A Paradox in the Constructive design of Robotic projects in School

Javier Arlegui[1], Alfredo Pina[2], Michele Moro[3]

[1]*Psychology and Pedagogy Department, Public University of Navarra*
Campus Arrosadía, Pamplona, Spain
arleguip@unavarra.es

[2]*Mathematics and Computer Engineering Department, Public University of Navarra*
Campus Arrosadía, Pamplona, Spain
pina@unavarra.es

[3]*Department of Information Engineering, University of Padova*
Via Gradenigo 6/A-B, Padova, Italy
michele.moro@dei.unipd.it

***Abstract*— In the training of teachers for teaching robotics at the primary school level, the methodological aspects of teaching and learning are important. Constructivist methodologies and project-based learning are two "quality" tools that are proposed to the teachers (in training courses) for the design of lesson plans. But, using them we can design constructivist teaching sequences which, although progressively lead to the resolution of real and complex situations, paradoxically may not lead to a parallel progression in the learning of robotic techniques.**

We emphasize here this paradox, showing two paradigmatic examples of constructivist lesson plans for the same theme "No-driver bus ...". Only the second one guarantees parallelism between the increasing semantic complexity of the problems and the positive gradient in the syntactic component of the robot programming.

***Keywords*— Educational robotics; Robotics in school; Teaching and training for robotics; Constructivism and robotics; Project-based learning and robotics; didactic approaches in the teaching of robotics in elementary school**

I. Introduction

Educational robotics is having a significant impact in the development of education at all levels, both primary and secondary. From the methodological point of view very often the literature refers to constructivism/constructionism as a fundamental guideline of 'good practice' in the use of robots in school classes. Particularly at senior secondary level, the literature offers also a wide range of proposals and examples that go beyond the usual applications in the framework of technical vocational schools, proposing experiences of valid multidisciplinary educational content. In this context, the authors were involved in a European project aimed to define robotic-enhanced teacher training actions where the emphasis is given to the robot as a teaching/learning tool with a broad spectrum of application [1].

Scientific education currently lives a critical moment, particularly in Europe, and a huge attention is devoted to enrich curricula to encourage the attraction of scientific subjects by younger generations. Starting from primary school in that direction seems compulsory for the success of these initiatives.

Educational robotics is widely regarded as a powerful engine to promote the interest for science and technology, and therefore a issue of correctly introducing robots in primary school has arisen. The literature shows, particularly from the experiences of the first pioneer teachers using robots in class, that there are two type of problems: one issue is the robotic architecture and the other issue is the robot programming 'philosophy'.

For the first problem, some teachers solve the relative complexity of some robotic kits choosing completely mounted robots and focusing almost exclusively with the strategy to the control the robot (this is for example the case of the well know Bee-Bot). Another solution is to use a flexible kit like LEGO Mindstorms NXT but providing, possibly different, completely or mostly mounted robots to reduce the complexity of the manual construction.

This paper deals only with the second part of the problem at primary school, the programming level, showing a teacher training experience conducted in Spain. This made evident the importance of using the real, live experience of pupils to maintain a fruitful parallelism between the increasing complexity of problems to be solved and the increasing knowledge in the chosen programming language domain.

Programming the tasks that a robot can perform with the use of sensors is an excellent example of the writing of a hypothetical-deductive type of text. Thus, programming robots can help students to build their formal thinking in the "Piagetian" sense, one of the main goals in the last stage of primary education (11-12 years).

The great advantage of programming robots is that it can be organized didactically as an exploratory writing, where the robot's behaviour provides immediate feed back that helps the student to correct the errors of coherence in the program (and correct, thus, their way to think ...).

It is therefore important that the student can use a programming language that has a close correlation between the syntactic expression of the tasks and the sequential behaviour of the robot.

The LEGO's NXT-G iconic language is well suited to the earlier proposal. An icon in this language is a clearly recognizable "block", which corresponds to a robot's behaviour clearly recognizable, whose execution makes a transition between well defined states. In cases in which this correspondence fails, as we shall see later, the NXT-G programming can lead to real cognitive problems for students. And it also causes difficulties for the teacher to imagine alternative structures of programming to restore this syntactic-semantic correspondence.

