “ The boy used the term ‘click’ instead of play.
- One boy and two girls enjoyed painting activity and playing with the bot. Another girl used the same words, but different order: „I enjoyed most how we played with the bee and drew. “
- One boy and one girl didn't mention the bee in their answer. Instead, they stated: „I enjoyed playing with the towns“, which recalls the whole activity and story-creating for programmable toy.

Organization of taking turn is a question of high importance. Sometimes teacher decides not to use programmable toys in the class because „*there's an awful 'I want a go, I want a go, I want a go! as opposed to actual just looking what it teaches us and how we are doing it.*“ [8]. Class teacher acts here as experienced observer knowing character of each child. She needs to provide equal chance to access a robot for all children, boys and girls, shy and self-confident ones. In two cases we saw children struggling to ask for their chance – for example a diffident girl didn't want to use Bee-Bot in the group of other children watching.

5.2 Bee-Bot and birthday party

In the last robotic session we prepared cardboard grids to create 3-D houses for children. Every house was then „settled“, personalized by a child's face picture. Children should use Bee-Bot as a postman to deliver birthday invitations to their friends in the same town. The activity showed us special social relationships among

children, some children refused to place their houses in the same town as a particular guy they didn't like. In such cases our team needed a support from class teacher to solve children interpersonal conflicts.

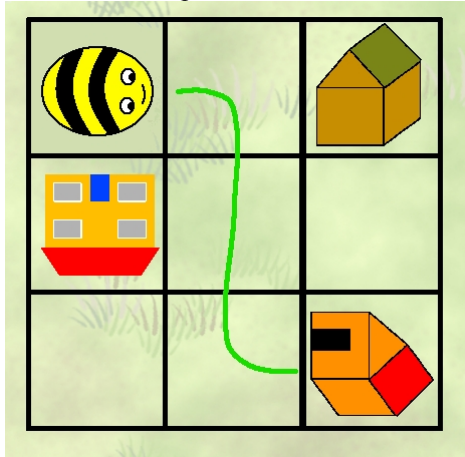


Fig. 2. Problem task set up by children

We noticed a group of boys trying to move a bot from one place to another one for a long time. They got more than five unsuccessful trials and they still didn't give up. Most of the time one boy was controlling the robot, while other boys made suggestions how to program it. The boy told the researcher he was teasing his brain. Although he and his friends had tried many times, they didn't succeed. In the end of the session a boy summed up: „I didn't like the brain – teaser“, although number of trials suggests that the group found the puzzle hard and still motivating.

When we analyzed this activity, we realized we had not taken care of monitoring and assessing child's progression with the toy. We observed special cases, children which excelled in work with a toy or those who showed clear miscomprehension of it. We didn't take notice about average child-users. Therefore we plan to prepare checklists mapping progression of each child. Some children succeeded in five from eight stages suggested for planning Bee-Bot's progression [4]. The most advanced one is described as Program Bee-Bot to move several steps forwards and backwards, including turns, in one sequence before you push GO button.

We need to provide feeling of success to each child. The proposed checklists can help us to do so. They will contain stages which child should reach or the behavior we can notice when he or she is playing with the bot. The researcher should tick a mark for every ability observed in the session.

Table 2. Checklist for introductory activities

	Sofia	Pa'ò	Dorka	Lukáš
...can go forwards		X	X	X
...can go backwards		X	X	
...can go forwards and backwards in one	X			

program sequence				
...counts number of steps needed to move the bot	X	X	X	
...can correctly say where the bot will stop				X
...can PAUSE the bot	X			
...advises to the friends	X			
...is a shy one.			X	X

6 Conclusion

Although children from the Frogs classroom haven't shown any shyness in using programmable toy as a new kind of digital technology in their classroom, the toy itself doesn't mean fun and meaningful play to them all the time. *Learning about technologies* [6], the activity *First steps with Bee-Bot*, in which we had introduced control elements of the toy to children, was interesting for children for very short time. On the contrary, children played essential role of story-writers in open-ended story about helping citizens of the colourful towns. They chose the way how the story would develop, set up own goals, challenges for a movement of a toy. Some children clearly demonstrated deep comprehension to principles of Bee-Bot's control, the others were cautious and their self-confidence didn't increase during whole series of activities with the programmable toy. However, all children enjoyed playing with the toy. *Learning with technologies* seems to be appealing to preschool children.

We evaluated several forms of work with children and Bee-Bot. The most successful model manageable by two teachers is splitting whole class into groups of no more than five members, with some groups designing, drawing, painting or building parts for Bee-Bot scenery. A group of children can challenge Bee-Bots together with one teacher. Teacher should be present in Bee-Bot group for managing taking turns, encouraging shy children and constantly providing challenges for a group. She plays important role also in monitoring and assessing children's progression.

The variability of the tasks for Bee-Bot is constrained because of its simple interface without possibility to change some parameters of its behavior. Still, it has manifested its attractiveness in open-ended activities including design elements.

Acknowledgements

I would like to thank my supervisor Ivan Kalaš and my colleague Milan Moravčík for the opportunity to be a part of unique preschool adventure. I would like to thank Mária Králiková, a headteacher of kindergarten and class teacher Darinka Belicová who enabled us access to their preschool classroom. Also I would like to thank all Frogs - children from preschool classroom.

References

1. Alliance for Childhood: Fool's Gold: A Critical Look at Computers in Childhood, www.allianceforchildhood.net/projects/computers/computers_reports.htm
2. Guziová, K., Haverlík, I., Uherčíková, V.: Programme of education for children in kindergartens (In Slovak). Ministry of Education, Bratislava, Slovakia (1999)
3. Learning and Teaching Scotland: Early Learning, forward Thinking. Learning and Teaching Scotland, Dundee, Scotland (2003)
4. Lydon, A.: Let's Go with Bee-Bot. Using your Bee-Bot across the curriculum. TTS Group. Ltd. (2007)
5. Making your classroom buzz with Bee-Bots: Ideas and Activities for the Early Phase, <http://www.earlyphaseicts.com/>
6. Plowman, L., Stephen, C.: ICT in Pre-School: A 'benign addition'? Research on ICT and preschool children. Learning and Teaching Scotland, Dundee, Scotland. (2002)
7. Qualifications and Curriculum Authority: Curriculum Guidance for the Foundation Stage. Qualifications and Curriculum Authority, London (2000)
8. Siraj-Blatchford, I., Siraj-Blatchford, J.: A guide to developing the ICT curriculum for early childhood education. 2nd Edition, Early Education, Trentham Books Ltd. (2006)
9. Siraj-Blatchford, J., Whitebread D.: Supporting Information and Communications Technology in the Early Years. 4th Edition, Open University Press, McGraw-Hill Education (2003)
10. Wyeth, P., Wyeth, G.: Electronic Blocks: Tangible Programming Elements for Preschoolers. In: Proceedings of the Eighth IFIP TC13 Conference on Human-Computer Interaction (INTERACT 2001), pp. 496-503. ACM, New York (2001)
11. Yelland, N. Playing with ideas and games in early mathematics. In: Developmentally Appropriate Technology for Early Childhood (DATEC): An Interactive Symposium, AERA, Seattle, WA (2001)

