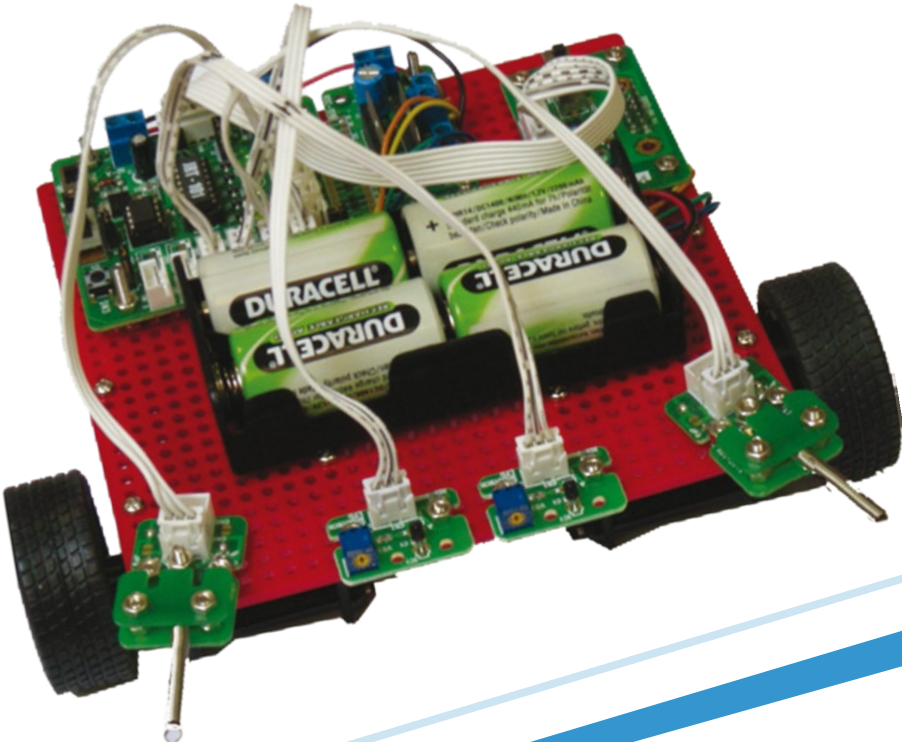


Education and Culture

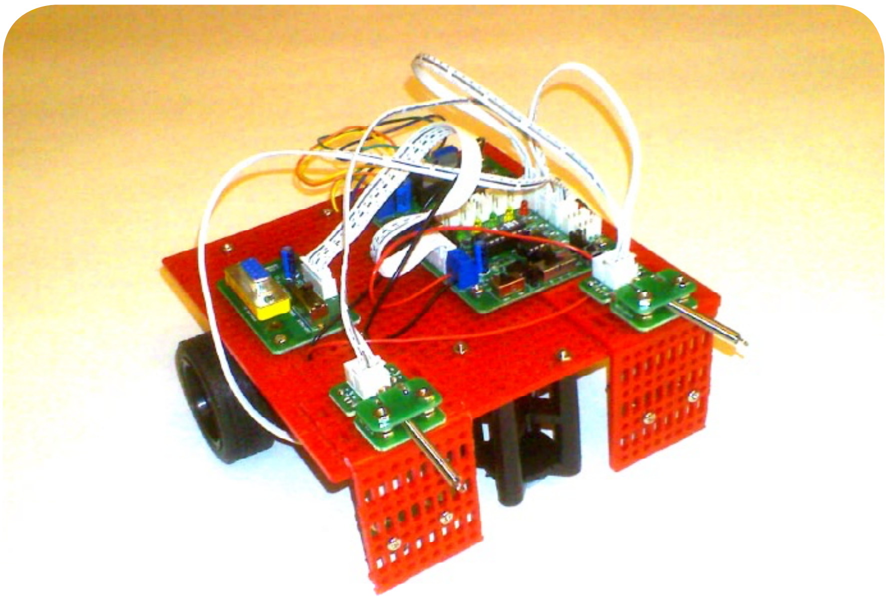
Socrates
Minerva

RoboDidactics Manual

by
Peter van Lith
James Caska

Graphic design by Alessandro Romani,
Fondazione Mondo Digitale

Contents



Summary of changes

The table underneath lists the status of the document and indicates the changes made in the course of the development of the methodology.

	Changes	Planned actions
1- *3-*	*Original version*	*Review and modifications for version 1*
	Review from local experts, added new information: • *Planning information* • *Relationship with other disciplines* • *Required preconditions* • *Background information*	*Review and discussion.* *Basic determination of additions required for version 2*
	Suggestions from some experts included • *Included experience gained during testing of early material*	*Made numerous changes* *Create first version of RoboDidactics manual* *Created first version of educational material*
	Extensive rewriting by Prof. Alfonso Molina to transform the document into a manual concentrating on the fundamental educational concepts of the RoboDidactics Methodology	*To be given out to pilot schools and the IPE for testing and comments*

This third version of the RoboDidactics approach forms the RoboDidactics Manual and explains the details of the RoboDidactics Methodology and associated hardware and software robot environments. It contains information about the structure of the lessons and teacher's material based on the backgrounds of the methodology. This third version is intended as the starting point for the workshops and testing sessions.

Introduction

This document explains the educational thinking informing the Robodidactics Methodology aimed at stimulating the didactic use of robots in schools. The methodology will enable children to learn important concepts of engineering, physics and ICT in a stimulating and entertaining manner. It will also offer examples for the didactic use of robotics in subject matters across the school curricula.

The RoboDidactics Methodology builds on earlier work done by Lego and, subsequently, adopted and extended by the AMSTEL institute, an educational institute at the University of Amsterdam. The work is part of an ongoing project aimed at providing technology education to teachers of children of ages 9 through 19 years. RoboDidactics borrows from existing educational material and several examples from the exercises developed for the world-wide RoboCup Junior initiative.

The document also explains the hardware and software environments used in the development and implementation of the RoboDidactics Methodology. It closes with a brief discussion on the testing and wider dissemination of the educational concepts of the methodology, along with the identification of basic research questions on the didactic use of robotics in schools.

