

Programming Robots in Primary Schools deserves a renewed attention

Barbara Demo

Dipartimento Informatica, Università di Torino, c.so Svizzera 185, Torino, Italy, barbara@di.unito.it

Abstract How to use computers in primary schools is often discussed among those involved in education. In our project we introduce robot programming in order to give both children and teachers the opportunity of concretely approaching the basics of informatics as a science while performing activities concerning the standard curricula subjects. Fundamental components of our proposal are: a) the use of different types of *small autonomous robots* as computer systems children develop programs for, b) *programming with a textual, Logo-like language* in order to avoid problems due to the language in use during first programming activities, c) a *cross-disciplinary didactical methodology* where each robot activity is a learning environment nurturing as its principal concern concepts from traditional primary school subjects, such as mathematics, physics, geography, music. Also grammar and linguistic abilities are concerned because the programming language introduction is harmonized with pupils learning of their native language writing. The choice of rather using a textual language is also for having the same representation for programming the robot and for the written natural language; d) a *community of practice* teachers can count on, so that they do not hesitate experimenting, after an introduction to robots, because they are guaranteed of pedagogical and quick technical helps. During the current 2007/2008 school year the project involved five primary schools, three kindergarten and four secondary, first level, schools for about 50 classes using various robots. Here different aspects of the experience are sketched.

KEYWORDS

Cross-disciplinary Activities, Schoolchildren-Centered Learning, Scientific Competences.

1. Introduction

S. Papert about 40 years ago and M. Resnick after two decades, began the best known projects aiming at offering children and teachers in primary schools the opportunity of approaching the basics of informatics by writing programs for small robots (Papert, 1980)(Resnick, 1993). Quite a number of researches can be found in the literature showing a continuous interest for the approach, yet neither the philosophy of Logo nor the “philosophy of robots” have entered schools everyday life. Nowadays a concurrence of reasons motivates our belief that the introduction of programmable robots in primary schools deserves a renewed attention: *several different types of robots are available suitable for all age children, robots are sold at affordable prices, small robots components are touchable and thus easier to understand, constructive learning of scientific and of non scientific subjects is naturally possible, a fast yet deep introduction to ICT is gained through robot programming, “global earth classroom” is nowadays possible via internet.* In Section 2. we develop these points here only listed.

Yet introducing programmable robots in primary schools is not easy: we must avoid the misuse of reducing robot programming to a syntactical playground as it already hap-

pened in several other cases. About Logo activities, sometimes beginning nineties, in Italy a paper appeared entitled “Beyond small houses and cute flowers” discussing difficulties teachers were having using Logo in schools. The title was too severe with respect to the paper itself but it explained the feelings of many teachers that, after a while, quit working with the turtle mostly because integrating turtle activities with the standard curricula teachers have to teach was not easy (and in most cases they had or no help). Another example, not concerning ICT yet well known, concerns the set theory. The rationale for using it in connection with mathematical basics was lost in too many primary schools in few years. We shall avoid introducing activities only syntactically perceived by pupils because this is the beginning of the most dangerous habit of superficiality.

In this paper we refer to the project *Using robots in primary schools standard curricula* begun by S. Siega and G. Marcianó in 2003 where we have more recently been active by contributing to reassess the programming language, to implement compilers, design and implement an Integrated Development Environment, create and maintain a community of practice connecting people within the project. The global experience addresses schoolchildren from around five to about fifteen years old. During the 2007/2008 school-year, the School-Net has counted three kindergarten, five primary schools and four secondary, first level, schools for about 50 classes using different types of robots starting with the BeeBot robot used by youngest, not yet writing pupils, then moving to the Parallax Scribbler robot or to the Lego RCX, for ending primary school using the NXT Lego brick (Marcianó, 2007).