Educational Robotics Initiatives in Slovakia

Pavel Petrovič¹, Richard Balogh²

¹ Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynská dolina, 842 48 Bratislava, Slovakia, ppetrovic@acm.org

² Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Informatics, Slovak Technical University, Ilkovičova 3, 812 19 Bratislava, Slovakia, balogh@elf.stuba.sk

Abstract. We summarize, categorize, and analyze the various kinds of educational robotics initiatives in Slovakia and share our experience we have obtained while organizing the contests and preparing non-contest robotics educational activities. We argue that the team-work is highly undervalued in the current school system, and that robotics contests and project work are a very suitable platform to strengthen team-based education. We describe a proposal for a robotics module curriculum for the 1st year of informatics for the secondary grammar school. We shortly describe the tools and platforms of non-contest initiatives that our group is involved in.

Keywords: educational robotics, robotics contests, Robotnačka, NXT Logo, robotics curriculum

1 Introduction

We believe that team work is much more important than it is currently recognized at all school levels. Those companies, groups, and research centers that are able to orchestrate the team work, where the team members can communicate efficiently, where they do understand and take on their team roles easily, where the workers are able to cooperate with each other despite of various specializations and professions, and where all team members share common goal and good team spirit, those companies have a competitive advantage over others. It becomes more and more of an importance in a highly developed and structured society, where a good team behavior becomes one of the crucial aspects of successful and productive work. Even this fact is not well understood and recognized, not to mention how much our schools are lagging behind, those that prepare the workers of the future. An excellent opportunity for introducing the team work to the schools is the project-based education, and a suitable platform for that are educational robotics activities. Robotics has the advantages of:

- being interdisciplinary,
- being highly attractive to young generation as it deals with some of the technologically most advanced equipment man has ever produced,
- robotics is becoming part of every day life, it is useful to learn about it,

- it suits perfectly the didactic concept of constructionism [6],
- preparing the students for the technological and scientific fields,
- due to its interdisciplinary nature, it can provide projects in multiple subjects: mathematics, physics, science and technology courses, art courses, or even biology, but more than that: it is an invitation to cross-subject topics.

Obviously, it also has very challenging disadvantages, in particular:

- extra space required,
- high purchase and maintenance cost,
- teachers' extra time, efforts and skills,
- relative short living time,
- difficult to reuse in parallel classes if the activities exceed a single lecture.

Therefore, the introduction of robotics in the schools on a broad scale is very controversial issue, requires careful planning, and good resources. It is most suitable when the school can cooperate with a research university. However, we would rather like to see establishment of specialized robotics centers that could provide courses and excursions of various types for all the schools in the region and that would have a qualified, specialized and skilled staff. These centers could provide life-long education courses, after-school club activities, and public events.

Among the robotics educational activities, we identify two streams – robotics contests, and non-contest activities. Competitions have the advantages of:

- fixed deadline,
- clearly and exactly specified task, which is usually defined so that it is solvable,
- typically a standardized platform, meaning that building parts and experiences can be acquired and shared easier, a large community of users is available,
- possibility to reflect on and compare one's abilities against peers,
- an opportunity to acquire a prestigious prize and let others know about one's club,
- the possibility to meet other teams, exchange experience, learn from the ideas of others,
- the good spirit that is present at the competitions, often combined with seminars or lectures.

However, the non-contest initiatives also have very strong advantages, and we believe they provide higher quality as we are sometimes tired of seeing 70 line-follower robots most of them alike one another:

- they do not force the teacher into a predefined framework, rather allow him or her to setup the experiments to fit his or her pedagogical goals,
- allow the groups in the classroom to work on different projects,
- are open-ended and more suitable for research and scientific training.

In the following parts of our paper, we review both the contest and non-contest initiatives in Slovakia, most of them where we are involved in some way.

2 Contest initiatives

The popular **Istrobot** contest is held at the end of April at Slovak Technical University, this year already for the 8th time. It consists of several standard categories – Line-following (with obstacle, tunnel, and interrupted line), MiniSumo – pushing robots: a category focused mainly on mechanical design, and Micromouse – robot maze navigating contest for more advanced roboticists. Istrobot enjoys rich international participation and recognition mainly from Czech Republic and Austria. The target group for this contest are all age categories, however every year, significant number of teams from elementary and secondary schools participate. The contestants can use any type of material, hardware and software, and the participants are usually individuals. There is no particular educational concept involved and the people participating have robotics as their hobby. Istrobot is the kind of contest aiming at promoting robotics as a goal. The popularity of this contest reaches so far that it is duplicated in a separate event Metodbot, organized by enthusiastic contestants from one secondary school in Bratislava.

More than 10 years ago, Czech and Slovak initiatives established a competition in **building and programming LEGO robots** for primary (and later also secondary) schools. The task in this contest is very creative one, and participants do not bring completed robots to the contest. Instead, they bring a construction set and during



Fig. 1. The popular Istrobot contest attracts participants of all age categories. Photo: Robotika.SK.