II. SOCIAL AND EDUCATIONAL CONTEXT OF THE PROJECT

The childhood education and primary school "San Jorge" is located in Pamplona (a city of 200,000 inhabitants, capital of the region of Navarra, northern Spain). This institution is surrounded by the district of "San Jorge", an area of 12,000 inhabitants, built in the industrial outskirts of north-west of the city. It wants to integrate the multiethnic population of the district through an inclusive education of quality for all "... valuing diversity as an enriching element of the teaching-learning process and thus favouring human development ..."

"San Jorge" school has been recently involved in a robotic-enhanced project satisfying the desire to incorporate a science and technology oriented project in order to counteract its identification as a school with a merely humanist and social orientation due to its multicultural characteristic in this disadvantaged area.

The design of the project followed this set of general objectives, both for students and teachers:

• A robotic education for everybody;

• Based on the special skills of robotics to promote the development of formal thought;

• A constructivist teaching and learning;

• A problem based teaching and learning;

• A teacher training program of the center, supervised by professors from the Public University of Navarra

• A project complementary to other projects developed by the school, such as: inclusive teaching and dealing with diversity.

III. TWO PEDAGOGICAL LAWS TO INTRODUCE ROBOTS AT SCHOOL IN THE EARLY STAGES

The objectives abovementioned lead to two basic methodological approaches for the design of robotic-enhanced teaching units:

A. *Designing a process of constructivist teaching and learning, according to the theory of Piaget and Vygotsky.*[2], [3].

This kind of teaching unit is designed as a progressive series of problems: each problem causes an "unbalance" in the student's initial cognitive state, which asks for a cognitive effort of adaptation (assimilation - accommodation) to a new "balance" [4].

To enable the real constructive work of the student, she should be able to use some prior knowledge to solve a problem, i.e. problems should be at least partly recognizable. Thus, if a first problem P1 is of a level A and requires rebalancing to a higher level B, the successive problem P2 should start from a slightly lower level than B (say “B minus”) to accomplish a stable cognitive growth (Fig 1). The same for a successive P3 with respect to P2.

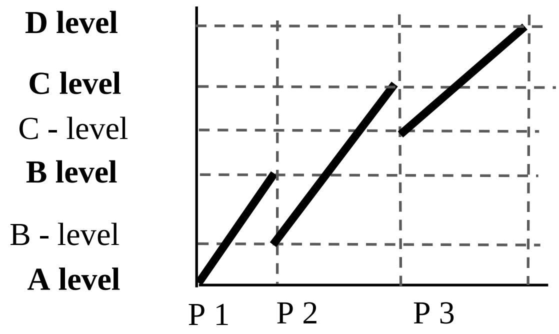

Fig. 1 Constructivist sequence

This approach requires that the constructivist work of the teacher consists in designing and applying each teaching unit regarding a certain theme or context as a sequence of problems P1, P2, P3 ... with these characteristics.

B. *Designing a project-based teaching and learning* [5].

This means that the teaching unit must have an applied nature, and that problems should gradually been formulated and motivated by the "reality" and not merely exercises of a model application. Following this idea, the constructivist work of the teacher in a project-based teaching will consist of designing each instructional unit (theme or context) as a sequence of problems P1, P2, P3 ... of increasing complexity in the real world.

IV. THE FIRST DESIGN OF THE TEACHING UNIT "NO-DRIVER BUS"

Applying the two previous methodological approaches, we have designed for school teachers in "San Jorge" a unit called "No-driver Bus". The chosen robotic architecture was LEGO Mindstorms NXT where the robot simulates a (simplified) bus moving on a linear path. The experimental progression is spread over a number of problems in contexts progressively more and more complex. The common goal can be explained as the designing of the path, and behaviour, of a bus without a driver to perform a passenger service along a highway.