Why we are doing this

In many European countries the number of students, selecting technical education is decreasing steadily to a level where industry expects to hit real problems in finding sufficiently qualified personnel in the coming years.

The exact reason for this decline is not known and many organizations are developing initiatives to improve the situation. Solid research into the causes of this phenomenon is yet to be done. Nevertheless, some things are clear enough to demand the adoption of specific measures.

Several initiatives exist seeking to make technology more interesting by offering a playful environment, while at the same time creating educational material to show that technology is not boring.

Getting interested students to participate in these initiatives is not so difficult; getting on the other hand, involving children who are clearly not interested in technology is much more difficult.

One approach to involving all types of students has been to embed an element of robot competition into the school curriculum and inviting the entire class to participate.

In the case of girls, an important element of attraction is the stressing of creativity and societal relevance implied in the use of robots. Several groups have developed educational material in this direction and RoboDidactics benefits from these efforts.

In countries where this more entertaining and relevant didactic approach has been running for over 8 years now, the first signs of an increasing influx of students into scientific and technological university careers are now becoming clear. RoboDidactics adds to existing efforts a didactical methodology that further enhances the various initiatives.

New educational material is being developed and will be tested in the course of this project is under development that will be tested and improved in interaction with schools in several European countries.

The RoboDidactics approach

Education using robotics

Over the past years, education in technology has suffered from a pedagogical approach excessively abstract and theoretical which, at least in several European countries, has led to a decrease in the number of students that select a technical education. This implies that technical education in general and Information Technology (IT) in particular have to be made more interesting and challenging than is currently the case.

In many cases, IT education starts with training in the use of programs like Word, Excel and PowerPoint. In some cases students are encouraged to develop their own websites and sometimes they are taught how to start programming computers in languages like Basic or Java. Although these approaches are aimed at teaching important concepts, they are either too simple or too complex. Many children are quite familiar with the use of browsers and office software like Word and find education in these matters unnecessary.

On the other hand, teaching how to program is often intimidating and scares off many children. The children that are attracted to programming commonly already know more about it than most of their teachers.

The RoboDidactics Methodology proposes a learning environment that concentrates on three different levels that grow gradually more complex and detailed, ranging from the use of very simple intuitive instructions to a full programming course. Starting at a very elementary theoretical level means doing very simple things that might be boring when using a traditional approach.

The RoboDidactics Methodology starts with simple things on a very high intuitive conceptual level that result in interesting experiments right from the start. From here complexity grows gradually while maintaining the element of entertainment in such a way that students face a single, integrated approach that allows them to learn some basic concepts and then gradually reach deeper and deeper levels of understanding, building upon the knowledge they have already acquired.

Background of the approach

The basic assumption is that making ICT and other technically oriented education more interesting and playful will stimulate more children to select this type of education. The application of this idea resulted in the creation of the International RoboCup society that aims at involving more children in technical education through the use of small, autonomous robots in several challenging competitions.

Interactive Lego Football (ILF) and Autonomous Lego Football (ALF)

The University of Amsterdam (of which the AMSTEL institute is a part) is participating in the world-wide RoboCup challenges and organized the 2000 championships in Amsterdam.

During this event Ben Kröse organized one of the first RoboCup Junior competitions in Amsterdam, with the help of prof. Lund of Lego and others. In this event an approach developed by prof. Lund, called Interactive Lego Football, was used. This program allowed children of age 7 and higher to develop programs that would play a game of soccer, using a Lego MindStorms robot. The approach is also known under the name of 'User Guided Behavior' and makes it possible to mix the children's insight with preprogrammed behaviors.

Figure 1
Layout of ILF Input screen

Regrettably Lego stopped development of this program in 2001. At the same time students of the University of Amsterdam developed software in Java that provides a real-time simulator that allows students to experiment with small robots, playing the different games of the RoboCup Junior challenge. Since these games are aimed at children between 9 and 19 years old, they offer an increasing level of complexity.

In doing this, the educational approach followed makes sure that children start early and are able to reach their first results quickly, thus demonstrating that making a working program is much easier than they think. With a program like ILF they reach this low-complexity early stage in a few days, sometimes even in hours.

Three-level Learning Environment: Conceptual, Physical and Programming

The RoboDidactics learning environment defines three competence levels that children need to master perhaps gradually in the course of several years. These levels are: the conceptual level, the physical level and the programming level. This stepped approach is part of a consistent teaching strategy with accompanying software and has a relationship with the three main class levels at school (see next section).

Thus, the "conceptual" robotic courses cover primarily (not exclusively) the elementary school level (from age 9 to 12), with the "physical" robotic package primarily for the middle school levels (12 to 16), and the "programming" robotic courses covering the middle and higher school levels (16 – 19). Of course, this is indicative because it is possible to conceive that some non-technical higher-school classes such as Literature, Philosophy or Music may well use the Conceptual level of the RoboDidactics Methodology for didactic purposes.

The basic Conceptual level concentrates mostly on achieving quick results in terms of mastering basic concepts. The middle physical level mostly concentrates on teaching important concepts from ICT, engineering en physics. The use of a variety of sensors establishes a firm link with physics and allows experiments with sound, light and movement. However, the most basic concepts covered by these courses relate to the principles of Control.

Although some countries do address Control technology in their official school curriculum, for instance the UK, most European countries do not include Robotics and Control as part of their educational programs. Clearly Robotics can be addressed at different levels in the school curriculum, starting at age 9 (primary school level) up to upper secondary level (age 16-18), for instance in the subject of Information Technology and Physics.

The RoboDidactics Methodology aims at helping teachers to realize these ideas in their schools by taking students from basic to complex robotics concepts in an entertaining and practical manner. Let us see how this works at each one of the levels in the methodology.

The Conceptual level

At this starting level, children program their robot with simple commands that they can easily relate to their own world. The idea is taken from the Lego program Interactive Lego Football (ILF) where a range of simple commands is provided. Children select the appropriate commands from a repertoire and string them together to form a program that allows a robot to play simple football. By imagining they are robots themselves, children may compose a series of commands that perform the football behavior in an experiential way.

