Chapter 2

Robotics as learning object

Authors: Javier Arlegui, Nello Fava, Emanuele Menegatti,
Stefano Monfalcon, Michele Moro, Alfredo Pina

2.1 Methodological Aspects for a Constructivist Teaching and Learning of Robotics

What are the main aspects to focus on when robotics activities are developed in the classroom and how? From a constructivist point of view there are two main subjects for discussion: which “learning objects” should be built and how to carry out this constructive activity, always in relation to educational robotics?

2.1.1 Which Knowledge Has to Be Built into Robotics Education?

In this section the key focus points will be on the following questions:

- What have students to learn through robotics activities?
- Which are the “learning objects” in educational robotics?

We have to conduct at least one elementary didactic transposition exercise, this identifies what to retain from the scientific, technical and social knowledge of the robots. This keeps it as the “school-teaching-robotics” knowledge which is the “didactic” knowledge.

Robots as physical systems to be programmed

A robot is defined as an “intelligent machine” implemented as an electrical and mechanical system that can be programmed to emulate human actions.

“We build machines that perceive, understand language, have common sense, learn, and act in the world…and our hypothesis is that humanoid intelligence requires humanoid interactions with the world." (Adams et al. 2000)

The traditional interaction abilities of robots were related once to the physical world, but an interaction with the social world had to be developed as well.

“A sociable robot will be able to understand us, to communicate and interact with us, to learn from us and grow with us. It will be socially intelligent in a humanlike way. Eventually, sociable robots will assist us in our daily lives, as collaborators and companions. Because the most successful sociable robots will share our social characteristics, the effort to make sociable robots is also a means for exploring human social intelligence and even what it means to be human.” (Breazeal, 2002).

In conclusion, a “school-robot” should have the following features:

1. A robot must *act in the physical environment* and perform sequences of actions to achieve some preset objectives.

2. A robot must *interact with the physical environment* and take *decisions* about the way in which to perform its tasks, based on the perceived properties of such environments.

3. A robot must *communicate with other robots* to exchange basic information that gives it a “formal knowledge” of the environment.

4. A robot must *have some knowledge to act* and obey a structured and functional *computer program*, which describes and guides its behaviour, actions, interactions and communications.

All these four characteristics should be taken into account when discussing a constructivist way of teaching and learning school-robotics.

Actions, States and Behaviour

The *acting in the physical environment* is related to three basic characteristics of a “school-robot”: the states, the actions and the behaviours.

At a given t instant, a robot is in a definite *state*, which is characterized by the values of its “state-variables”, i.e. the Kinematics variables. These values specify the “properties” of the state (the robot properties) at that time.

The *actions* are aimed at changing the state of the robot causing a progression of the states which can be best described by verbs such as “turning”, “rising” etc.

Robots´ *behaviour* can be seen as the whole sequence of states, from the initial to the final objective state, caused by one or more actions (see figure 1).

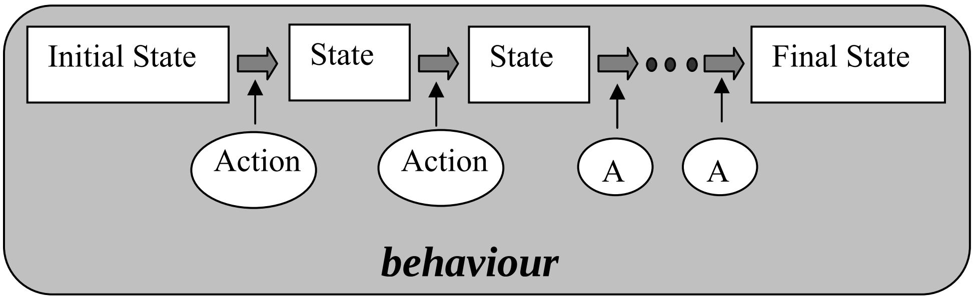


Figure 2.1.1: *Actions, States and Behaviour*

The possible actions of a robot are restricted to the physical environments and their laws. Robots do not necessarily do what students want them to do, but what they can do in interaction with the environment. Robots can “resist” the student’s task-formulations, forcing them to enter a phase of exploration. This constructivist learning is based on the student’s cognitive imbalance provoked by such “resistance”.

Physical Environment Interaction and Conditional Formulations

In human language, as well as in formal robot language, actions are always context dependent. In pre-established contexts, a robot’s task stands as an “imperative text” which is made up of a linear sequence of commanding instructions. This will be referred to as an “*action text*”. For example: In LOGO language, the task to tell the robot to walk along a square is: REPEAT 4 [GO_FORWARD :length TURN_RIGHT 90]

When the context is not predefined, we must provide all the possible contextual conditions and find suitable tasks for the robot to achieve its behaviour. We have to make a hypothetical-deductive reasoning which has to be written as a “*true-false conditional text*” which controls the contextualized *action texts* (see figure 2).