C. *The sequence of problems*.

For this we have proposed a "constructive" sequence of four problems, corresponding to four different cognitive level scenarios, described below:

Problem 1

"*The* bus *must start from point A and travel for 60 cm before stopping at P1, then it must travel for 100 cm to stop at P2 and finally an additional 40 cm to reach the end of the route at point B. Each stop takes 5 seconds*"(Fig. 2)

In this enunciation of the problem, data are formulated in a robot-centered logic: in fact they are given as relative distances and thus they can be easily transformed into angles to be used as parameters in the basic movement command (the so called MOVE block of the iconic language NXT-G).

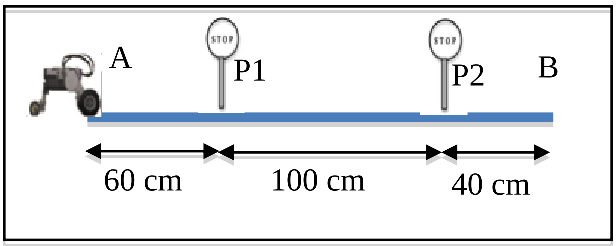

Fig. 2 Problem 1

Problem 2

"*The* bus *must start from point A and stop at P1 and P2 on the way on, then it must go back from B to A, stopping at P3. Each stop takes 5 seconds and its position is given on the chart* (Fig. 3)"

This time the text of the problem provides data in a designer-centered logic, because the given Cartesian coordinate of the stops must be transformed to relative distance in order to be used as 'operative' values in the used programming language.

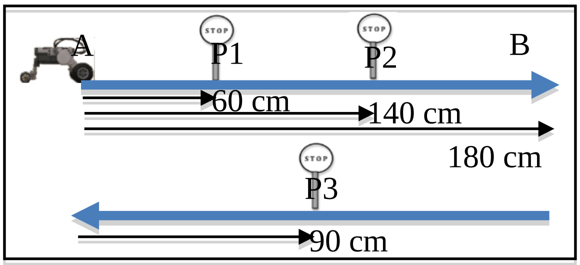

Fig. 3 Problem 2

Problem 3

"*Set up a* bus *to do the tour of the given model"* (Fig. 4).

Now data regarding distances no longer appear in the text of the problem and they should be taken from the model, i.e. a student is requested to measure the appropriate distance on the model. The model is not simply a representation of the experiment but becomes a intermediate representational space between the text (of the preceding problems) and the real world.

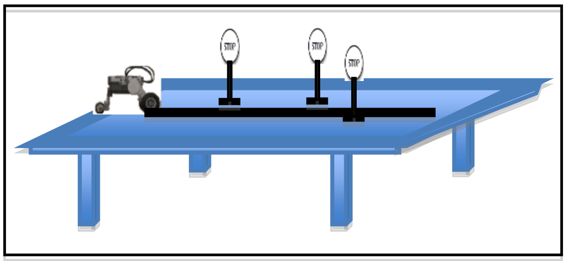

Fig. 4 Problem 3

Problem 4

"*Set up a* bus *to do the tour of San Jorge Street shown on the map*" (Fig. 5).

Now reality is much closer to the exercise: data should be directly taken from the real scenario. Students should go to 'Calle San Jorge', decide where to put the stops and take appropriate actions (the path of the robot, then, is designed on a suitable scale).