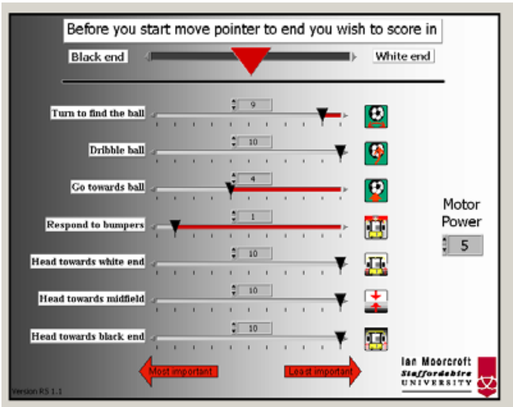
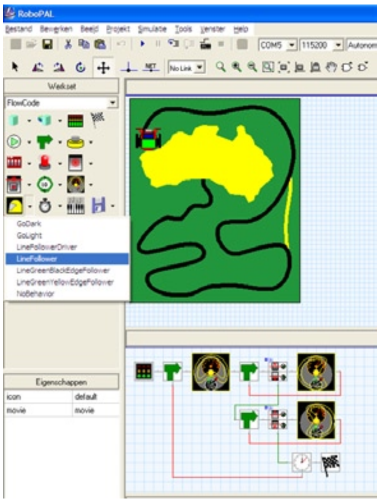


Figure 1
Example of input screen for RoboSoccer (ALF)

The conceptual level starts by concentrating on what needs to be achieved with the robot. The RoboSoccer program can be played with simple intuitive commands like: "find the ball", "go towards the ball", "kick the ball" etc.

The RoboDidactics Methodology aims at building upon this approach. Today specific platforms do not allow the students to progress through the three levels we have identified. The RoboDidactics approach will contribute to change this situation. .



The plan is to start with the Rescue challenge. This is a simpler exercise than football and consists in rescuing a puppet that is drowning in a swamp.

Here the computer-based learning environment offers a variety of basic line followers. The children have to fine-tune these basic behaviors and string them together to form a working program.

Figure 1
Layout of the rescue field and a collection of line following behaviors

The important thing during this conceptual stage is that the concepts learnt should be translatable to the physical level, so that the children are able to switch back and forth between these twolevels and can actually see how, for instance, the concept of "moving forward" is translated into commands that make the motors spin in the correct direction.

Today, no existing software allows this to be done. Thus, to enable the application of the RoboDidactics Methodology, an integrated environment, using a simple iconic graphical language, called roboPAL is currently under development by a commercial company who will provide the software as a product to those schools or other organizations piloting and/or applying the Robodidactics Methodology.

The Physical Level

At this level, once children have become familiar with the basic way to control their robot, they advance to learn how the commands in the conceptual level are actually executed. This is made clear in the form of an iconic language a little more technical than the iconic language of the conceptual level. With the "physical" iconic commands the children learn how to steer their robot, how to find a line on the floor and how to find a ball. By putting together iconic commands they learn to control the properties of their robot. The commands from the previous level can then be seen and programmed in the more detailed iconic environment.

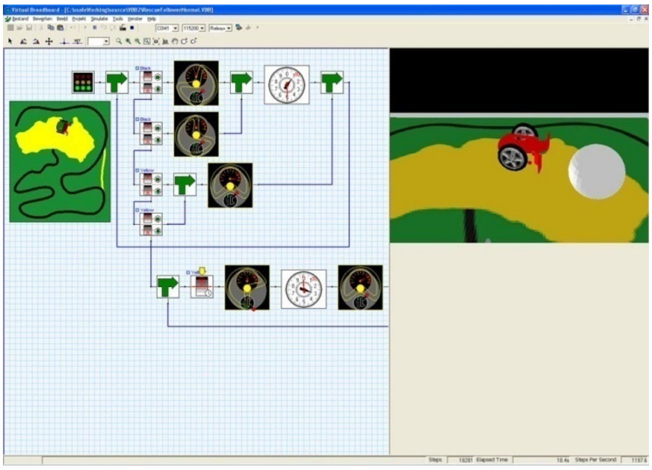


Figure 5
Screenshot of roboPAL running a line follower for RoboRescue featuring the joBot Junior robot.

As said, the important point in this approach is to allow a direct connection with the previous and next levels in such a way that all knowledge gained at the Conceptual Level can now be applied to the Physical level and later on to the Programming level.

In the RoboDidactics Methodology this is ensured in the following way. Whereas the Conceptual level offers a variety of line followers, at the Physical level the students are now taught to build their own line followers. This gives them more control over what is happening and allows them to gain higher speeds with the robots, thus improving their chance of success in competitions.

The Programming Level

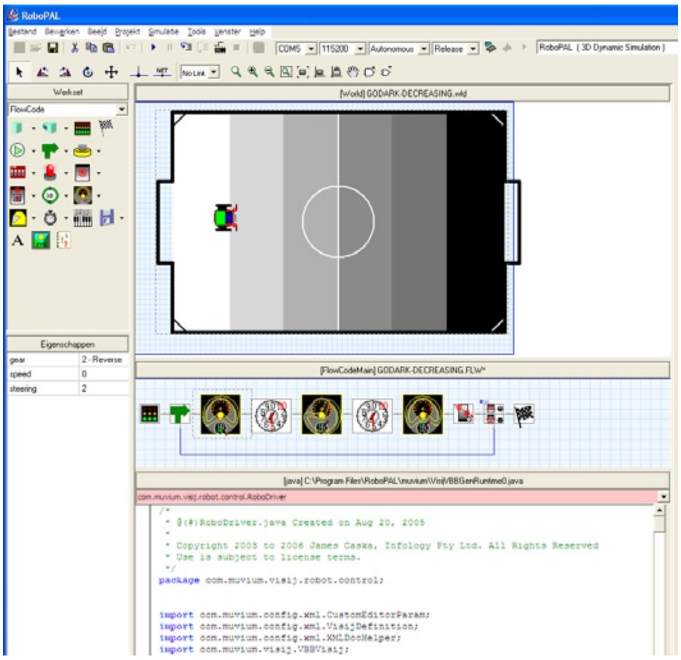


Figure 6
Showing the Java code generated by one of the driver controls

The transition from the Physical level to the Programming follows a similar principle as the transition from the Conceptual level to the Physical level. Thus, once the children have reached the stage where they create soccer playing robots, they will need to see how for instance "finding the ball" is first translated into a spinning movement, trying to locate the ball and stopping when the ball is detected. Here the transition from the Physical level to the Programming level is done by now looking at the actual Java code which is generated by the iconic language.