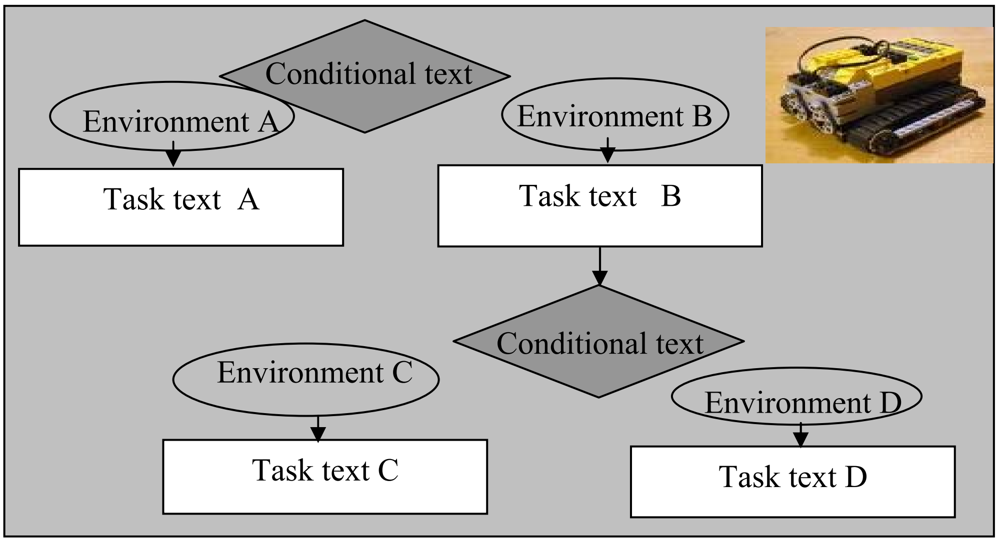


Figure 2.1.2: *a hypothetical-deductive reasoning*

Therefore, a conditional task stands as a two-level hierarchical text program, which has control instructions such as if, then, else, when, until...when the selected specific command instructions are to be executed. The conditional text constitutes a meta-text of the action text and imposes a no linear linguistic structure to the global text, with a hyper-textual or hierarchical character and a higher level of complexity. Robots must be prepared to do tasks with a flexible and context-adapted behaviour, thus robot programming must include these conditional properties as an intrinsic aspect. This will also require a new level of constructivist activity by the students, which will be shown below.

Communication amongst Robots and Levels of Communication

Although this is not a primary goal of the project, social robots are an extension of individual physical robots, creating bigger potential for teaching and learning situations in Primary and Secondary schools with the use of constructivist methods.

Any system of multiple robots is characterized by robots that operate in the same environment in a cooperative way with a view to achieving a global goal. This cooperation consists in the exchange of significative information among robots, whereupon their individual behaviour is regulated.

As Jiménez, Ovalle and Branch say (Jiménez et al. 2008), in a multiple-robot system there can be two kinds of communication.

A non intentional or indirect communication is either where information is transmitted by modifications of the environment made by the robots, or by modifications of the robots. In this kind of communication, messages are not received specifically and, therefore, there is no guarantee of reception. For example, in a "sumo" competition, a robot must try to “communicate” with the other robot, looking for where it is through distance or contact sensors.

At this level, there is no established channel; instead, there is just communication by external symbols produced by the robots. Robots must be fully capable to distinguish each other. This kind of communication is only useful in solving problems that do not require intentional coordination. However, it is very important that robots do not interfere with each other.

An intentional or direct communication is the second type. Robots use specific communication channels to communicate effectively. Messages have well-defined transmitters and receivers, which can be differentiated at two levels, depending on the complexity of the communication.

Communication based on states: the values of the robot internal state-variables constitute the important information. These values can, in turn, be obtained from the environment and can be used to “teach” other robots about them. For example: A robot adapted to locating seats of fire along its linear path can send a message to another robot telling it the position and the temperature of several fire seats, encoding this information as a list of sequential two values positions, one for the X position of the robot (in cm) and the other for the T temperature (in ºC) given by a sensor (when the temperature values are greater than 70 ºC). Within this protocol, the message [[110 80] [165 75] [240 85]] means that there are three fire seats (with 80, 75 and 85 ºC) at distances of 110, 165 and 240 cm from the origin.

Communication based on shared objectives: the information is made up of more complex texts that try to emulate the communicative roles, the typologies and textual functions of natural languages. For instance: The communication between the “She-Duck” robot and her three “Ducklings” robots could be, with a previous known protocol with acknowledge, as follows:

```
BROADCAST MESSAGE [FROM She-Duck TO [Duckling1 Duckling 2 Duckling 3]
FOLLOW_ME]
MESSAGE [FROM Duckling 1 TO She-Duck OK]
MESSAGE [FROM Duckling 2 TO She-Duck OK]
MESSAGE [FROM Duckling 3 TO She-Duck OK]
```

The present school robots (as the ones used in the TERECoP project) are able to set up intercommunication possibilities amongst themselves. There are university projects with an advanced development on this subject[i]. The latest developments are those referring to a constructivist methodology about “social robots” in Primary and Secondary schools (Picard et al. 2004).

