Semantic and epistemological continuity in educational robots' programming languages

Emanuele Micheli, Marco Avidano, Fiorella Operto [1]

[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

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 80-89

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.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 80-89

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).

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 80-89

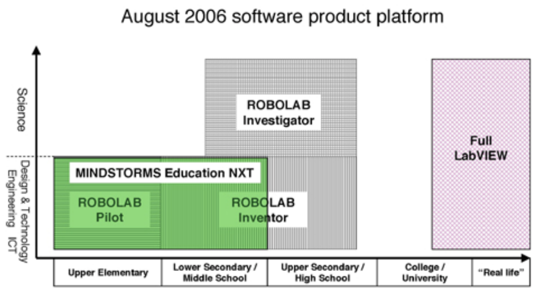

Fig. 1. Lego software production following the release of the NXT [11]

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).

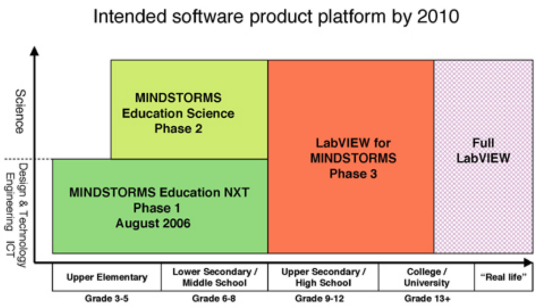

Fig. 2. The forecasted Lego's software's production by 2010 [11].

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.

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 80-89

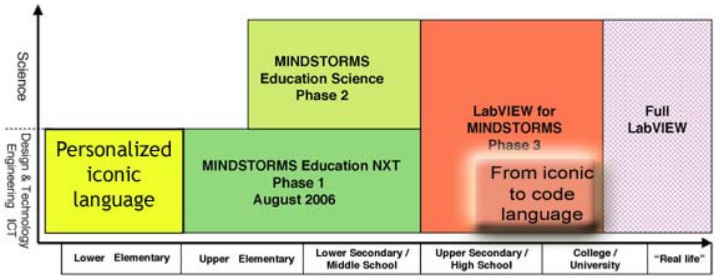

Fig. 3. Author's proposal: a "converter icon-code" which could be used by students from 5-6 years of age to high school's.

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 Ferlag, 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)

Workshop Proceedings of SIMPAR 2008
Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS
Venice(Italy) 2008 November,3-4
ISBN 978-88-95872-01-8
pp. 80-89