The iconic programs of the Physical level may be opened up and the kids can see what kind of Java program instructions are behind them.

In this manner the translation process becomes clear to them and they can start making their own programs, gradually going from the known iconic language to the more professional Java code environment.

The environment thus generates Java code which will become their entry into a different programming environment where they will use industry-standard development tools like Eclipse, but then of course they will have reached an age of over 15 years.

The code generated by the roboPAL environment however is a direct representation of the iconic program structure. So an additional level needs to be introduced to explain how such Java-coded program actually run in a standard environment.

For the older students (ages 18 and up) who have mastered the programming level, a second software environment is available. This environment contains a simulator, written entirely in Java and the robot programs are also written in Java. This simulator runs inside the open-source IDE Eclipse and is available as an open source package.

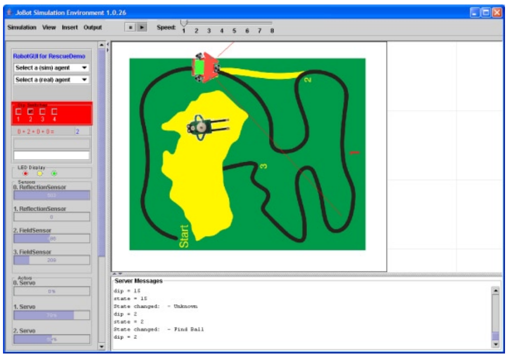


Figure 7
Java simulation for RoboRescue

In this manner the full range of software development is covered over the entire school period ranging from the age of 9 to 19 and spanning from elementary to advanced school years.

The change from the Physical level to the Programming level will be the last aspect of the RoboDidactics Methodology to be developed and, depending on resources; this is likely to happen towards the end of 2008, beginnings of 2009. Before the focus of the development work is first on the Conceptual level and then on the Physical level.

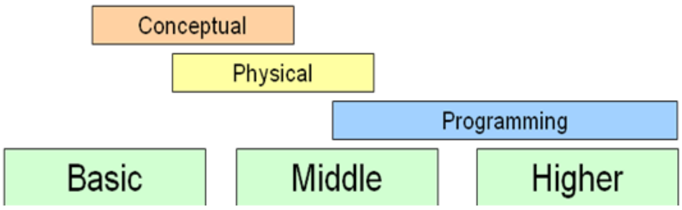
Relationship with the school levels (Basic, Middle and Higher)

One of the important goals of the didactical approach is to realize an ongoing learning path for kids to follow through their entire education, spanning ages 9 thru 19 and eventually preparing them for an academic education.

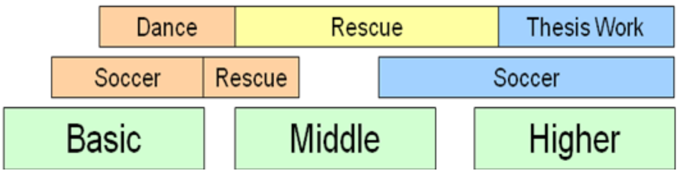
A continuous learning path consists of educational material that allows children to learn step-by-step an entire body of important concepts, with each learnt concept becoming the base for the next one, from beginning to end without having to learn a new teaching method.

Current educational systems do not allow this. For instance, children switching from Physical level to Programming level require learning a completely new programming model which interrupts their ability to absorb new knowledge. With the didactical approach developed as part of the RoboDidactics Methodology, this is no longer necessary.

Teaching material for students will be in the form of course modules. In addition, there will be material for teachers and information on how to set up workshops and small contests in schools. The following is an indication of educational content associated with the three school levels: basic, middle and higher.



- Basic schools
- Simple programs to allow younger children to perform simple assignments like rescue or soccer.
- Middle schools
- Here the concentration is on explaining the physics behind robot movements and the use of sensors in courses like Basic, Dance, Rescue, and Soccer
- Higher
- Here the concentration is more on the programming level with Java based education in a simulation environment that supports Rescue, Soccer and some experiments with advanced sensors like sound, ultrasonics and odometry. The RoboDidactics Methodology will not reach this highest level in the foreseeable future given that the focus of attention is first on the Conceptual level, then on the Physical level, and only later on the Programming level.



One continuous-learning educational scenario is shown in the figure. Education may start with Soccer and Rescue at the Conceptual level in Basic school levels. Dance gradually moves into the Physical level while soccer and rescue also migrate into the PPhysical level.

Halfway middle education the level shifts from Physical to the Programming level. Students in their last year may start the development of Java-based development, preparing for the use of the higher education facilities where powerful industrial development environments are used.

The Rescue Challenge Example

The Rescue challenge has been designed to teach a number of important concepts. Thus the field layout calls for good calibration of the light sensors. That allows connections with Physics topics and to some extent mathematical concepts.
A robot not properly calibrated will not perform well its functions.

To perform the rescue challenge with a well-calibrated robot, a line follower must be used to follow a curved line. Several curves are introduced that are difficult and call for fine-tuning of the line following parameters.

Once the robot has followed the line, it reaches an area where an object must be found and, once found, it must be moved to another location. This involves scanning an area and the development of an algorithm to visit effectively all parts of the field.

The educational material concentrates on the concepts to be mastered and it does so in a gradual manner, with kids reaching an ever deeper understanding as they progress with the material.

Building on the experience of other projects

The development of the RoboDidactics Methodology takes advantage of the experience gained in earlier projects, building on those aspects of direct relevance to the methodology. One of these earlier projects is the RoboCup Junior initiative which started in 1998 and has been adopted by over 20 countries so far. Australia is one of the biggest advocates with many schools participating. The idea started in the Scandinavian countries (started by Lego in cooperation with MIT). Almost simultaneously Japan, Canada and Australia picked up the same idea.