Knowledge, Task formulation and Programming

The students can not directly act on a robot. They act on a representational space; they build a text, and this text controls the behaviour of the robot. In a computer environment, “writing is acting”. We label the text as “tasks” that command robots´ behaviour. A task is written as an instruction (or a sequence of instructions) in a computer program. One task creates a particular kind of behaviour (see figure 3).

In turn, the specific tasks come from “generic tasks” that command “generic behaviours” (or class-behaviours). These generic tasks are implemented as *procedures* in a computer language and become the real robot “knowledge”. Teaching and learning to build up these procedures will be the main objective of our constructivist approach.

The Nature of the Formal Objects that are built in the Programming for Robots

As we have seen above, the school exploration with robots is the formal exploration carried out by students to construct "good programs”. But, what are the formal objects that the students build when they program? And what is their nature? We are going to approach this from a didactic point of view, which is particularly interesting and well adapted to our objectives. This is the theory of "praxeological organizations» (Chevallard, 1999).

Behaviours: **T**he framework of the Teaching Anthropological Theory (TAD) postulates that human intervention (and robot intervention) in the environment is achieved through the implementation of specific actions aimed at a particular target in a given context. We call these specific actions “behaviour”. For instance, drawing a square of 20 cm per side by a pupil is a concrete behaviour.

Tasks: The behaviour can be strictly private, but often the behaviour is culturally pre-established and can be formulated and communicated. The formulation of a particular behaviour aimed at a specific goal is what we call a "task". The teacher's request "draw a square of 20 cm per side on the board" would be the task, in the above example.

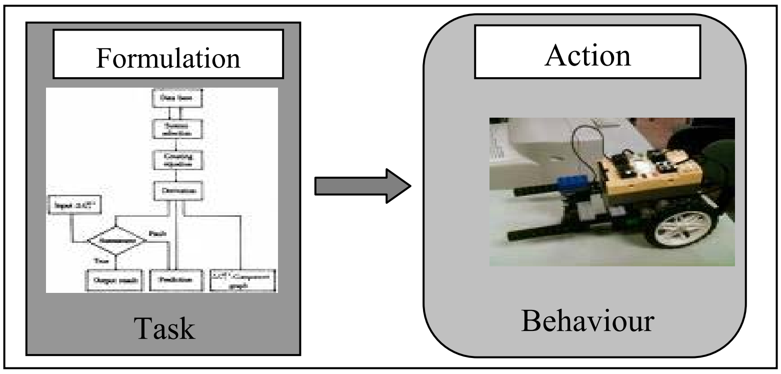


Figure 2.1. 3: *Praxeological Organizations*

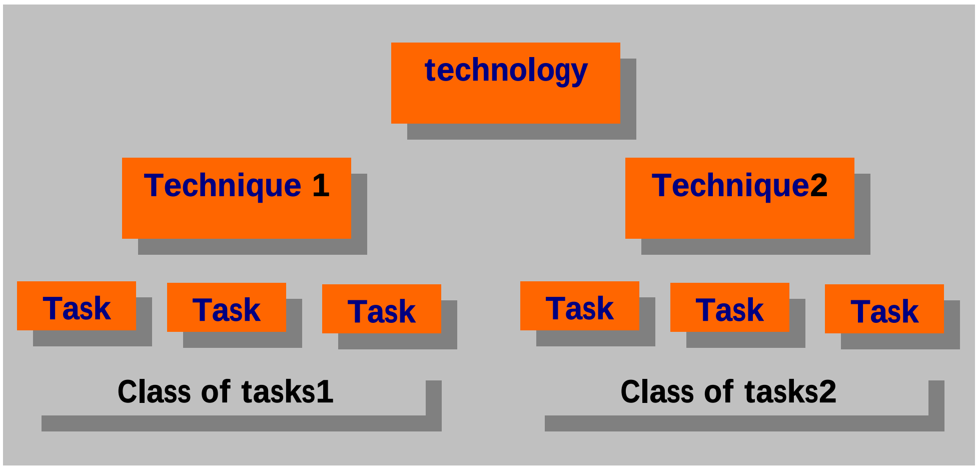


Figure 2.1.4: *Praxeological Organizations*

Classes of tasks: The social experience of human beings groups together cognitive and linguistic tasks into "classes of tasks". One class of tasks brings together tasks that are "similar" in the sense that they have similarities and differences. They have the same values for certain properties (the identifying variables) and different values for others (the discriminatory variables or state-variables). "Draw a square" is a class of tasks. The length of the square is the discriminatory variable (with different values for each square), and the number of sides and the angle are the identifying variables (always number of sides = 4, and angle = 90).

Techniques: The texts describing how to resolve a class of tasks are called texts of technical know-how, or simply “techniques”. In the above example: “Repeat four times [go forward a D distance and then turn right 90º]” is a technique.

Technologies and Theories: We can think of a meta-text on techniques, that is, a text that expresses “know-how” about a class of techniques. We refer to this text by the term “*technology*”. In the above example, once the techniques to draw squares, rectangles, diamonds… are known, we can build up the technique to draw regular polygons. In the process of making meta-texts more and more inclusive and abstract, we can formulate more academic texts. We refer to this whole text structure by the term “*theories*”. In our project, it is enough to think about the elementary levels of the theories: techniques and technologies.