Fundamental components of our project are: a) the use of *different types of small autonomous robots* as computer systems children develop programs for, b) the *programming language* NQCBaby, textual and children oriented, i.e. Logo-like, based on the macro version designed by G. Marcianó in 2004, described in (Marcianó, 2006); indeed, during first programming activities, problems often are due to the language in use. c) a cross-disciplinary didactical methodology where each robot activity is not a goal but a learning environment concerning one or more traditional primary school subjects, such as mathematics, physics, geography, music, grammar. Besides, NQCBaby introduction is harmonized with pupils learning of their native language writing and the choice of rather using a textual language is also for having the same representation for programming the robot and for the written natural language, d) *a community of practice* where, after a short introduction to robots, teachers do not hesitate experimenting with their pupils because they are guaranteed of pedagogical and quick technical helps and, also, where both teachers and schoolchildren can collaborate and foster their experience to other (new) schools. Our project community address is <http://i-teach.educ.di.unito.it/course/view.php?id=89>.

NQCBaby language, sketched in Section 3, is conceived as a set of languages having a common small kernel with few commands named NQCBaby0 corresponding to buttons present as the only programming tools in basic robots. The first actual programming language is NQCBaby1 used by children when they are given more powerful robots, with richer hardware components thus to be programmed via a programming language. NQCBaby1 is enriched to NQCBaby2 and so on till the top level language NQCBaby6 allowing a concrete and full programming experience as discussed in (Demo, 2008). Thus, the language grows with children, with their school education and with what they can/want to do with their different robots. In Section 3 we also briefly describe software tools developed for supporting activities with robots such as compilers and an integrated development environment (IDE), all open source systems implemented by undergraduate students in Informatica of the author’s university. Tools downloads are available at our community of practice address.

In Section 4 we mention some examples of robot programming and describe an activity where schoolchildren ended with discovering that for a given task each group had given a different interpretation. The task then became discovering what problem a code was going to solve, i.e. which interpretation the group, author of the code, had given to the same given task. The challenge of working on non standard problems with non standard solutions is quite relevant for the following education life of each student and it is an ex-

perience that schoolchildren have very little possibility of living because in primary schools, in particular for the scientific subjects, they are most typically given problems with unique solutions. This becomes another reason to judge that robot programming can have an important role in primary school education.

2. Motivations

A small robot is a very simple computer that can autonomously move, by means of wheels or tracks, executing a program specifying its behavior. Among teachers, we often hear saying that “Children have to stumble on a problem” to be interested in it. The possibility of moving autonomously makes the very difference between a robot and a computer in primary schools and moving a robot on a trip becomes the “problem against which schoolchildren stumble”. Moreover it is a problem pupils know how to solve “on their body”, whose planning and testing, once children begin coding the trip, becomes quite a physical activity, reason why we called *concrete programming* the use of programmable robots in primary schools (Demo, 2008). Several reasons motivates our belief that the introduction of programmable robots in primary schools deserves a renewed attention today:

- *Several types of small programmable robots are available* suitable for the different ages of schoolchildren thus allowing an experience covering the entire education life of a student. At the beginning, children about five years old are not given kits rather they use already assembled robots such as the BeeBot, <http://www.tts-group.co.uk/Bee-Bot>. This is a big “bee” with buttons on its back, each button corresponding to a command for moving one step forward, backward, right, left, for clearing (the commands previously given) and, obviously, a button for starting a sequence of commands children have given to the BeeBot by pushing its buttons. Thus children, not yet writing, begin robot programming using this button i.e. *iconic language*. Grown up pupils are given robots to be assembled and then programmed by using different types (iconic or visual or textual) of programming languages;
- *affordable prices*: small autonomous robots or kits for assembling them are these days offered at very affordable prices, thus it becomes possible organizing pupils in small groups each working at programming one robot;
- *deep introduction to ICT*: activities with small robots are quite suitable for providing both teachers and schoolchildren a reasonably fast yet deep introduction to computer programming because one can manipulate hardware components getting a very concrete understanding of how robots and software–hardware connections inside them work. Such immediacy is generally impossible when dealing with most commonly used programming languages these days, with their trappings that beginners cannot get rid of while doing first activities.
- *small robots components are touchable*: we often hear that teachers are involved in many activities and often cannot sufficiently concentrate on teaching methods for new technical subjects. In small robots, hardware components are simple enough that pupils can easily grasp how each one works: hence if robots are used with children convenient languages and software systems, they contribute by themselves conveying to pupils a good deal of the learning environment we want they experience and, even, teachers can learn with their pupils.