In 2000, the European championships were held in Amsterdam and, together with prof. Lund, the RoboCup Junior was organized with children from elementary and secondary schools. In this one-day tournament, the Interactive Lego Football (ILF) was used and the children had a short instruction session and then moved on to play soccer, using pre-built robots.

Until 2003 no other activities took place in the Netherlands, year in which Peter van Lith restarted with 5 schools, with the result that in 2004 the first competitions took place with 16 teams. Since then the number of participating teams has doubled almost every year, reaching 75 active teams, with 60 teams participating in the National Championships in 2007.

These competitive activities have encouraged the development of educational material used by most of the participating schools. Apart from the teaching material, courses for teachers were also developed by the AMSTEL institute. This course material has been used in several schools and the experience has led to a number of important educational conclusions. In particular, the pattern of use of the course material can be divided into three different categories:

- Some schools are using the material in the classroom and follow the lessons with either a group of selected students or an entire class. These schools can again be subdivided into two groups:

- Students that follow the lessons step-by-step, just copying the given examples. They gain some experience but after having finished the last lessons, they do not have a clear understanding of what has been achieved. The real learning starts after when they try to improve on their working robots.

- Students that follow the lessons by giving themselves much more freedom in interpretation, sometimes skipping lessons, sometimes only skimming the contents.

- Several schools are leaving the use of the material entirely up to their students. Most of them opt for a 'discovery' approach where they only consult the material when they get stuck. In other cases the material is only used briefly, for instance, starting at the last and most difficult lessons. The evidence shows that the schools that follow this procedure are usually among the best performers, sometimes with surprising results in elementary schools. Some schools opt to change the material or develop their own.

- Other schools do not use the material at all. Here the students prefer to develop their own ideas from scratch. Although this group learns the most and takes up the biggest challenges, they sometimes draw erroneous conclusions and when they get stuck sometimes conclude that a solution is not possible.

On one extreme in this approach are the exact followers of the material who get working robots but do not learn. Most questions that we get from these schools show that they have not gained any understanding at all. Explaining the most important concepts usually helps.

On the other extreme are the students that do their own development entirely. They learn a lot, but sometimes get on a wrong track and seem to get stuck there. When learning how to program they often develop bad habits that have to be 'unlearned' at a later stage. What is especially noticeable is that some of these students form the habit of making a conceptual design to solve a certain problem and, if this does not work, they conclude that the given hardware and software is incapable of solving the problem, without doubting their conclusions that in some cases are dead wrong.

As an example, a team that was going to get involved with Soccer and, after having made a design, concluded that they could only solve the problem using a compass. Although they have seen many operational robots working quite well without one, they insisted that this was impossible. When they got their compass and tried to use it, they claimed that this compass was not fit for use in this competition. This, in spite of the fact that it is the most used type of compass amongst participants of RoboCup Junior.

So it is clear that 'discovery' is the best way to learn new concepts, and the RoboDidactics approach should seek to facilitate the occurrence of many such 'discoveries'. The teachers must however be aware that guidance is required when an obvious dead end has been reached, to prevent the students to deviate too much from the right learning track. On the other hand, blindly following the material gives quick results but must be followed by a period of reflection to allow the students to comprehend what they have actually done.

Experience at the Conceptual Level with ILF and RoboSoccer

The previous experiences are all with course material targeted at the Physical Level. The experience with the Conceptual Level to date is very different. This usually starts at an age of about 9 years old. The children are given a 15 minute explanation of the program and then are invited to start experimenting with the program and play their first games.

Within about 30 minutes they begin to grasp the idea and notice the effect of their changes to the program. They discover how to avoid obstacles and how to follow the ball. Getting the ball into the goal area is the next achievement, although it proves much more difficult to get it into the desired goal. Scoring a goal in any of the goals is much easier and generates a lot of fun. After about an hour all students are playing soccer and it is always very difficult to try to stop them when the time runs out.

Some elementary schools participate in competitions with a program like RoboSoccer but always find that a dedicated program performs much better. This stimulates the students to start learning to develop their own game. Soccer proves too complex however to do this and in most cases switching over to a simpler game like Rescue is a better alternative. So the first experiments with user-guided behavior in the RoboDidactics Methodology will be with Rescue, providing pre-cooked line followers.

Current experience therefore shows that it is very important to let the children gain a rapid feeling of mastering the subject. This challenges them to delve deeper into robotics and to want to do some more programming. The feeling of having 'control' over the robot is very powerful and they seek to repeat it with other competitions. This is the main reason to start the development of the RoboDidactics Methodology at the Conceptual level.

Experience at the Programming Level

The experience with the Programming Level goes back further than the RoboSoccer experience. University students started working on a simulation environment that permitted the use of programmable robots in a classroom situation. The main reason for this approach was to speed up the laboratory exercises and to keep the hardware requirements down.

The students who worked on the development of the simulator were Artificial Intelligence students and Software Engineering students, who gradually developed the software over a period of more than five years. . Later on, 14-19 year old students in their last school year have used this software environment to do their practical projects. Students that select AI or ICT or Robotics receive guidance from the university and are provided with hardware and software.

At the programming level there is some course material, but most of the work is done by the students, using literature they find on the web. No educational material like for the other courses is available. There is a sizable body of lecture notes and PowerPoint slides about robotics that may be used. There also are many examples of working prototypes as part of the simulation environment; however most students prefer to develop their own.

Building on proven educational concepts

The RoboDidactics Methodology is based on a number of ideas, gradually developed by pioneers in the fields of Educational Robotics and Information Technology. As a pupil of Piaget, Seymour Papert experimented with ideas to teach children complicated concepts by placing them in a learning environment. The first results of this approach led to the well-known Logo programming environment, in which children were stimulated to develop their own micro-worlds.

Recent research indicates that the real learning takes generally place at a much deeper, unconscious level and that play and frequent repetition plays an important role. Just as frequent exercise makes motor-behavior stronger and more automatic, the same is now thought to happen on an unconscious level even before reaching the conscious level. Playing and frequent rehearsal makes these learned concepts stronger and stimulates the development of consciously experienced concepts and hypotheses forming.