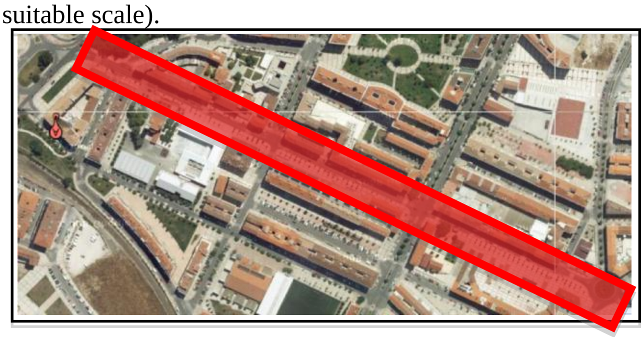

Fig. 5 Problem 4

D. THE PARADOX OF THE PREVIOUS DESIGN

The previous design of the "No-driver Bus" teaching unit performs acceptably when compared with both the teaching laws presented in section III because you can recognize the constructivist progression but also the increasing level of complexity the real world can suggest. Through the presented problems students progress adapting the robot's capabilities to the increasingly realistic conditions of a real urban travel. They finalize the project knowing various aspects of the involved population, of their activities, schedules and travel along the real Calle San Jorge. The final design of the robot bus travel could be a real and complex response to the needs of distribution of citizens in the considered urban area.

But when we see the actually task performed by the robot in the subsequent situations, we see that is essentially the same. The increasing complexity of the problems are always resolved with the same level of elementary programming because in all cases it is reduced to a repeated use of the MOVE and STOP basic blocks.

Therefore we can argue that in this type of design there is the intrinsic paradox that the increasing semantic complexity is not accompanied by a corresponding progression in the formal complexity of the programming task. Whereas students can learn more about transportations within their district, they do not learn anything new about robot programming and thus they do not exploit all the cognitive potential of the used command language.

The result of this reflection is that we need to add a "third teaching law" to the previous two, that could be expressed as follows:

"Designing a teaching and learning process based on the increasing complexity of the robot programming tasks when implementing increasingly complex behaviors of the robot".

V. A SECOND DESIGN OF THE TEACHING UNIT "NO-DRIVER BUS"

Now integrating this third criterion with the previous two, we have designed for the same group of teachers a second unit called "No-driver Bus - 2" through the constructive sequence of five problems described below.

Problem 1 is a transitory one: it has the simple aim to justify the introduction of a sensor as a component on which to make decisions. A student already knows how to control the robot to make it move through given distances and has a first idea of the importance of a sequence of commands; but she has also the direct experience that bus stops can be optional and stopping might be requested by the traveller. So the problem leads the student to relate the stopping of the motor to a condition based on a sensor. Problem 2 is a reformulation of Problem 1 with the adding of a small but logically important detail that produces a solution which corresponds more strictly to the control logic. In this solution a conditional wait is substituted by an 'active' permanent control of the stopping condition which is closer to the student's perception. Problem 3 shows how increasing requirements, such as the approach with reduced speed to the stop, can actually produce a more advanced control program, improving the previous solution.

A. Problem 1

"Designing a bus which stops at the request of a traveller: the request is represented by posing a hand in front of the robot at a distance D <30 cm".

Now the bus stops are no longer in fixed positions. The proposed problem is formulated so that it is necessary to use sensors for the solution: in this case, the student must incorporate and program an ultrasonic sensor. A possible core of the solution in the iconic NXT-G language is given in Fig 6.

The complete solution (fig. 7) must include the stopping for a given time and the repetition of the entire sequence in an undefined loop.

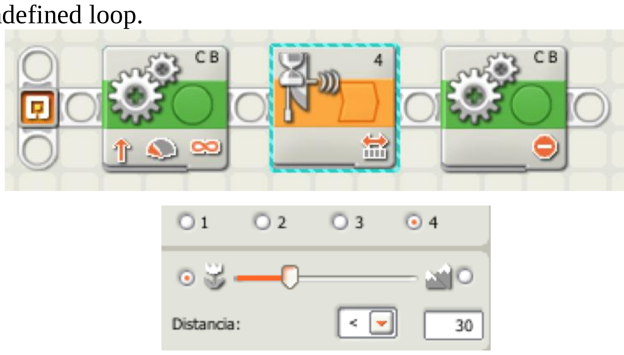

Fig. 6 Problem 1: the core sequence

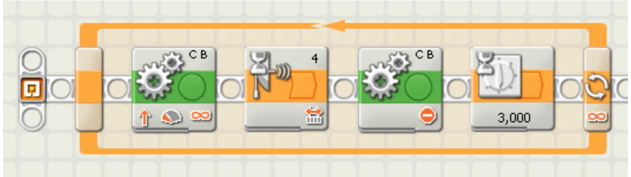

Fig. 7 Problem 1: the complete solution

The fact that the control flow is stopped waiting for a condition regarding a sensor might surprise the student: her personal experience is based on the behaviour of a bus driver who is continuously taking actions and monitoring the situation around the bus. Thus the next step is to suggest a less passive solution.