- *constructive learning of sciences*: in several European countries, but particularly in Italy, the number of students having considerable difficulties in scientific subjects is increasing and consequently a diminishing number chooses a scientific career on entering the university. Robot programming can provide a constructive learning environment where scientific concepts are manipulated and thus better understood than by only using other learning approaches
- *constructive learning of non scientific subjects*: cross-disciplinary robot programming activities contribute to shape learning environments suitable for a better understanding of more difficult subjects, scientific or not. That is robots are not exclusively oriented to help science learning. Indeed this requires a cooperation among teachers so that robot programming activities are crosswise used as learning environments: reducing programming to a technical subject is an error to be avoided, even worst if confined to a laboratory activity, excluded from the other subjects.
- *nowadays the global earth classroom” is possible*: the wide use of internet and the many existing online communities on the net prove that we are on the way of realizing Alan Kay’s auspice of “going from a single classroom to the global earth classroom” in (Kay, 2003). This means that teachers can count on getting help from net mates beginning from mates in the community of practice among people within each single project. Often teachers are uneasy or even worried from ICT activities: consequently little suggestion comes from them concerning their subjects where they are likely quite good.

3. One Programming Language for different robots

For all different robots we use a single programming language called NQCBaby, based on the language G. Marcianò presented in several Italian conferences since 2004 and was sketched in a paper at the EuroLogo 2007 Conference (Demo, 2007). NQCBaby is a textual language mother-tongue-based and, according to the Logo philosophy, having primitives coming from children language, i.e. children oriented rather than robot oriented. Schoolchildren are firstly introduced to a kernel of the language, then to several extensions as they are introduced either to a different robot needing/allowing new primitives or when new hardware components, in general sensors, are introduced for a robot already in use. Ordered introductions of new components, for example sensors, and of primitives for using them in robot programmed behaviors shall comply the advances of schoolchildren logical and linguistic abilities (Marcianó. 2007) Thus robot programming fits the general learning progresses children go through and becomes an original tool for contributing to strengthening standard curricula advances. As we wrote in the Introduction, the language grows with children, with their school education and with what they can/want to do with their different robots.

NQCBaby is not a complete language because our purpose is not making children become good programmers rather giving them the opportunity to solve problems by using the basic yet complete structures of algorithmics, as from Jacopini-Bhöm theorem (Bohm, 1996). In this section we focus on the rationale of NQCBaby gradual introduction to schoolchildren and sketch its enrichments from children at prewriting level using NQCBaby0 to NQCBaby6 level, usually for the last grade of primary school or first grades of the secondary school.

3.1 Children's first activities with robots

Up to now a unique compiler has been implemented translating NQCBaby into the NQC (Not Quite C) language. NQC is a complete programming language, released beginning 2004, for different types of robots and developed by Dave Baum (2004). In first experiences with robots in primary schools, programming by using available languages was found too difficult for schoolchildren: a Logo like, mother tongue based language was then introduced by Marcianó's macros in order to offer children and teachers a language children oriented, i.e. with expressions present in their language and with a semantic near to natural language expressions. The idea was designing easier languages rather than implementing tools for making easier using existing languages as in other approaches, for example Tern proposals by Horn and Jakob (2007).

NQCBaby is not a complete language: restrictions concern variables and data structures in general and conditions, typically for selection statements. In a natural language, selection statements are one of the ways to introduce subordination in sentences thus they are introduced late in primary school being enrichments of NQCBaby sentences parallel to the enrichment of children ability in her/his logical abilities and natural language writing.