Therefore the educational material of the RoboDidactics Methodology starts with the high-level Conceptual approach in a playful setting, without concentrating on the underlying backgrounds. Children learn to 'pick up' the rules without needing any explanation. Once the game has been mastered, sufficient experience has been gathered to allow discussions about understanding what is happening.
This method contains the research ideas just expressed as well as the principles listed below.

- Learning by Discovery. The ideas behind constructivism and related concepts stress that students learn to form ideas and mental models especially during the construction of real or imagined artifacts. The most important driving factor however is that children 'discover' these ideas while experimenting or experiencing.

- A Continuous Learning Path. This means that from elementary school up to college level a consistent educational approach is proposed, consisting of overlapping topics, gradually allowing students to discover and experiment with relevant IT concepts.

- The combined use of simulation environments and real hardware. This enables to speed up the development process and allows for the learning of basic concepts in a laboratory setting that can be run at any place like school, at home or during workshops and classes. Using real hardware after finishing an experiment in a simulator results in the use of less hardware in school while speeding up the execution of experiments.

- Strong connection with other disciplines like Engineering, Information Technology, Mathematics, Science and Biology. The educational material should stress the links between these fields and stimulate cross-fertilization and cross-experimentation between all fields covered.

- Using a development environment that permits experimentation, extension and, above all, gradual deepening of insights without having to switch environments with every new step.

Of course, no approach guarantees automatic success. Thus, using similar approaches in earlier projects has shown that although discovery of important concepts does happen frequently, sometimes relevant ideas and concepts are also missed.

Unguided discovery frequently leads to 're-invention of the wheel' approaches. In some cases this may lead to misconceptions that result in learning bad habits and practices.

The educational material should clearly describe the scientific backgrounds, so that students may learn that these concepts have been created or identified by other people as well and may compare the state-of-the-art with their own findings.

Another factor that seems to be very important is copying behavior. If one student has success and shows others what can be achieved this generally has a profound impact on the entire group. Conversely, if no such person is present, a state of lethargy usually sets in and not much is accomplished.

This should be used to the students' advantage with signals that stimulates positive impact. One way is to give students stimulating assignments, like no robot must be smaller than 50 cm if engaged in dancing for instance.

With rescue, the teacher can put in a time challenge and publish the times achieved by other groups and tell the students they need to beat this time.

Many links exist with other scientific fields. For instance the working of a gearbox can be explained to show the relationship between speed and power in an electric motor.

Important scientific concepts are also involved when calibrating sensors, and the same is true for the calculation of a trajectory where important math concepts come into play; while the explanation about autonomous systems has strong links with biologically inspired reactive behavior systems.

Current state of the educational material

The approach advocated by RoboDidactics Methodology is the result of ideas gradually developed over the last eight years. The teaching materials, recently developed and expanded with the concepts behind RoboDidactics, have been pioneered within robotics communities in several countries.
These efforts mainly covered the Physical level and were originally developed by RoboCup Junior Australia by Brian Thomas and Ian Maud. This material was extended by Peter van Lith and the implementation of the roboPAL environment was done by James Caska.

At the Programming level, Peter van Lith with students from the University of Amsterdam developed, over 8 years, a Java Simulator for the Java based Omni directional robot (JoBot) and associated hardware. This environment is also based on muVium, developed by James Caska and is closely connected to the roboPAL environment.
Educational material is now available for the topics listed underneath. Most of this material is available in Dutch and is being translated into English. This material covers the Conceptual level with RoboSoccer, the Physical level with Lego RCX, NXT and roboPAL and the Programming level for several muVium based robots, JoBot being the predominant one. The following modules have been developed, where D stands for Dutch and E for English versions:

- Course information and Course backgrounds (D)
- RoboSoccer course (D, E)
- Basic course (D, E)
- Dance course (D)
- Rescue course (D, E under development)
- Soccer course (D)
- Educator's guide (D)

As part of the ComLab 2 project (Leonardo Da Vinci Community Vocational Training Action Program project SL-05-B-F-PP-176008) further material has been developed, translated and tested in three different countries.

RoboDidactics adds new Conceptual Level educational material and a new Teacher's Guide. It also introduces a new robot that may be programmed in roboPAL -a third-party developed simulation environment- entirely based on the approach of the RoboDidactics Methodology.

RoboDidactics Material

The RoboDidactics manual

This document -the RoboDidactics Manual- is aimed at teachers and, thus far, it has explained the didactical approach adopted by the Robodidactics Methodology. In the following, it provides guidelines on the structure of the lessons and on how to use the material and the approach in a school setting.

Structure of the lessons

To implement the ideas described in the methodology, it is important that the structure of the educational material makes it easy to follow the lessons without having to read all textual material.

The most important objective is that the students 'discover' the relevant ideas. For this purpose, it is advisable that each lesson starts with an assignment, which is set up in such a way that the student easily discovers the intended idea. In addition only a limited amount of information should be provided, allowing the student to get started almost immediately. As was described before, students who are permitted to learn by trial and error generally gain a better understanding than students who blindly follow the examples. So the lesson and the material are to be organized to facilitate this approach. However, students who are interested in background information and more detailed information should definitely be offered this as part of the lessons.

Background material and theoretical information is therefore provided but not as required reading. It is offered as reference material to be consulted by interested students or students who get stuck and need help.

Every lesson therefore should containing the following information:

1. What you will learn, containing a brief description of the main points covered in the lesson.
2. The assignment, describing what needs to be achieved or developed. The information should be brief and sufficient to get started and to understand when

the assignment is successfully achieved.
3. Background information, providing helpful information to be consulted mostly when a student gets stuck. It also contains more background and reference material to deepen the understanding of topics covered in the lesson.
4. Technical aspects, giving the most important technical information about the assignment. This information is not essential to execute the assignment but it is useful for students interested in more detail about what happens on a deeper level.
5. Theoretical background, providing theoretical information to students who want to understand why certain things are handled in a specific manner in the example programs.
6. Test and questions, providing a test for a certain result or a list of questions to check if a student has achieved the goal and understands the concepts in the lesson.