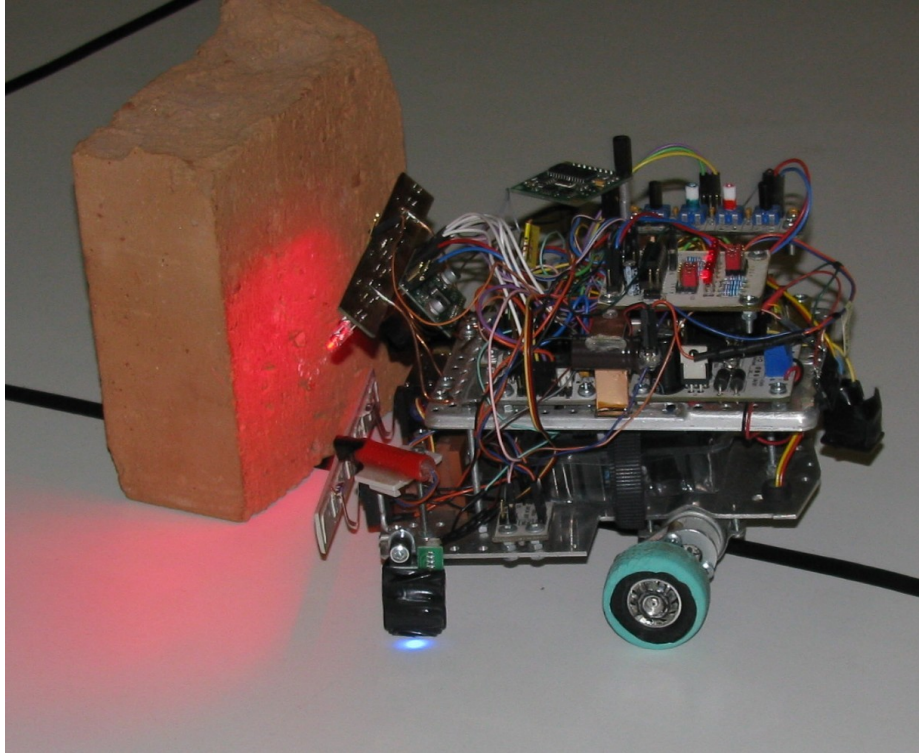


Fig. 2. Participants of Istrobot contest are primarily robotics hobbyists and/or engineering students who construct their robots from arbitrary materials and use various control platforms. Photo: Robotika.SK.

the contest, they spend several hours building and programming LEGO robots based on the topic they have learned at the start of the contest. Better prepared participants have higher chances to succeed, but the real abilities of the contestants are the main factor contributing to the team achievement, i.e. this type of contest completely eliminates any external help from the tutor, teacher, or parent. This contest continues until today, although, this year, experimentally, we have tested a different approach: instead of telling the students a topic (such as agriculture, or tourism), they received a very specific task, two training fields, and used about 5 hours to solve the task at their best (the task was a slightly modified task from the World Robot Olympiad contest). The advantage of a specific task is that the evaluation is objective and fair. In the previous creative version of the contest, the models were evaluated either directly by the contestants or by a jury, however the evaluation was a difficult discussion. The feedback we received from the participants was that the task-specific version is more interesting and more fun. In the creative version of the contest, participants often built the very same model as they already built in their club before, and modified it only a little bit to fit the assigned topic. In the task-specific version, this is impossible, and everybody has the same starting conditions. The only challenge relates to another feedback we have received from the most successful participants, who were disturbed

by the possibility of other teams copying their working ideas when testing and debugging their robots on the field. This could possibly be avoided by always allowing only a single team in the practice area.

The LEGO competition was augmented with the categories of **RoboCup Junior** contest to form one large robotics contest for primary and secondary schools, including the RoboDance, RoboRescue and RoboSoccer categories. In the year 2008, we were happy to welcome teams from three neighboring countries: Austria, Czech Republic, and Hungary. There is enough information on RoboCup Junior available, for an example, see [13]. Teams in this contest usually consist of 2-3 students. We see the main strength in the large project experience that the students acquire: they can learn what does it take to work on and successfully complete larger-scale project. This experience is of a special value as it is not available in many other forms. The notorious challenge in the RoboCup Junior is that the teams are allowed to participate multiple years and pass their knowledge and equipment onto the younger team-mates. In consequence, the best teams are those of a strong tradition and it is very difficult and thus little bit discouraging for newcomer teams to win or even advance to the finals.

Starting in 2008, we are organizing a pilot year of **FIRST LEGO League** in Slovakia (robotika.sk/fll), which we find most suitable in regard to the team-based education. The contest comes with extensive documentation, manual for the coach,



Fig. 3. The goal for the robots in the experimental modified task of World Robot Olympiad was to knock down the tins from the wooden triangular platforms. Photo by Miro Kohút.

with recommended strategies for didactics and it is the first class world standard for the primary school robotics contest. Both RoboCup Junior and FIRST LEGO League are promoting robotics as an instrument (as contrasted to robotics as a goal). Before closing the section on the contest, it is important to mention the excellent team in **FIRA robotic soccer** (category Mirostot), who are achieving the best results in European and International contests (robosoccer.sk).

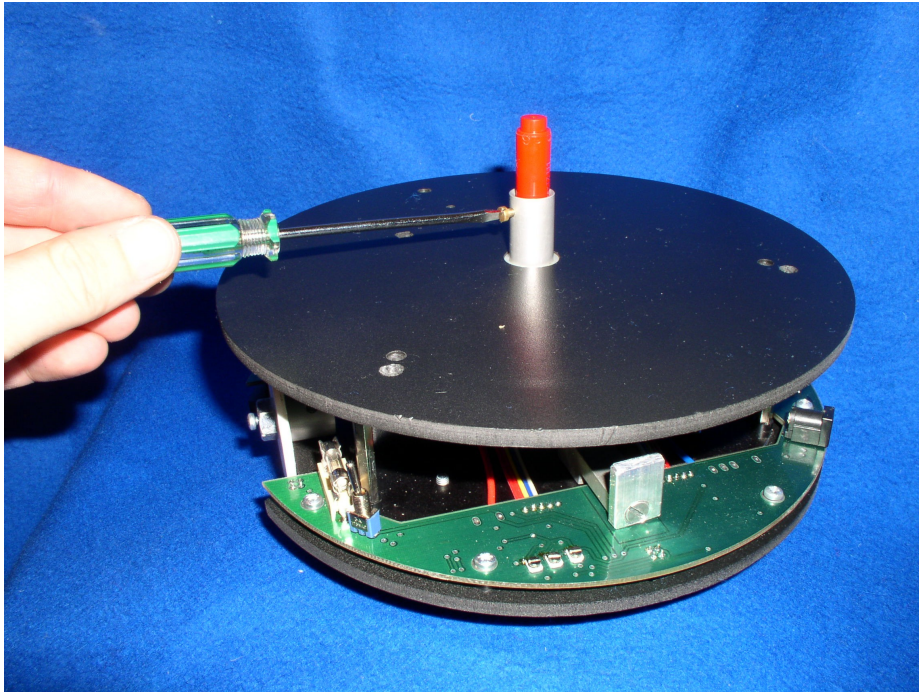


Fig. 4. Robotnáčka drawing robot. Photo: Richard Balogh, Robotika.SK.

3 Non-contest initiatives

While contest initiatives are very important for increasing the popularity of robotics, we find the non-contest initiatives to bear greater potential. A set of projects originates from the association Robotika.SK i.e. the Institute of Control and Industrial Informatics of Slovak Technical University and Department of Applied Informatics of Comenius University, and a commercial company Microstep-MIS. This consortium has built the educational drawing robot shown at Fig.4, Robotnacka [2], which can be controlled directly from the Imagine Logo programming environment [4], and can be used in the classroom to teach mathematics, physics, and programming [7]. Modified versions of Robotnacka are permanently installed in Remotely-accessible robotics laboratory, where teachers and students from anywhere on the Internet can connect directly from Imagine Logo, or any other programming language [9]. Recent member

of this family is an educational robot Sbot, a simple low-cost differential-drive robot with multiple sensors and extension possibilities.