Implications of the Praxeological Analysis

In the framework of the praxeological organizations, we will say that a human being and, by extension, a robot, performs a socially "intelligent" action, when it corresponds to a task that has been formulated and specified on the basis of a technique.

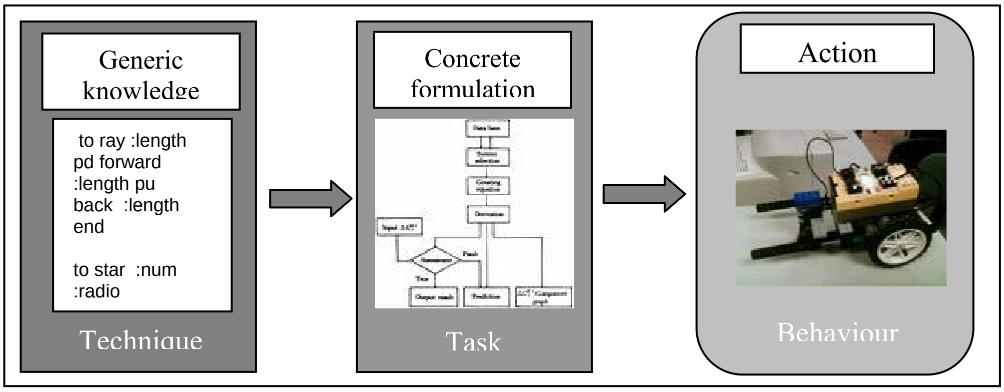
When one learns and uses a technique, the tasks associated with it are no longer "problematic” to solve, but simply “tasks” to accomplish. They give rise to the “intelligent” behaviour, as opposed to the stuttering actions corresponding to unspecified "trial and error" techniques. Learning to solve a problematic situation means moving from a problem-situation to a task-situation and needs the previous acquisition of the proper technique, that is, the building-up of a know-how text.

If we get this generic knowledge, if the technique is understood, it can be used to write the “formulated-action” adapted to the problem (the written task) that will originate, finally, the “physical-action” (the behaviour). Thus, constructivist learning in robotics is basically the pupils´ construction of robot techniques, the construction of techniques related to class-tasks (generic solving methods).

Textual Aspects of Techniques and Tasks

Techniques are written as “general propositions” and must be expressed in terms of generic class with state variables representing the formal parameters of the technique. Techniques are sentences such as *Turn “g” degrees*; *Lift up the arm “y” cm; Forward “x” meters with “v” meters per second…*

Tasks, by contrast, are “local propositions” that instance a technique for a particular case by specifying the actual parameters. Tasks are sentences like: *Turn 360 degrees; Forward “2” meters with 0.5 metres per second …*



```
 to ray :length
pd forward
:length pu
back  :length
end

to star  :num
:radio
```

Figure 2.1.5: *from techniques to behaviours*

Summarizing…

- Robots´ knowledge is implemented as a computational technique (The LEGO NXT blocks are excellent examples of techniques).
- A robot can have “knowledge for action” implemented as computational command techniques.
- A robot can also have “knowledge for decision-making” implemented as computational logical techniques.
- A task is implemented as a computational instruction.
- A task causes a robot’s behaviour. A simple task causes basic computer behaviour and a computational program with a sequence of instructions causes a more complex behaviour.
- The invocation of procedures within the new procedures allows a hierarchical -“praxeological-like” structured knowledge.
- It can be seen that the capacity and interest of a language like LEGO NXT expresses techniques and tasks which build up praxeological structures. These are well adapted to a constructivist teaching and learning process of problem-solving.

2.1.2 How to Carry out Constructivist Learning and Teaching on School Robotics?

Constructivism as a Learning Process

The expertise in commanding tasks for robots is such that they have certain behaviours with a goal in mind and can be the object of constructivist education on the teacher's side and learning on the student's side. It is necessary to select and adapt to our objective the most pertinent characteristics of the theories of Piaget and Vygotsky, known as cognitive reconstruction theories and assuming constructivist teaching and learning.

Remembering Piaget…

The Piaget theory is a theory of the *dynamic construction* of knowledge. Piaget bases this construction on the process of "improving" adaptation that is formulated as the tendency to a growing equilibrium (balance) between the processes of assimilation and accommodation gets under way.

Assimilation:
*The process by which people interpret the information coming from the environment, depending on their available conceptual structure***s**
scheschemes available"

Accommodation:
The process of modification of conceptual structures by people, when they try to assimilate new characteristics of the environment

The *assimilation* suggests that "we see" all things not as they are, but as "we are", according to our available patterns of understanding. We only incorporate from reality those inclusive elements that can be recognized by our previous *schemes*. If only assimilation was there, much of our knowledge would be unreal fantasy and would lead to constant mistakes.