Structure of the Teacher's Guide

An important part of the methodology is the teacher's role in the discovery process. If students are allowed to explore the exercises without consulting more detailed information in an attempt to discover important ideas by themselves, the teacher must make sure that this discovery process does actually happen.

In some cases students will not discover the important ideas and keep trying or simply give up or proceed to the next lesson. It is very important that the teacher notices when this happens and proceeds to offer help and hints to ensure that the student learns as intended.
As indicated earlier on, sometimes students may think they understand and get some assignment to work, but based on a misconception. When this is not detected early enough, students may teach themselves erroneous concepts that later on have to be 'unlearned'. Although this may lead to even deeper understanding, this process should not be allowed to continue too long, since it may lead to frustration.

So the teacher's guide prepared as part of the Robodidactics Methodology will offer information about what actually needs to be 'discovered' and additional information about things that are likely to go wrong. At least the following major aspects are contained in the teacher's guide:

1. Ideas to be discovered. It is very important to define clearly what concepts are supposed to be discovered. Students that do not read the additional material may need some help along the way. The teacher should make sure that when a

student gets stuck or seems to be taking a wrong direction, this is detected early and then offer hints.
2. Common problems and pitfalls. Experiences from earlier tests and suggestions from the course designers should help teachers to prevent problems or offer help in guiding students who get stuck.
3. Background information. Some more detailed information is provided to the teacher to help students in gaining a better understanding. This background information also contained tips or additional information not available in the student material.
4. Additional exercises. For students who are bright and the assignment seems too simple, suggestions for additional exercises are included.

Organizational information

The items mentioned so far have to do mostly with the course content. Also useful to teachers is information that provides ideas or examples of how the flow of activities for specific lessons and classes can be organized. Therefore the following information will also be given as part of the Teacher's guide:

1. Learning Objective

-Knowledge
-Life Skills
-ICT skills

2. Material

3. *Time*

4. Class organization

5. Procedure

6. Exercises or Homework

7. Suggestions for Further Activities

Not all items need to be present and care must be taken, not to create a 'cookbook' type of lesson environment. The discovery process should always be central to the setup of the lessons. Classes should not be given as a tutorial but must be seen more as a workshop in which a single assignment is being executed by the students.

Hardware and Software

Hardware and Software plan

The RoboDidactics Methodology, as its name implies, is a didactic methodology that makes use of the powerful educational potential of robots.

This means that, ideally, it should not be attached to any single robot platform, i.e., it should be platform independent.

In practice, this is not easy to implement and the first implementation of the methodology is based on a modified version of the RoboDesigner RDS-X01 robotic platform produced by RoboTech Japan and distributed by RoboTech Italy.

The hardware and software modifications to the RoboDesigner RDS-X01 platform make it consistent with the demands of the proposed RoboDidactics Methodology, enabling in particular the use of a development environment that supports the three Conceptual, Physical and Programming levels and includes a simulation facility that allows for greater richness and flexibility in organizing the learning process.

In the longer term other platforms should be added to the RoboDidactics approach. It is also possible to include other types of software, as long as the main ideas behind the RoboDidactics Methodology are fulfilled realized.

We now describe the current hardware and software platforms that may be considered as part of the methodology.

Hardware

The hardware used in the course material aims to be multi-platform. Initially the RoboDesigner RDS-X01 is the platform of choice. We intend to support the Lego MindStorms RCX and the new NXT system perhaps at a later stage.

We will support the JoBot and JoBot Jr. as well as the JelloBot and other robots, running MicroChip processors.

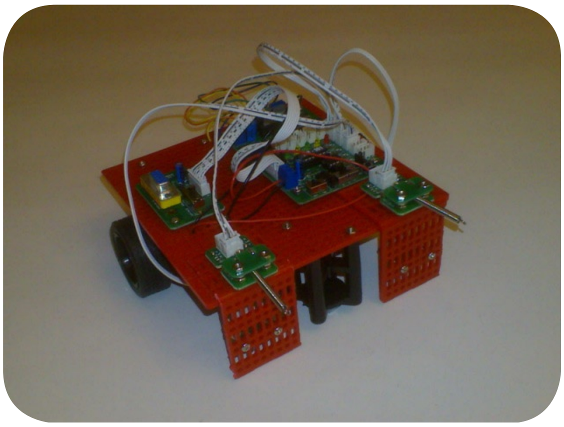
A distinction must be made however regarding the levels in schools that the different platforms are aimed at. The following is the current situation:

- Elementary Level
- RoboDesigner RDS-X01 and JelloBot platforms
- JoBot Jr
- Lego RCX and NXT (maybe later)
- Middle Levels
- RoboDesigner RDS-X01 and JelloBot platforms
- JoBot Jr and JoBot
- Lego RCX and NXT (not yet with roboPAL)
- Higher levels will want to build their own hardware and experiment with a variety of sensors:
- JoBot for higher schools
- Advanced sensors
- Sound
- Ultrasonic
- Mouse odometry
- Intelligent Java Camera

For the roboPAL software developed in accordance with the approach of the RoboDidactics Methodology, any robot that uses the industry-standard PIC processor that can run the muVium environment may be used. Currently, roboPAL supports PIC processors only. The various robots shown here are either equipped with a PIC processor or they can be provided with a plug-in processor board that enables them to work with roboPAL. For the Lego NXT and possibly the RCX, we are considering a port to the ARM7 processor and the Hitachi H8.

Also available is instruction material that describes how to build a robot, suitable for the courses, using Lego MindStorms and the other mentioned platforms. Those schools that will not be using roboPAL may use the instruction material, created for RoboLab en NXT-G for both Lego platforms. Please note that these approaches do not support simulators and make implementation of the RoboDidactics approach much more difficult. Especially problematic will be the transition between levels.

RoboDesigner RDS-X01 robot



This is the platform, designed by RoboTech Japan and distributed by RoboTech Italy, one of the RoboDidactics partners.
The standard version runs TiColla.
With an optional expansion board this robot may also be used with roboPAL and the Java Simulator.
This robot has been included in the roboPAL environment and tests have been performed, using a prototype board.