Children about five years old use already assembled robots such as the BeeBot carrying buttons on its back, each button corresponding to a command for moving one step forward, backward, right, left, for clearing (the commands previously given) and , obviously, for starting a sequence of commands children have given to the robot by pushing buttons. Thus children not yet writing use an *iconic language*, we think of it as NQCBaby0, where icons are buttons on the back of the bee. Often, at the beginning of their robot activities, children say aloud what button they are pushing likely for an altogether check of what they are doing. NQCBaby0, contains few primitives with no operands. Once children have begun learning how to write their native natural language, they are given different robots, often still already assembled if security reasons suggest it because pupils are too young, but without buttons on their backs thus needing software tools to communicate the behaviour children have decided for them. Some schools of our Net use the Scribbler robot by Parallax, others use RCX and NXT Lego programmable bricks that children have to assemble. These robots can be moved either by writing the same commands that children say aloud when pushing related buttons on a BeeBot or by writing other commands with operands: i.e. kids can write *n* times forward as they pushed *n* times the forward-button on the BeeBot, but they can also write *forward(n)*. NQCBaby1 language contains NQCBaby0 primitives, same primitives with operands, i.e. *forward(n)*, *backward(n)*, *right(n)*, *left(n)* and commands such as: *speed(n)*, *stopeverything*, *repeatalways*, *repeat(n)* and few others.

A typical first example of program is shown here where schoolchildren simply try many of the primitives of the language without a specific goal; it is shown in English for sake of comprehension:

```
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
```

The following code is the equivalent program in NQC:

```
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);
}

```

First activities, where schoolchildren only want to try some primitives, in NQC would result quite elaborated, even disappointing for pupils and teachers using Italian keyboards without { and }.

3.2 Selection statements

Selection statements are one of the ways to introduce subordination in sentences. With the touch sensor we first introduce the one-way selection statement only, *if-touches*. The two-ways selection statement appears with the light sensor with its testing primitives *if-light* and *if-dark*, one opposite of the other, in NQCBaby5 normally used during the Italian fifth grade. All fundamental algorithmic structures, as from the *Ācopini-Bohm* theorem, are present in NQCBaby5. At this step, schoolchildren have got a full programming experience by means of *concrete programming experiences* as those carried out for making robots move the way each children group has planned as we point out in (Demo, 2008).

An example of NQCBaby5 shows the function *flip-coin* that in the NQC language version corresponds to a call of the function *random*. This robot is named Susi.

```

Hi Susi
  repeatalways
    speed(75)
    forward(500)
    if ( flip-coin = heads)
      right(360)
    else // it's cross
      left(360)
    end;
  end-repeat;
thanks-bye

```

The NXC version for this program is as follows:

```

task main()
{
  while(true)
  {
    OnFwd(OUT_AC, 75);
    Wait(500);
    if (Random(2)=1)
    {
      OnRev(OUT_C, 75); }
    else
    {
      OnRev(OUT_A, 75); }
    Wait(360);
  }
}

```

3.3 Technology support to children and teachers

Tools giving support to teachers and schoolchildren are obviously the compilers and also the integrated development environments (IDE) for a better accessibility to the different tools for editing, translating, possibly correcting and, finally, sending to the robot the coded behavior. Moreover tools to collaborate and learn together are mandatory when introducing robotics in schools: thus another must is an on-line community of practice environment where teachers and technical staff can integrate their skills for each other knowledge evolution.

The introduction of robot programming in schools heavily depends on all teachers because, in our purpose, robots must be used in activities where teachers introduce or make children work with concepts from primary school standard curricula they have to teach children. Thus teachers must be confident with using robots and also shall possibly get hints from colleagues as if they were, almost, together in the classroom and quickly knowing what activities have been carried out by the class. Using robots shall be a learning environment for children and for teachers as from Marcianó, 2007: hence the support of a community of practice is necessary for teachers do not hesitate experimenting as they feel guaranteed of quick helps from technicians and colleagues and, also, where teachers and schoolchildren can collaborate and foster their experience to other (new) schools, according to what Alan Kay said in a 2003 interview “Our idea is to extend the one-room schoolhouse to the entire world”.