Informatics teachers training at the Faculty of Mathematics, Physics and Informatics provides two courses introducing the future students to LEGO robotics programmable sets in a series of practical hands-on seminars [3].

Secondary grammar school of Jur Hronec, Bratislava is preparing to teach robotics with the LEGO construction sets in the 1st class as part of the informatics classes after gaining experience in after-school robotics activities earlier this year. The proposed syllabus for this course module is provided in the next section.

Robotika.SK in cooperation with BEST and Slovak Technical University organized its first robotics summer school in 2008 with approximately 25 participants attending lectures, tutorials, workshops and hands-on lab sessions: one with well-prepared course on basics of control using the Boe-Bot educational robot [1], and one with creative hands-on LEGO robotics experience.

4 Proposal for a syllabus of robotics course module for first year of secondary grammar school

Based on our cooperation with a secondary school in Bratislava, where we run an after-school robotics club, the school decided to incorporate a robotics module into their informatics curriculum. This section describes the curriculum in detail.

1. Introduction to principles of robotics – theoretical lesson with video and graphics presentations. Concepts: sensor, sensor types, motor, motor types (DC-motor, servo-motor, stepper-motor), principles of controlling robotics systems, manipulators, inverse kinematics, feedback, safety rules [5].

2. Lab – building the first model with the touch sensor based on the Constructopedia instructions in LEGO Mindstorms NXT-G. The first program. Principles of operation of sensors and motors. Modification of the model with application of the sound sensor. Disassemble the model at the end of lesson.

3. Lab – building the second model utilizing the ultrasound distance sensor, tasks/exercises:

- program the robot so that it will drive forward, but it will avoid collisions with obstacles
- program the robot so that it will stand still, and avoid approaching objects
- program the robot so that it will follow a near moving target (you can use two distance sensors)

Disassemble the robots.

4. Theoretical lesson – theoretical solution to the problem of finding shortest path in a maze (category Micromouse in Istrobot contest), solving the problem in simulation, robotic simulators, the challenges faces in robotics simulation.

5. Lab – line-following robot. The principle of the light sensor, various approaches to line following. In-depth understanding of robot interaction with its environment using sensors.

6. Lab – extending the model from the 5th lesson with obstacle avoidance, navigating an interrupted line and locating victims (category Rescue from RoboCup

Junior, and Line-follower from Istrobot).

7. Theoretical lesson – state automata, programming robots using state automata. Event-driven programs. Measuring and sampling data. Transmission of measured data to the computer and their visualization. NXT Logo – programming environment for interactive robotics projects.

8. Lab – simple football player programmed using state-automaton. Students build a football player robot based on a simple instruction sheets and program it using state automaton so that it will play the role of a football player.

9. Lab – further work on football player, improving the programs, tournament.

10. Theoretical lesson – Designing 3D graphical models for LEGO robots on PC with LEGO Digital Designer (LDD) software, exporting building instructions. Communication of robots using radio BlueTooth connection. Hardware: principles of sensors, A/D converters, pulse-width modulation for controlling motors, measuring signals using oscilloscope (topics selected based on students' interests).

11. Lab – robot communication. Simple example of remotely-controlled robot, cooperating robots (team search of exit from a maze using radio communication).

12. Lab – Measuring, processing and visualization of data: quality-measurement system. Measuring the profile of objects using ultrasound sensor. Project using the NXT Logo system. Students build a system that will measure the profiles of objects moving on a conveyor belt, and transmit this information to the PC, where it will be further processed and visualized. The system will identify the faulty objects (those that do not fit the specification) and notify the user.

General rules: The pairs of the lab lessons should be combined in 2-hour sessions. It is possible to adopt a slower pace, and spread the material over larger number of lessons. During lab, the students work in pairs, and use prepared work-sheets, where they note all their progress during the lesson, measurement results, etc. They deliver these sheets to the teacher, who provides a short feedback at the start of the next lesson. The practical lab lessons are designed so that the students disassemble their robots at the end to make them available for the next group. All the programming is performed using the NXT-G system, manual is enclosed on the CD from LEGO, NXC, documentation is available at: <http://bricxcc.sourceforge.net/nbc/>), and NXT Logo in combination with Imagine Logo, documentation is available at <http://robotika.sk/NXTLogo>.

5 NXT Logo

A particular non-contest initiative focused on providing a rich and children-friendly learning environment for interactive projects, NXT Logo, is available as a prototype, while it is implemented in an interpreted language [8]. Currently, we are designing a newer version of the system [10] in standard GNU C compiler based on the open-source firmware from LEGO, which allows for higher performance, larger memory storage capacity, cleaner code structure, and tuning the low-level functionality, which is not particularly good in the standard LEGO firmware (very complex motor model, poor memory management, limited manipulation with arrays to mention a few issues). The unique combination of features of NXT Logo include: it

is a general-purpose educational Lisp-like functional language; it introduces new level of LEGO Robots programming: students can create interactive educational LOGO projects that control LEGO robots with easy button/turtle controls, and finally, it allows flexible visualization of data collected by robots – programmable by children Logo programmers! It is implemented in Imagine Logo and Next Byte Codes (NBC). NXT Logo has three levels of use 1) Interactive Imagine Logo projects with direct GUI controls that allow steering NXT robots over Bluetooth radio, 2) Loadable imagine library (`nxt.imt`) that contains a set of procedures for direct control of NXT robot over Bluetooth from your Imagine projects, 3) Interpreter of Logo running on the NXT that can run logo programs (with restricted syntax), which can communicate with Imagine projects and control the robot motors and sensors. In addition, NXT Logo is a self-contained programming language and can be used completely without Imagine Logo. The latest addition to NXT Logo is the library for data visualization for Imagine Logo, named Charts [11,12]. It allows automatic plotting of collected data in bar-charts, line-charts, xy-charts, visualization and editing of the data in tables, connecting the tables and charts, and providing logo call-back functions that can update the data based on the user entry or input from robot, see Fig. 5 and 6 for examples of charts and tables.

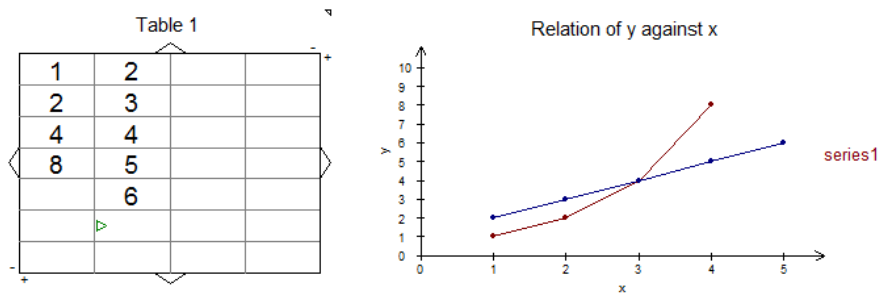


Fig. 5. The Charts library provides a set of classes with transparent interface for manipulating charts and tables in Imagine Logo. The chart on the right is updates with the table on the left.