B. Problem 2

"Designing a bus which stops at the request of a traveller: the request is represented by posing a hand in front of the robot at a distance D <30 cm. Act as a driver who is looking for a travellers' request while moving the bus".

You must consider that the programming instruction "move the robot until ..." implies the use of the MOVE block with a meaning corresponding to a "special" treatment of the NXT interpreter. In fact, when you set as 'indefinite' the time/angle parameter of the motion (see fig. 6), you are not setting an action that corresponds to the transition between two distinct and well defined states S_i and S_j, as it would be in the case of a finite (in time or in space) move command. Actually the interpreter activate a (logically separate) thread indefinitely piloting the motor while the main thread continues to execute the interpreter on the following commands. In this sense the final STOP command acts as the 'killer' of this separate, previously spawn thread. Another state-oriented interpretation could be that, while waiting for the sensor, the state S_i remains unchanged and this corresponds to a (logical) loop insisting on the same state, whereas the transition from S_i and S_j is labelled by the condition when positively verified (D<30) (Fig. 8).

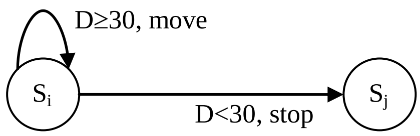

Fig. 8 A relatively complex state diagram

The comprehension of this more complex situation is, in either view, very difficult for a primary student and the solution can 'run away'.

Problem 2 could be solved using an alternative approach closer to the hypothetical behaviour of the bus driver at least in the perception of the student. In this approach the indefinite motion is broken into several micro-movements. So we define a personalized MOVE command (*small forward*) with a very small displacement and executed as an alternative of the STOP command inside an unconditional loop. This small forward corresponds to the action associated with the loop on state Si of fig. 8. The stopping of the bus is signalled by the known requirement (D<30) (Fig. 9).

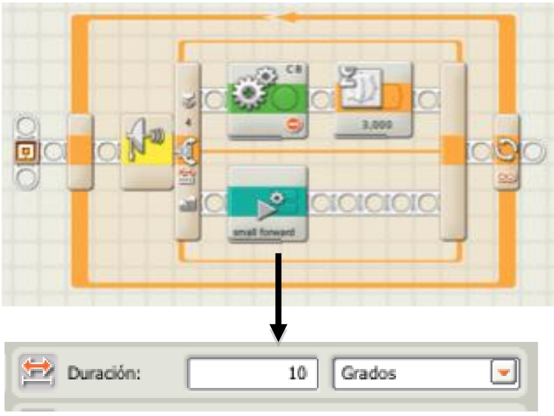

Fig. 9 Problem 2

C. Problem 3

"Designing a bus which stops at the request of a traveller: the request is represented by posing a hand in front of the robot at a distance D <30 cm. Use separate blocks for the sensor and the conditional statement".

NXT-G allows also the independent use of a sensor and a conditional statement, which makes you distinguish more clearly what is the role of the two instructions (the sensor block is of 'operation' type whereas the conditional block is of 'command' type) (Fig 10). Such a separation could be suggested to students as a further improvement, for example saying that the driver not only is aware whether a traveller on the street is requesting the stopping but he can also estimate the distance of the traveller during the approaching phase (observe that the sensor block gives also the distance measure together with the overcoming of the distance threshold).