In Section 3 we described how NQCBaby is introduced to children by subsets according to educational steps measured to their learning. This is the reason to have Baby1, Baby2 till Baby5 on top of the left column in Figure 1 where our integrated development environment window is shown. On top left side, we have the tool bar where the button T is used for translating what children write in the white “blackboard” of the window. Errors are reported on the bottom with a code line number.

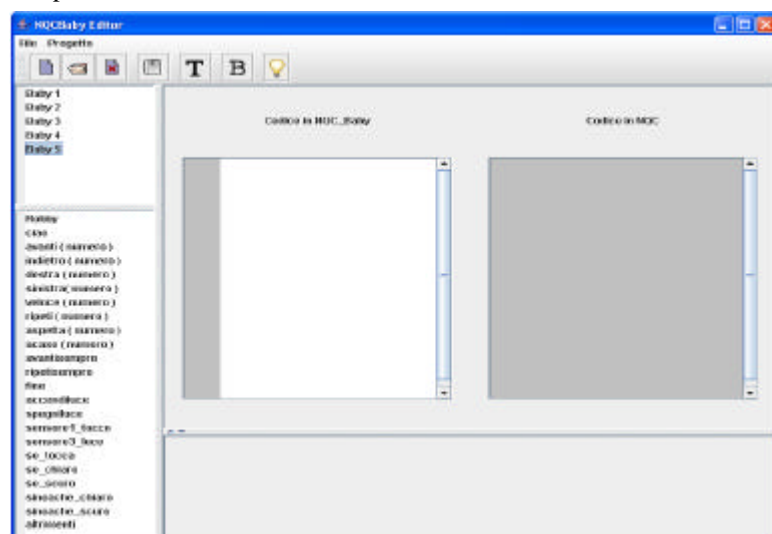


Figure 1. The integrated development environment

4. Discovering what problem a given code solves

Small robots walking through a maze and soccer games with robots teams are becoming quite popular but these activities are yet complex exercises as for algorithms and modelling components involved. Thus they are more appropriate activities for secondary schools. Simpler walks shall not be underestimated because they are a necessary first step needed into robot programming at all ages and because even simple walks can be quite interesting for pupils and teachers. Indeed teachers can convey interdisciplinary components in children ideas as those conceived and implemented by Baveno pupils with Simonetta Siega during the school year 2006/2007. Let's remember two of them: The marriage and My holidays, good examples of how also robot behaviors simple to code can produce quite successful shows.

In *The marriage* two robots dressed up one with a bowtie, the other with a long white veil leave their homes in different part of a town. They meet in front of the church and then go together to the altar with a proper music sounding. The robots paths shown in Figure 2, with the altar as the end of the walk, refer to The marriage show.

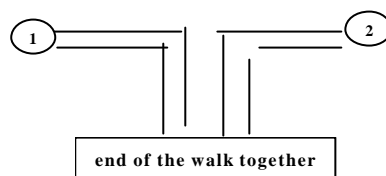


Figure 2. When we meet we walk together

In *My holidays* activity we have one robot wearing a bright t-shirt, hat and sun glasses. He leaves Baveno where the school is situated, near the mountains, on the north-west of Italy, to go to Venice to visit around. Then he moves to Ravenna (on the coast south of Venice) to say hallo to Giovanni's grandmother. But she is not at home, thus the robot goes to the beach going on and off to the grandmother's house checking whether she is back. After a while, the robot can say hallo, then goes to Rome for the summer concert where he sings. Finally he moves and reaches the seaside in Sicily, to rest!

We conclude our paper with a third interesting activity concerning fifth grade pupils of a primary school. They were asked to design an exhibition where their robots could show the geometrical figures children had "thought" her/him during the school year. Some groups came out with programs where several geometrical figures are drawn on the floor one after the other as coded in the program, always the same in the same sequence. One group, after discussions and trials, came out with a program, shown here in a short version (we do not show a code sequence concerning triangles, similar to the one shown here for quadrangle figures):

Ciao Susi	repeatUntilDark
ripeti(4)	forward(1)
ripetiSinoCheScuro	end
avanti(1)	repeatUntilLight
fine	right(1)
ripetiSinoaCheChiaro	end
destra(1)	end
fine	thanksSeeyou
fine	
grazieCiao	
Hi Susi	
repeat(4)	