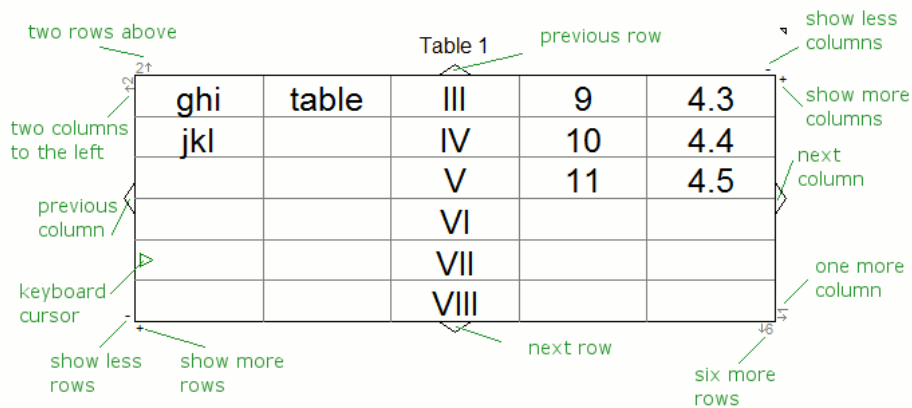


Fig. 6. The Table Logo class implements many controls for easy navigation.

6 Conclusions

Robotics in our latitudes and longitudes is still not recognized as an individual field requiring a lot of resources and attention and it struggles for support, recognition and understanding its potential and value. Therefore, promoting educational robotics depends primarily on endeavors of strongly-motivated and dedicated individuals. Broad implementation of educational robotics in the schools is not yet ready and would have to cope with large challenges, although it can be very beneficiary at the locations with sufficient resources and staff. The article describes the robotics educational initiatives in Slovakia, most of which we are involved with in some way. While the contest initiatives are very effective way of popularizing robotics, the non-contest initiatives provide more pedagogical value, and flexibility. Most important of all is to provide sufficient and good-quality tools, teaching materials, student worksheets, curriculum, platforms and options. In addition to three different contests, we are developing a rich programming environment NXT Logo, and are cooperating with the secondary grammar school that is starting to implement a robotics curriculum module in the 1st year of informatics class, which is also described in this paper.

Acknowledgments. Part of the work is supported by grant APVV LPP-0301-06.

References

1. Balogh, R.: Basic Activities with the Boe-Bot Mobile Robot, In: Proceedings of conference DidInfo 2008, FPV UMB, Banská Bystrica, Slovakia (2008)
2. Ďurina, D., Petrovic, P., Balogh, R.: Robotnačka – The Drawing Robot. Robtep 2006. Acta Mechanica Slovaca, 2-A/2006, ISSN 1335-2393, pp. 113-118. (2006)
3. Kabátová, M., Pekárová J.: Hra = učenie sa, lego a robotika vo vyučovaní budúcich učiteľov. In: Proceedings Didinfo'2008, FPV UMB, Banská Bystrica, Slovakia (2008)
4. Kalaš, I., Hrušecká, A.: The Great Big Imagine Logo Project book. Logotron (2004)
5. Kalaš, V: Tridsať rokov svetovej robotiky, STU Bratislava (2006)
6. Papert, S., Harel I. (ed.): research reports and essays 1985 – 1990. MIT Media Lab, Ablex Pub. Corp (1991)
7. Petrovic, P.: Mathematics with Robotnačka and Imagine Logo. In Proceedings to Eurologo 2005, pp. 353-360. DrukSfera (2005)
8. Petrovic, P., Lucny, A., Balogh, R., Ďurina, D.: Remotely-Accessible Robotics Laboratory, Robtep 2006. Acta Mechanica Slovaca 2-A/2006, pp. 389-394 (2006)
9. Petrovic, P.: Program Your NXT Robot with Imagine. In: Hungarologo 2007, XII. nemzetközi konferencia a Logo programnyelvvel logo-pedagógiáról, pp. 16-27, Budapest (2007)
10. Petrovic, P.: Low-level programming of NXT robots, Robolution summer school. <http://www.robotika.sk/holls/2008/> (2008)
11. Petrovic, P.: Riadenie modelov a spracovanie údajov v systéme NXT Logo, In: Proceedings to conference Didinfo 2008, FPV UMB, Banská Bystrica, Slovakia (2008)
12. Petrovic, P.: Charts for Imagine, Tutorial and Reference Guide. http://wiki.robotika.sk/index.php/Charts_for_Imagine (2008)
13. Sklar, E., Johnson, J.H., Lund, H.H.: The Educational Value of Children's Team Robotics: A Case Study of RoboCup Junior. In: Proceedings to the Seventh International Symposium on Artificial Life and Robotics (AROB) (2002)

The use of Educational Robotics for the teaching of Physics and its relation to self-esteem

Sarantos Psycharis¹, Evi Makri-Botsari², Georgios Xynogalas³

¹University of the Aegean, Department of Education and Advanced School of Pedagogical and Technological Training (ASPAITE)-Greece, sarpsy@otenet.gr

²Advanced School of Pedagogical and Technological Training (ASPAITE)-Greece, maeducation@aspete.gr

³xinogalasgeorge@hotmail.com

Abstract. The aim of this article is to use Educational Robotics(ER) in the discipline of Physics in order to investigate certain attitudes of Grade 10 students about Physics and to correlate these with certain cognitive structures and the learning performance. It is well known that the Computational experiment includes three phases, namely the modeling phase, the simulation phase and the computational phase. In this framework ER is a good candidate to implement the computational experiment since it uses the simulation phase not as a screen simulation but using a real device control. In our work ER was also used as an active learning theory tool in order to investigate the development of the algorithmic approach, a fundamental ingredient of the computational science. In our research we used the programming language Basic-Stamp and during the project students had the chance to explore-change the pseudo as well as the real code in order to make different measurements of various physical quantities and to deal with the algorithm of the application.