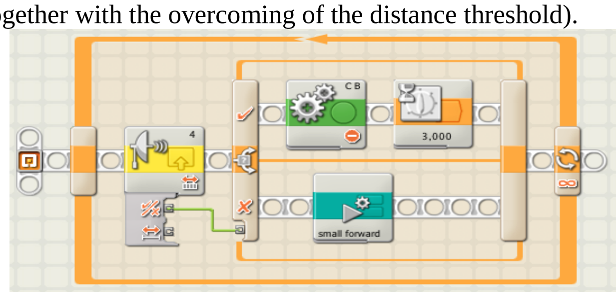

Fig. 10 Problem 3

D. Problem 4

"Designing a bus which acquire the request of stopping from a traveller, and approaches the traveller enough to permit she can get in".

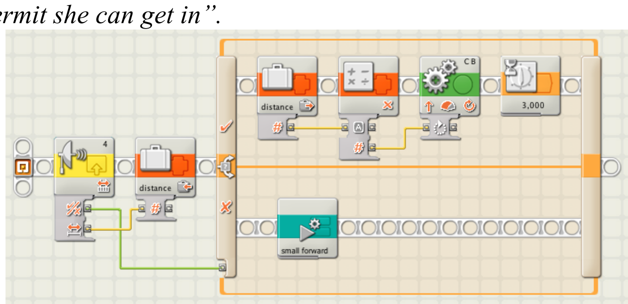

Fig. 11 Problem 4

In this problem, once located the requesting traveller, the bus must go on sufficiently to be close to the person. It is easy to verify that this can (softly) introduce the use of variables. The *distance* variable in Fig. 11 is used to calculate the necessary approaching space, improving the realism of the solution. The example shows also how the sensor output can be used both for logical (comparing with a threshold) and numerical (in absolute term) purposes.

VI. CONCLUSIONS

The sequence of three problems discussed in Section V can be further extended facing other tasks like slowing down when the bus approaches the traveller, turning a brake light on when it reduces the speed, etc. But the previous sequence is enough to demonstrate the great difference between the two constructivist views presented.

We showed that the second one can guarantee the parallelism between the increasing semantic complexity of the problems and the positive gradient in the syntactic component of the robot programming. This is in a nutshell the constructivist teaching model we propose.

This model combines in a dialectical mode a teaching/learning process where robots are "object of knowledge" and a teaching/learning process where robots are "learning tool". The first aspect corresponds to the progression of the formal complexity, the second, the semantic progression. It is possible, and sometimes desirable, to design sequences that focus on one of the two views, but in any case, the teacher should always have clear in mind these two "didactic variables " when designing teaching sequences based on constructivist educational robotics.

ACKNOWLEDGMENT

The authors of this paper wish to thank the primary school "San Jorge" (Pamplona, Spain) and their teachers, especially Juan Jose Aragon and Milagros Lorenzo, for their work and their enthusiasm for carrying forward the innovation project "Robotics education for everybody".

REFERENCES

[1] Alimisis, Arlegui, Fava, Frangou, Ionita, Menegatti, Monfalcon, Moro, Papanikolaou & Pina, "Introducing robotics to teachers and schools: experiences from the TERECoP project", *Proceedings of the Cocstructionism 2010 Conference*, Paris

[2] C. Coll, "Constructivismo y educación escolar: ni hablamos siempre de lo mismo ni lo hacemos siempre desde la misma perspectiva epistemological", *Anuario de Psicología*, vol. 69, pp. 153-178, Facultad de Psicología, Barcelona, 1996.

[3] S. Chaiklin, "The Zone of Proximal Development in Vygotsky's analysis of learning and instruction." in A. Kozulin, B. Gindis, V. Ageyev, & S. Miller (Eds.) *Vygotsky's educational theory and practice in cultural context*, pp. 39-64, Cambridge University, 2003.

[4] M. Bosch, L. Espinoza, J. Gascón, ,"El profesor como director de procesos de estudio: análisis de organizaciones didácticas espontáneas", *Recherches en Didactique des Mathématiques*, vol. 23(1), pp. 79–136, 2003.

[5] J. R. Savery, T. M. Duffy, "Problem Based Learning: An instructional model and its constructivist Framework", Indiana University, Center for Research on Learning and Technology, Tech. Rep. 16-01, June, 2001.