Children organized a show where their robot Susi starts moving on a white ribbon on the class floor. When the robot finds a black stripe it goes right until a white ribbon is reached again and then moves straight as before until a black ribbon is found. Each run of the program one different child of the group is Susi's pilot thus in charge of deciding her path, i.e. deciding which four sided geometric figure perimeter the robot has to move on by sticking on the floor white and black ribbons. The robot goes right 4 times roughly moving on the quadrangle, regular or not, perimeter decided by her current pilot assuming the pilot closes the ribbons sequence.

Six schoolchildren groups were given the same idea to work on and began planning how their robot shall move, writing short code sequences to recall how long are the distances covered by some robot commands, drawing several designs each group for the trip they have to make his/her robot cover. The most important results concern the experience that each pupil went through while trying to plan the robot show. Among these results, particularly relevant for the following education life of each student are:

- realizing that his/her other group members can *differently understand a given non standard problem*. Non standard problems are an experience that schoolchildren have very little possibility of getting in touch with because they are usually given problems with unique solutions.
- *grasping these possibly different understandings,*
- *finding to which problem is a solution what they think and, perhaps, they code in a program* when they find out that it is not a solution to the case meant by other children in the group.

Discussing different interpretations of a given task is the beginning of learning that having questions on a subject is the true starting step of the learning process. As we said for the above robot show, a general solution to the given task has been produced by one group only but all the other group works have been considered positive solutions to different interesting tasks. Moreover, as a final show, this class presented all solutions in a sort of game where the public had to pair the six code sequences with six tasks.

Besides all the cross-disciplinary activities that primary schools pupils experience while robot programming, other important results concern digital literacy competences since 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; they learn what a translator is, its error finding action and use different translators for the different robots. Thus their digital literacy is to the one of pupils only using any Office suite or similar, as the musical technique of piano players is to the one of stereo players, following the *Pianos Not Stereos* paper by M. Resnick, Bruckman and Martin (1996).

References

- D. Baum (2004) NQC language, <http://bricxcc.sourceforge.net/nqc>
- C. Bohm and G. Jacopini (1996) "Flow Diagrams, Turing Machines and Languages with Only Two Formation Rules". *Communications of the ACM* 9 (5), 1966.
- G.B. Demo and G. Marcianò (2007) "Contributing to the Development of Linguistic and Logical Abilities through Robotics", *Proc. EuroLogo 2007 Conf.*, August 2007, p.46.
- G.B. Demo, G. Marcianó, S. Siega (2008) "Concrete Programming using Small Robots in Primary Schools", *Proc. ICALT Conference*, Santander, July 2008.
- M. S. Horn and R. J. K. Jacob (2007), Tangible programming in the classroom with tern, *Proc. CHI '07 Conference On Human factors in computing systems*, San Jose, pp 965-1970, 2007.
- A. Kay (2003) "Face to Face: Alan Kay Still Waiting for the Revolution", April/May 2003, <http://www2.scholastic.com/browse/article.jsp?id=5>
- G. Marcianò (2006) Robotics languages for schools, *Proc. Didamatica 2006 Conference*, Cagliari, 11-13 May 2006, pp. 185-197 (in Italian).
- G. Marcianò (2007) Robotics as a learning environment, *Proc. Didamatica Conference 2007*, Cesena, 10-12 May 2007, pp. 22-30 (in Italian).
- S. Papert (1980) Papert, S., *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, New York, 1980.
- J. Piaget (1983) Piaget's theory. In P. Mussen (ed). *Handbook of Child Psychology*. 4th edition. Vol. 1., 1983, New York: Wiley.
- M. Resnick, (1993) Behavior Construction Kits , *Communications of the ACM*, vol. 36, no. 7, pp. 64-71 (July 1993).
- M. Resnick, A. Bruckman, and F. Martin (1996). Planos Not Stereos: Creating Computational Construction Kits. *Interactions*, vol. 3, no. 6 (September/October 1996).