Key words: Educational Robotics, Didactic of Physics, Modelling, Simulation, Psychology

1 Introduction

Problem-Based Learning (PBL) is a total approach of education and involves a constructivist approach to learning [1]. The basic principle of PBL is that students learn through the process of solving so called ‘real-world-authentic’ problems. Additional features of PBL are learning in context, elaboration of knowledge through social interaction, emphasis on meta-cognitive reasoning and self-directed learning [2] and [3]. PBL can also be considered as an instructional system that simultaneously develops both problem solving strategies and learning by placing students in the active role of problem solvers confronted with practical problems in the workplace. The term Approach to Learning has been adopted (instead of term “level of processing” which had been derived from information processing theory [4] to describe differences in students’ experiences contexts, and for explaining the variation

in learning outcomes. What has been more difficult to establish is how teaching and learning environments can be designed to promote deep approaches to learning [5]. Three approaches to learning are identified in the research, namely: the conceptual approach, in which the intention is to understand concepts; an algorithmic approach, in which the focus is on calculation methods; and an information-based approach, in which the intention is to gather and remember information. The literature suggests that «approaches to learning» is a valuable tool to conceptualize the different ways in which students experience a learning context [6]. Approach to learning and the learning process is also related to cognitive styles of users [7]. Cognitive style deals with the ‘form’ of cognitive activity (i.e. thinking, perceiving, remembering), as opposed to its content. Cognitive style is usually described as a personality dimension, which has an impact on attitudes, values, and social interaction. It also refers to the preferred way individual processes information and is related to the approach of learning. Approach to learning and the learning process result is also influenced by many aspects of the human behavior such as the choice of activities, the effort exerted, the persistence on the accomplishment of a target and the skepticism about the final choice. These aspects of human behavior are related to psychological constructs such as self-esteem [8] and [9]. Self-esteem is the global perception that we develop in relation to our value as individuals, besides our self-descriptions and our self-evaluations on the various domains of our lives. Self-esteem is an intervening variable in the educational and professional decision-making process, since it relates to a group of psychological variables (self-perception of ability, accomplishment stress, values, educational attitudes, interests, personality, centre of control etc) which influence the students’ decisions. Rosenberg [10] found support for a selectivity hypothesis in that an individual will be disposed to value those things at which one considers oneself to be good and to devalue those quantities at which one considers oneself poor. Students’ beliefs were classified according to their approach to learning in Physics, using the following criteria: I am interested in explaining phenomena in a simplistic way without referring to the fundamental laws of Physics.(category 1). I am more interested in solving problems (category 2). I am interested in the various concepts in Physics in a coherent way, giving meaning to various observations in a holistic way (category 3).

2 The Computational Experiment

Computational science (which we have to distinguish from the computer science) focuses to a problem to be solved, with the components that constitute the solution separated according to the scientific problem-solving paradigm (Figure 1). Being able to transform a theory into an algorithm requires significant theoretical insight, detailed physical and mathematical understanding, algorithmic thinking and a mastery of the art of programming. The actual debugging, testing, and organization of scientific programs is analogous to experimentation, with the numerical simulations of nature being essentially virtual experiments [11].

The problem-solving method of computational physics is presented in Figure 1.

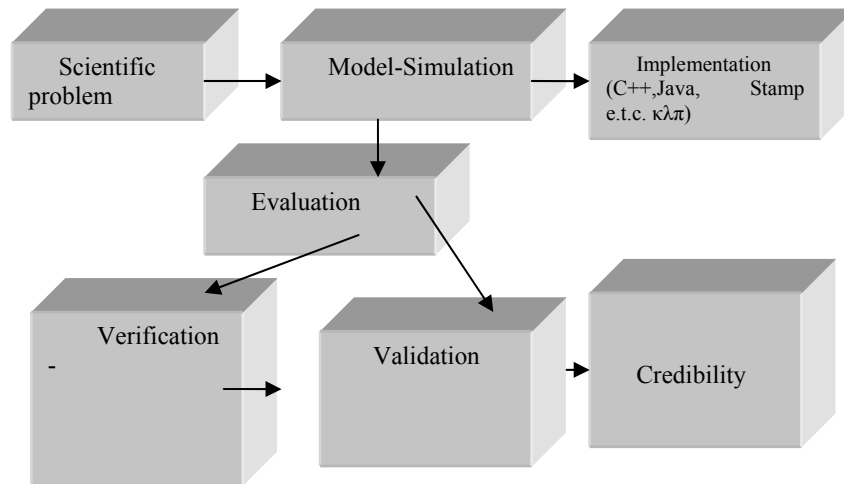


Fig. 1. The problem-solving method.

3 Talking to the Robot

A robot is a reprogrammable, multifunctional manipulator designed to move materials, parts, tools or specialized devices through various programmed techniques in order to implement various tasks. We need robots to do our jobs, communicate, and even entertain. Today's robots have also been an essential tool in a lot of fields of study. Teachers and schools use them to help students develop a better knowledge and understanding about the concepts in Physics. The aim of this project is to achieve making the robot walk and next to construct and compare two different methods of walking in order to compare their efficiency. The robot we used is called the Hex Crawler and was invented by the company "Parallax", which deals with the development of robotics.



Fig. 2. “The Hex crawler” robot

The robot consists of the following parts:

1. Hi-tech HS-322HD Servos (six for vertical and six for horizontal movement). These servos are attached to the legs of the robot. Every leg has two servos, one helps to perform the vertical movements and the other one the horizontal movements, and therefore since there are 6 legs there are 12 servos (www.robotcombat.com).
2. Board of education programming board
3. The board of education includes a power switch and a servo jumper which provides voltage to power the 12 servos in order for the robot to work. The board of education also includes a DB9 connector for BS2-IC programming and serial communication during run-time and therefore is the tool that allows the robot to interface with the computer.
4. Parallax servo controller (www.parallax.com/detail.asp?product_id=28150). The parallax servo controller is the main motive machine of the robot which controls all of the 12 servos and gives the guidelines by which the legs of the robot will move.
5. The legs. The hex crawler can work both if it has its 6 legs in operation and when it has its 4 legs in operation. Furthermore, the computer was connected with the robot via a serial cable attached to the computer and the board of education at the main body of the robot because it has the capability to hold the problems written by the computer and then execute them (Fig. 3).