The *accommodation* explains the tendency of our schemes of assimilation to adapt themselves to reality, becoming more "consistent" (or balanced) with it. If these schemes are insufficient to assimilate a given situation, they will probably require future modification to interpret additional characteristics of the situation. The accommodation supposes not only a modification of the previous schemes based on the assimilated information, but also a new assimilation or reinterpretation of the previous knowledge based on the new constructed schemes. It is what we call "reconstruction" and is the most important effect of the constructivist process.

Teaching and Learning as a Process of Successive Re-equilibration

When students first start contact with new knowledge, they are generally imbalanced. They apply their previous cognitive schemes and usually assimilate only part of the aspects of the object. To develop a progressive adaptation, the student has to do a double work: a *direct empirical interaction* with the object and *a linguistic interaction* with a teacher (in reference to the object). This will facilitate the student to have a progressive adaptation both to the understanding of the actions of this object and to the understanding of the terms of the language with which we describe these actions.

This will take students to a first *re-equilibration stage*, but a further interaction with the object and / or a new problematic question of the teacher regarding the object will lead students to a new situation of *disequilibrium* that must be overcome through the same procedure to reach a new state of equilibrium (see figure 6).

The teacher's role is to trigger successive "controlled" imbalances to the students through "controlled" questions, being careful in order not to introduce too many new features in each question, and then helping them to reach a new re-equilibration stage with "demonstrations" in the Vygotskian way, showing "well realized" actions on the object, and "well formulated" linguistic expressions referring to the object.

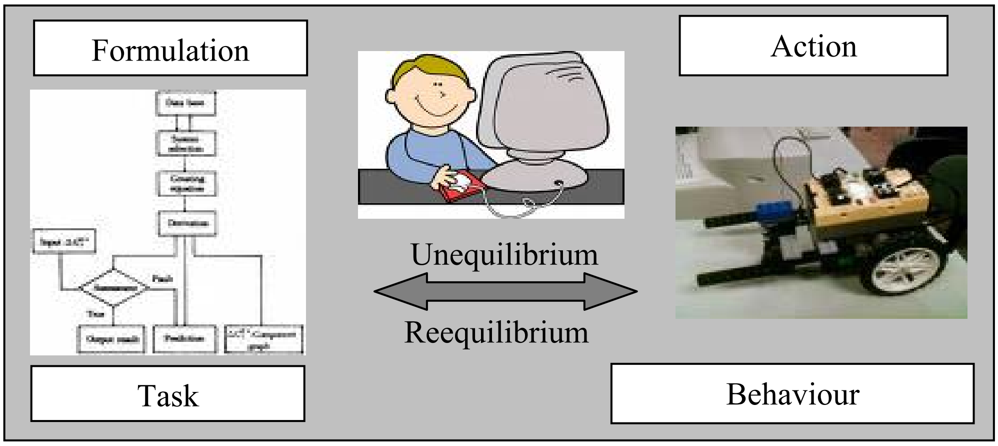


Figure 2.1.6: *the interplay between disequilibrium and equilibrium*

The role of the student is essentially to be intellectually "active" in the process, striving to identify new "inclusors" (new cues of reference) in the previous schemes and trying to give significance to the "demonstrations" of the teacher.

We must emphasize that when the students have to deal with "linguistic objects", for instance, to construct adequate programming "techniques", the individual constructivism of the student does not fit. On the contrary, it is necessary for a student to engage with "guided" constructivism and act for a "double" reequilibration: the student interacting with the object for a "semantic" feedback while talking with the teacher for a "syntactic" feedback.

Levels of Complexity of the Re-equilibration

Piaget elaborated on several models of the equilibrium process (Pozo 1989). In the latest of them, he says that the equilibrium between assimilation and accommodation takes place at three levels of increasing complexity:

- The first level is *equilibrium with the facts*. The initial schemes must reach a balance with the new objects to assimilate.
- The second level *equilibrium with the schemes* is a balance between old and new schemes that have to be reached in order to be assimilated and accommodated to each other.
- The third level *equilibrium with the hierarchic structure of schemes,* where new re-equilibration of the schemes hierarchical structure should be achieved.

Class Problems for a Constructivist Teaching and Learning Process in Robotics

The robotic education should proceed with the *formulation of successive problems* grouped as class-problems:

a. *Problems on the same class of behaviours (robot actions),* which must give origin to the same class of programming tasks, from which "the best" technical programming procedure (a "technique") must arise. For instance, to instruct the robot to walk along a square of 20, 30, 40, … cm per side is a class of tasks.

b. *Problems on a new class of behaviours*, close to the previous one, but slightly different, from which a new technical procedure, different from the previous one, (a new "technique") must arise. The comparison and contrast between those two techniques should create a "technological" piece of knowledge. A contrast between the technical procedures to build up squares, rectangles, diamonds,… could be lead to the construction of a more general procedure to move a robot along a generic parallelogram (having the length, the width and the two angles as parameters), which is a technological procedure.