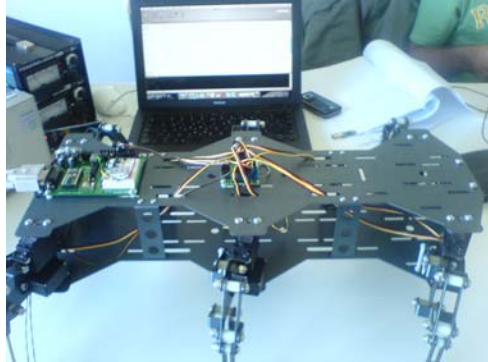


Fig. 3. Robot connected to the PC

The Basic stamp 2 module is a microcontroller (it has its own processor, memory, clock, and interface) and is used for the communication with the PC. Programming the robot lasted for four days. The CD –provided with the robot- contained a software and a compiler. The understanding of the code and the programming language was a very difficult task because it was designed only for the specific robot. This specific programming language is the BASIC stamp (<http://www.phanderson.com/stamp/index.html>). Every command written in the original software was studied independently in order to transform this to a new code suitable for the course under consideration. Robot could either walk with six legs (as initially designed), or with four legs. The program for the robot to walk with six legs was provided by the company and is called “Little step”. From the documentation it was stated that the software could be used for changing certain parameters of the motion (stability, acceleration e.t.c.) and consisted of 8 modules. The first part of the code sets values to variables and commands for the motion of the robot and determines which servo is to be moved. Other parts of the code determine the velocity of the robot, and the time delays. An example of the source code is presented below.

```
servoAddr      VAR Byte      'Servo addresses-declaration
of variables

ptrEEPROM      VAR Word      'Gait select

servoPosition  VAR Word      'to declare the position of
Servo

ramp            VAR Byte      'Ramp used in SEROUT

rightRamp      VAR Byte      'Right side ramp values

leftRamp       VAR Byte      'Left side ramp values
```

S. Psycharis, E. Makri-Botsari and G. Xynogalas

This module corresponds to the declaration of variables.

Calculations (part of the code)

```
Stride      CON  100
Delay       CON  Stride/2
Leg1Center  CON  Center1
Leg1Forward CON  Center1+Stride
Leg1Back    CON  Center1-Stride
Leg2Center  CON  Center2
Leg2Forward CON  Center2+Stride
Leg2Back    CON  Center2-Stride
Leg3Center  CON  Center3
```

After programming the robot for 6 legs we changed the code in order to have the robot running with 4 legs. The algorithm was implemented in order to give specific orders and the main changes concerned the motion of servo.

4 The Pedagogy of the Robotics

Educational Robotics deals with the concepts from different disciplines (Physics, Maths, etc) aiming to explore at all the levels of education in order to improve understanding of students of various conceptions, processes and phenomena. [12], [13] and [14].

We can consider that ER cuts the curriculum in such a way that implies a cross thematic approach to education.

ER is strongly connected to the computational experiment approach since it involves modeling, simulation and the computational phase by writing code and developing algorithms leading to the creation of cognitive structures.

In Figure 4 we present the pedagogical and computational approach of the use of ER.

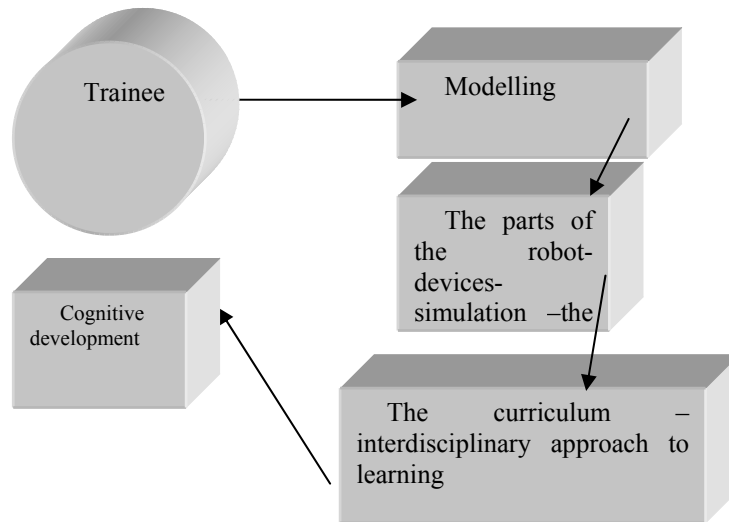


Fig. 4. The pedagogical-computational approach to the use of robot in class and its cycle

5 Research Methodology –Results

20 students of Grade 10 participated in the project which lasted for 3 months. During the research students worked with the teacher in order to explore the functioning of the robot in all aspects of its use, namely the way it operates, the algorithm of the code for the motion with 4 or 6 legs, to measure the velocity and make graphs of the displacement versus time. According to Tobochnik [15] types of manuscripts that would be appropriate for physics education fall generally into three categories.

The first of these categories consists of papers that describe a new algorithm or one that is not well known. There should be enough detail in such a manuscript so that readers could write their own program. The manuscript should not only explain the algorithm, but provide some significant examples of how it will help students learn some new physics. The algorithms might include methods of visualization, animation, numerical analysis, and simulation. In our project we wanted to combine the education in physics with ER and make students involved in the transformation of the algorithm or even the model under consideration. During the teaching-learning sequence students had to explore the parts of the robot and to relate their functioning with the modules of the software code and the algorithm. Controlling the software they could change for example the time interval for certain distance, or the acceleration, the coefficient of friction and to connect these values and measurements with the number of legs of the robot. They actually had to measure the distance and connect this concept with the time interval in order to make measurements and plot their results.

5.1 Phase 1 – Before instruction with the use of ER

Students participated in Rosenberg’s test for their classification according to their self-esteem. After that classification, a questionnaire was given to students in order to find out the approach to learning they preferred. In this questionnaire there were three possible outcomes:

A) I am interested in explaining phenomena in a simplistic way without referring to the fundamental laws of Physics (category 1). B) I am more interested in solving problems (category 2). C) I am interested in the various concepts in Physics in a coherent way, giving meaning to various observations in a holistic way (category 3). We scored the approach to learning with the scale: category 1 with score 1, category 2 with 2 and category 3 with grade 3. The total score of the Rosenberg questionnaire was in the scale 0-30. We have considered that scores ranging between 15-25 correspond to individuals with normal self-esteem (category 2), scores that are equal to or less than 15 correspond to low self-esteem (category 1) and the scores that are equal to or higher than 25 correspond to high self-esteem (category 3). Before instruction using ER, students had also to answer 20 questions for the duration of 2 hours about the issues of velocity, distance, displacement and friction. The performance scale for this diagnostic test (learning approach, learning performance) ranged from 1 to 4, with 1 being the score which corresponds to wrong answers without reasoning, 2 to correct answers with correct reasoning for less than 5 questions, 3 to correct answers with correct reasoning for more than five and less than 15 questions and 4 to correct answers with correct reasoning for more than 15 questions. We should mention that students had a level of knowledge about the physical quantities of this course from previous classes.

5.2 Phase 2 – After instruction with the use of ER

After the instruction we measured the self esteem ,the perceptions about Physics as well as students’ learning performance(learning approach ,diagnostic test). Students had to answer 20 questions for the duration of 2 hours .The performance scale for the test ranged from 1 to 4, with 1 being the score which corresponds to wrong answers without reasoning, 2 to correct answers with correct reasoning for less than 5 questions, 3 to correct answers with correct reasoning for more than five and less than 15 questions and 4 to correct answers with correct reasoning for more than 15 questions.