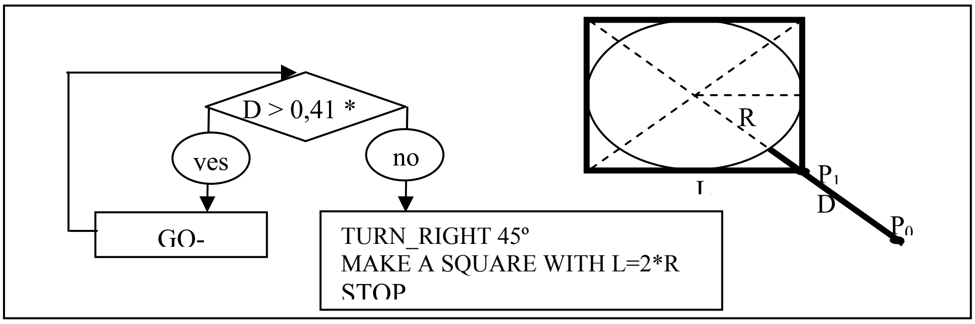


Figure 2.1.7: *a conditional procedure*

c. *Problems on a conditional (context adapted) behaviour* for a class of contexts, from which a conditional procedure should arise to control the conditional tasks to be done, building the "knowledge for decision-making" mentioned above. For example, making a robot move along a square circumscribed on a circumference of radius R, starting the robot initially at an external point P_0, is a conditional problem. To solve it we have to write a conditional procedure (see figure 7): While the robot is approaching the circumference from P_0 (measuring its distance D with an infrared sensor) we have to decide if it has already reached the P_1 point to start then the square route.

2.1.3 Designing Didactic Situations in a "Constructivist" Robotic Course in the Classroom

The Curricular Dimension of Constructivism

For strict consistency, a course that focuses on constructivist learning should lead a constructivist methodology. This methodology, as we shall see, should be focused on learning based on problems and projects. What does a "constructivist course" mean? A "constructivist" model usually emphasizes the *learning* process and thus the role of the learner. Learning must be active, meaningful, through inquiry, and the students are seen as the builders of their own autonomous learning.

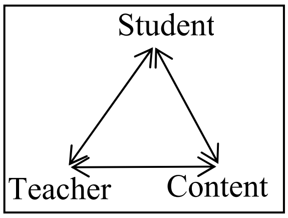


***Figure 2.1.8**: student's relationship with teacher and content*

Often, little is said of the *teaching* process and the teacher's role, which is regarded as that of a mere facilitator of the above process ("…It's enough to provide the students with rich environments" in order to ensure their constructivist activity). Also, little is said about the role of content (curricular subjects), as if "anything goes" if multiple learning activities can be designed around it.

This expression of constructivism is primarily a *psychological proposal* on the methodology of the student learning. This is especially true in "natural" contexts (pupils learn from the environment without a teacher), more than in school contexts. It is *not a curriculum proposal* because it focuses on the student autonomous work and neglects the student's relationship with the teacher and the content (see figure 8), relationships that inevitably exist in school education. Constructivism in school is not only a matter of the student (or the social group of students) but a matter of the students in relation to the teacher and the content. These relations are to be made explicit in a curricular design (deciding what and how to teach, and how to learn).

The Importance of Formulating the "Problem":The problem is at the core of the constructivist activity

One must question the role of teachers in this type of course. As we deepen in the root of the child constructivist process, we see that Piaget identified it with the process of “unbalancing” and “rebalancing”, associated with any new adaptation. The child constructs rebalanced new schemes with an environment that previously had changed and had become problematic. The construction process is the gradual process of "*re-equilibration*”.

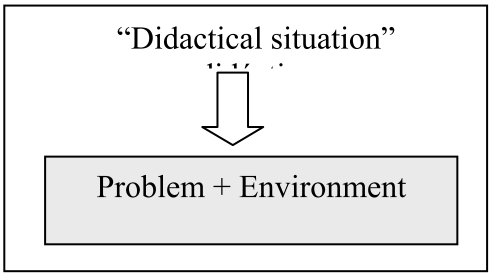


Figure 2.1. 9: *a specific exploratory interaction triggered by the problem*

In science and technology in school, how is the "environment” presented to students? It is a "pack" of two elements: a physical environment and a problematic question concerning it. It is not merely an environment that allows the student to make random interactions, but an environment that is offered for (see figure 9). This pair (problem and environment) is known as "didactical situation" and is the teachers´ duty to design and present it to pupils. Sometimes we refer to it simply as the "P problem."

Hence, the importance of this problem, which will guide the student exploratory-constructive activity, should be carefully planned by the teacher taking into account the following aspects:

- Any problem formulation should refer to aspects already known by the students and must contain one or more elements of novelty that cause an imbalance (disequilibrium) in the students. Any new problem should start below the last problem knowledge level so that the "cognitive distance" between problems (A, B, C ...) is small.
- Although in many cases it is difficult to predict the quantity of novelty the problem has, it should be just enough to allow a resolution by independent exploration on the part of the student (enquiring work). In most cases it should

allow a resolution by working in Vygotsky’s "zone of proximal development" with the help of the teacher, for example by developing activities undertaken jointly with the teacher, in which the teacher progressively transfers responsibilities to students helping them to solve the problem. In most cases teachers should play an active part in solving the problem.

In the diagram of figure 2.1.10:

- The student is initially at level A, equilibrated with a particular environment located at an A level of understanding in front of a given phenomenon. For example, a student can understand and skilfully use the “move” block (in Lego Mindstorms NXT-G programming language) to have the robot following different linear routes with different values of power and time.

- At the moment t_1, the teacher suggests the problem (P1), which requires the student to understand the environment from a new point of view (level B), in which the student does not fit. At this moment we say that the student is in a new didactic situation where he is imbalanced. For example, the teacher can ask the student how to reformulate the primitive “move” block to make a “my_move” block with only two parameters (power and time). This is a problematic task for the student that has no idea how to generate this new user block…

- From t_1 to t_2 the student initiates an active process of re-equilibration (assimilation and accommodation) that progresses to a new level of understanding, where the problem P1 leaves gradually the status of a "problem" and is becoming a "task". For example, the student may start to look for help on the NXT-G environment, reading about “my blocks and how to create them or starts to imitate the way the teacher acts when he shows him how to generate a “my_move” block…

- At the moment t_2, the student recognizes and solves the problem (P1) in a consistent manner. We say that he has achieved a re-equilibration with the environment at level B or simply that he has learned to solve the problem P. For example, the student finally generalizes the technique of how to generate a user block from pieces of previous blocks. Gradually the student becomes skilful in this technique using it in a class of programming situations. How to generate user blocks is no longer a problem for him, but a task to accomplish.

The Class-problems and the Generic Techniques

It should be remembered that, although for simplicity reasons we have talked about “problems” in the previous paragraph, we should always speak of "classes of problems." A “unit of learning” is built only when students give solutions to a set of problems of the same class. We say that we have found a *technique* (technical knowledge) to solve such class of problems.

It is, therefore, the teacher's responsibility to ask questions that are "class problems" of some generality, the answer to which requires the resolution of several "punctual problems" (P1, P2, P3 ...) of the same class. When the problem concerns an "artificial" environment linguistically controlled, such as a robot, writing new "techniques" for the robot requires a dual activity on the part of the student:

- An initial deductive enquiry activity explores (from the existing procedures) the instructions I1, I2...to solve the punctual problems P1, P2, (from the same class).
- A final inductive enquiry activity showing findings from I1, I2…and a general formulation to solve the class-problem P (integrating P1, P2…).

Building a technical knowledge to solve a class-problem involves for the students a process of successive approximations. These contribute to the improvement of the global adaptation from level A to level B, which is achieved when students construct a generic technique that allows them to solve any problem associated with the class problem.

Learning Objects and Tools for Learning

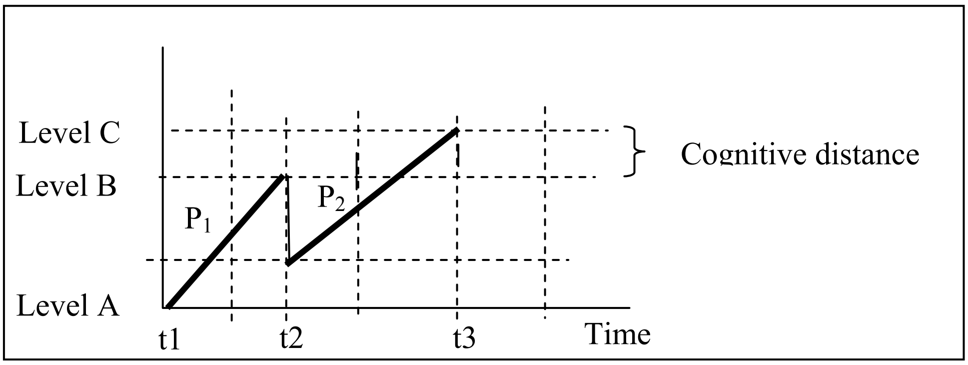


Figure 2.1.10: *an active process of re-equilibration to a new level of understanding*

Any object has potentially a double status: it may be taken as a "knowledge object", when we are studying and learning about it, and can be taken as a "knowledge tool", when we use the object (making it part of the interactive environment) to learn about something else. In the case of the robotics-based learning, the dual role the robot plays in the learning process seems clear:

- *The robot as a learning object* (building knowledge about the robot): The robot, as a physical and programmed object, should be the focus point of knowledge. Therefore, in a training course we will find class-problems related to the characteristics of robots. They will be sequenced gradually in topics depending on the increasing complexity of the robots movements and functions and their diverse sensors.

- *The robot as a learning tool* (building knowledge with robots): On the other hand, robots can be used to gain knowledge of certain characteristics of the physical environment.

The stages in a constructivist learning process

From the above we can afford to separate the process of building a constructive course in two components:

The task of the teacher: The teacher is responsible for

- the formulation and proposition of the problems
- the "Vygotskian" help to the students' learning
- the institutionalisation of knowledge emerging from the students

The task of the students:

- the "enquiry effort" to build up a piece of knowledge from a "didactical situation"
- the search for meaning with the building of "learning skills" ("know-how" learning),

Thus, a basic minimum teaching unit in this constructivist process should be able to show the following steps:

-*Formulation of a class-problem by the teacher*, referring to a class-behaviour of the robot.