Table 1. Results for perceptions about Physics (1 stands for Phase 1, 2 for Phase 2).

	<i>Mean</i>	<i>N</i>	<i>Std. Deviation</i>	<i>Std. Error Mean</i>
PERCEPTION FOR PHYSICS 1	2,00	20	,725	,162
PERCEPTION FOR PHYSICS 2	2,75	20	,615	,145

The use of Educational Robotics for the teaching of Physics and its relation to self-esteem

We observe a significant shift from category 1(I am interested in explaining phenomena in a simplistic way without referring to the fundamental laws of Physics to category 3 (I am interested in the various concepts in Physics in a coherent way, giving meaning to various observations in a holistic way).

5.3 X^2 test for SELF ESTEEM and LEARNING PERFORMANCE

Table 2. Results for the relation of self-esteem and the learning performance (diagnostics test) before the instruction(phase 1).

		<i>Diagnostic test- phase 1</i>				<i>Total</i>
		1	2	3	4	
Rosenberg	1	2	1			3
	2	2	5	2	1	10
	3			3	4	7
Total		4	6	5	5	20

5.4 X^2 test for SELF ESTEEM and LEARNING PERFORMANCE

Table 3. Results for the relation of self-esteem and the learning performance(diagnostic test) after the instruction

		<i>Diagnostic test-phase 2</i>				<i>Total</i>
		1	2	3	4	
Rosenberg	1	1		1		2
	2	0	4	3	3	10
	3			3	5	8
Total		1	4	7	8	20

6 Conclusions

The main goals of the project were:

1. to investigate the development of thinking skills about certain concepts of physics due to the involvement in the algorithmic approach,
2. to study the relation of the algorithmic approach with the cognitive structure of self-esteem and learning performance and

S. Psyccharis, E. Makri-Botsari and G. Xynogalas

3. to examine the change-if any-of students' perception about Physics.

The algorithmic approach is fundamental in any kind of process which involves teaching and learning. The algorithm entered in the teaching sequence through the involvement of students in the pseudo code provided by the software and they had to 1) understand this 2) alter this by proposing certain changes.

Most of the students for example considered that the robot with six legs had a bigger velocity that the robot with four legs. Also the concept of the gradient of the graph position vs time was clarified since students could measure instantaneously both the gradient and to control the velocity in order to identify that these quantities are equal.

Our results show also a big improvement concerning the self-esteem as well as the learning outcome after the teaching-learning sequence using ER. Despite the fact 10 students remained at the category 2 of the self esteem they optimized their learning performance. Also one student shifted from category 2 to category 3.

The average value for the learning performance has increased from 2,55 at phase 1 to 3,1 at phase 2.

Interviews with the students after the experiment revealed that students felt that "doing" during the experiment provided the impulse to consider themselves as active and they actually had the control of what they did. They also considered that dealing with the algorithm of the software enabled them to be fully conscious of the problem under consideration and handling of the parameters of the code increased their self esteem.

The learning outcome (students' performance) was also quite encouraging to continue our efforts for further developments in ER. One point worthwhile to mention is that students expressed their willingness to deal with the computational phase of the experiment. They considered that with the help of the teacher they should deal with _at least –with the pseudo-code, while others wanted to deal with the source code. ER can thus enhance students' understanding of software despite the constraints helping bring a sense of authenticity to the classroom [16].

In addition, this project could also serve as a proposal to shift from the view of computational – physics education, in which the dash indicates a union of computation and physics on pretty much equal footing as individual courses or formal programs, to the computational physics--education, which views the computer as a tool to advance physics education [17] and ER can facilitate this transfer.

References

1. Hanafi , A, Rozhan ,I, Fauziah ,S.: The effectiveness of problem-based learning in the web-based environment for the delivery of an undergraduate physics course *International Education Journal*, 6(4), 430-437 (2005)
2. Boud, D. and Feletti, G.: *The Challenge of PBL*. Kogan, London (1991)
3. Norman, G.R. and Schmidt, H.G.: The psychological basis of problem based learning: A review of the evidence. *Academic Medicine* 67, 557-565 (1992)
4. Marton, F., & Säljö, R.: Approaches to learning. In F. Marton, D. Hounsell, & N. Entwistle (Eds.), *The Experience of Learning* (pp. 36-55). Edinburgh, UK: Scottish Academic Press (1984)

The use of Educational Robotics for the teaching of Physics and its relation to self-esteem

5. Jennifer, M.: Students' perceptions of context, approaches to learning and metacognitive development in a second year chemical engineering course. A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy, Faculty of Education, Monash University (2000)
6. Entwistle, N. J., & Ramsden, P.: Understanding student learning. Croom Helm, London (1983)
7. Milosevic D, Brkovic, M Dragana Bjekic, D.: Designing Lesson Content in Adaptive Learning Environments iJET International Journal of Emerging Technologies in Learning, Vol1, No 2 (2006)
8. Harter, S.: The construction of Self: A developmental perspective. 51-54 Guilford, New York (1999)
9. Makri-Botsari, E. and Robinson, P.: Harter's self-perception profile for children: A cross-cultural validation in Greece. Evaluation and Research in Education 5, 135-143 (1991)
THE ACM 49(2) (2006)
10. Rosenberg, M.: Psychological selectivity in self-esteem formation. In M. Rosenberg and H.B. Kaplan (eds) *Social Psychology of the Self-concept* (pp. 535-546). Arlington Heights, IL: Harlan Davidson (1982)
11. Landau, R. H. Paez, J. & Bordeianu, C.: A Survey of Computational Physics Introductory Computational Science. Princeton University Press, Princeton and Oxford (2008)
12. Resnick, M. and Silverman, B.: Some Reflections on Designing Construction Kits for Kids. Proceedings of Interaction Design and Children conference, Boulder, CO (2005)
13. Parsons, S. and Sklar, E.: Teaching AI using LEGO Mindstorms. In Proceedings of 2004 AAAI Spring Symposium "Accessible Hands-on AI and Robotics Education" (2004)
14. Mataric, M.J.: Robotics Education for All Ages, Accessible Hands-on AI and Robotics Education. In Proceedings of 2004 AAAI Spring Symposium "Accessible Hands-on AI and Robotics Education" (2004)
15. Tobochnik, J.: Writing articles on using computers in physics education. American Journal of Physics, 71(9) (2003)
16. Blank, D.: Robots Make Computer Science Personal. COMMUNICATIONS OF
17. Landau, R.: Computational Physics. A Better Model for Physics Education? Computing in science and engineering (2006)