Constructing particular solutions by the students, writing different tasks to accomplish different behaviours.

Generalizing the solutions: building up a procedure.

Empirical validation: getting evidence of good "behaviour" of the procedure.

Using the procedure: to solve new problems of the same class.

Looking for the limits of the procedure and adapting it to new-type problems.

An Example

a. Formulation of a Class-problem by the Teacher

Modern automatic trains (without drivers) stop after precise distances between stations... We could build a robot that acts like an “intelligent” train. To do so:

How can we make a robot travel distances D pre-established...?

b. Constructing Particular Solutions

task	distance AB (cm)	degrees
	0	0
1	50	1033
2	100	2070
3	150	3106
4	200	4138

Figure 2.1.11: table and diagram of four tasks drawing linear routes with different distances

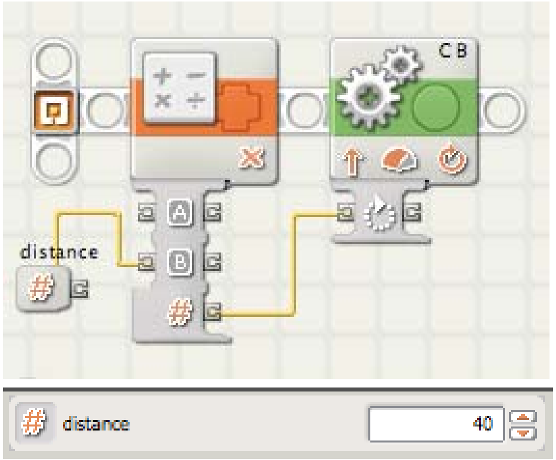
A group of pupils (Primary school) tentatively propose to use the primitive "motor" instruction, implemented in NXT-G, and initially decide to change the "degrees" (leaving the "power" with a fixed value).

They prepare a series of four tasks (of the same type), drawing four linear routes on the ground with different distances each: distance 1 = 50 cm; distance 2 = 100 cm; distance 3 = 150 cm; distance 4 = 200 cm. Students begin planning, by trial and error, various trips for each robot with different values of the "degrees" parameter until they get a trip with the desired distance.

They obtain finally the data shown in the figure 11 and draw their graphic representation. From the four “local” solutions the students can generalize that: *"the robot requires a value of 2070 degrees per hundred centimetres it progresses”* or, proportionally: *"the robot requires a value of 20.7 degrees per one centimetre it progresses”* and as a result: *degrees num. = distance in cm. x 20.7*

c. Generalization of Solutions: Development of a Procedure

With this "knowledge" students can now program the robot in a general way. They write the procedure called "*move_a _distance*", which takes as parameter the value of the variable "*distance*."



distance 40

Figure 2.1.12: *Development of a Procedure*

d. Empirical Validation

The students program a *move_a _distance* instruction with the value *40* as the distance parameter. They check the behaviour of the robot and measure the distance it travels and confirm that it travels 40 cm with high precision. They program different *move_a _distance* instructions with different distance values, confirming the good robot behaviour in any case.

e. Using the Procedure

From now, the students can use this procedure “as a tool” to make the robot move over specified distances, integrating it in other problems and projects.

f. Generalization of the Procedure to new Class-problems

What happens if we change the wheels of the robot? What happens if we simultaneously change the value of "Power" on the engine? These are examples of new class-problems that lead, by a similar sequence of steps, to construct a "second level” (based on two parameters) more general procedures.

References

Adams, B., Cynthia Breazeal, Rodney A. Brooks, Brian Scassellati (2000), "Humanoid Robots: A New Kind of Tool", IEEE Intelligent Systems and Their Applications: Special Issue on Humanoid Robotics, Vol. 15, No. 4, pp. 25—31.

Breazeal, C.L. (2002), Designing Sociable Robots (Intelligent Robotics and Autonomous Agents), Ed. Massachussets Institute of Technology. Boston.

Chevallard, Y. (1999), Teachers training in the Didactic antropological theory analysis (El análisis de las prácticas docentes en la teoría antropológica de lo didáctico), Recherches en Didactique des Mathématiques, Vol 19, nº 2, pp. 221-266.

Jiménez Builes, J. A. Ovalle Carranza, D.A. Branch Bedoya, J. W. (2008), Comunicación en sistemas de múltiples robots desde la metodología MAD-Smart, REVISTA INGENIERÍA E INVESTIGACIÓN, vol. 28, num. 2, pp 59-65.

Picard, R. W., Papert, S., Bender, W., Blumberg, B., Breazeal, C., Cavallo, D., Machover, T., Resnick, M., Roy, D., and Strohecker, C. (2004). Affective Learning — A Manifesto. BT Technology Journal 22, 4, pp 253-269.

Pozo, J.I., Cognitive learning theories, Ed. Morata, Madrid, 1989.