Software

Using Simulators

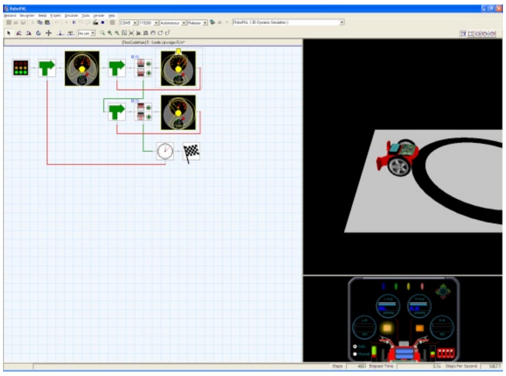
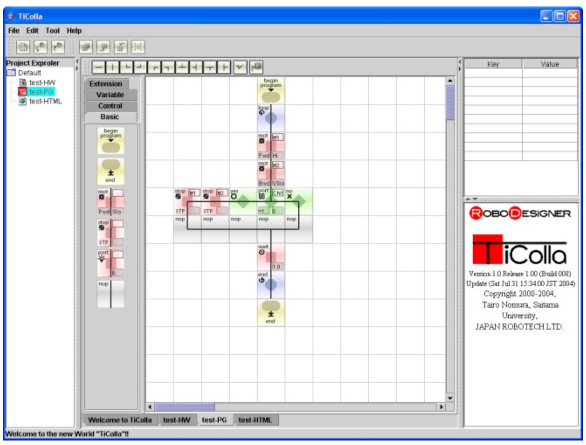


Figure 1
roboPAL simulator following a line

When using a simulator, children developing their software may directly see the results of their program on their PC. This saves a considerable amount of time during development and decreases the need for hardware in the classroom. Although there is no substitute for working with a real robot, working with the hardware is postponed until the children have first demonstrated that their program works as intended. When they then load their program onto the real robot, they will find that the real world has some interesting surprises for them. This is an excellent stepping stone to explain Physics and show the results of a noisy environment, sensors that need to be calibrated, motors that do not behave exactly the same and introduce differences in speed with things like gearboxes and transmissions.

Visual Environments

The development environments for robotics that will be used with the RoboDidactics Methodology are TiColla and RoboPAL.



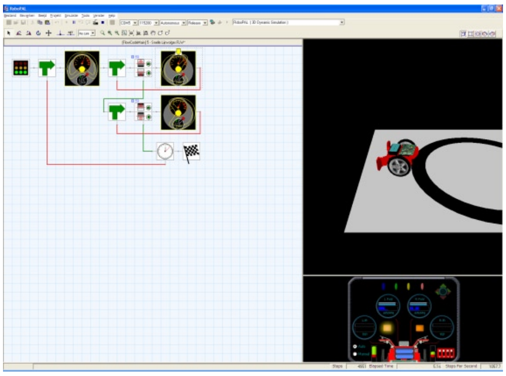
TiColla

TiColla is the standard software environment that comes with the RoboDesigner RDS-X01 robot but does not contain a simulator. In terms of the three levels of the RoboDidactics Methodology, its operation is on the Physical level but its building blocks are more towards the Programming level and the code generated cannot be inspected.

If TiColla is to be used as part of RoboDidactics, it is important to be aware that each student will need its own robot.

A simulation environment for this software seems to be available but has not been marketed for the RDS-X01 robot.

RoboPAL



RoboPAL is a development environment entirely based on the approach for the RoboDidactics Methodology. It contains a 3D simulator and currently supports the Conceptual and Physical levels.
The Java code generated is executed through in intermediate interpretation level and is not yet on the same logic level as that of the Java Simulator.
The Icons used are either of a high conceptual level or at the physical level. Students may also create their own new functions.

RoboDidactics based software

The simulator allows students to develop software for the robot and to test it in the simulator and then downloading it onto the robot platform.

Currently this is only available for Java based PIC processors, running the muVium environment. This Java environment is developed for use at universities. The environment is meant both as a toolbox for robot development and as an interesting environment to experiment with simulations and emulation of microcontrollers.

The environment consists of the icon-based roboPAL programming language (Play And Learn) that is developed by muVium in cooperation with RoboCup Junior. The initial educational material, courses and the Java based software are all developed by the AMSTEL Institute and the University of Amsterdam and form the starting point of RoboDidactics.

The roboPAL environment is a commercial product under development, specifically designed to allow students to work with small robots.

The lowest level will be a further extension to the roboPAL software, specifically aimed at elementary school levels and developed along the lines indicated before. This will be aimed at elementary schools and taken up as a later development, after the iconic and Java layers are both finished.

There will be experimental software in the form of two development environments.

- Experimentation version RoboPAL is a visually based programming environment that is icon based and addresses the following levels.

- Basic and middle levels

- The Java Simulator is an experimentation environment that allows students that have mastered the iconic environment to use the code generated by this environment to be loaded and used in a professional environment in Java, thus gaining real programming experience. This experience is driven by the programs that were developed with RoboPAL, thus allowing them to reuse the knowledge gained in the more elementary levels.

- Advanced Java software

- Hardware simulation

It is our intention to keep the software platform independent. Both the RoboPAL and the Simulation environment will run under Linux and Windows.

Current status

The Java Simulator has been in operation for several years and is available as open source software. The support for the hardware emulation layer is done by muVium and requires the muVium Virtual BreadBoard software library. This library is available free of charge and is an integral part of the support software for the muVium Java based chips. Currently most PIC chips are supported with new versions for ARM7 under development.

The roboPAL environment is a commercial package, developed by muVium and is sold as a separate product either on a single user basis or as a school license. It also comes bundled with either a plug-in processor or complete robots like the RoboDesigner RDS-X01.

The Evolution and Dissemination of the RoboDidactics Methodology

The RoboDidactics Methodology and associated hardware/software robotic environments are not static products. As the previous sections show, they are products in continuous evolution both in their educational and technological dimensions. Indeed, the first versions of these products should be conceived as the first steps in an evolutionary process leading to their full development. In this process, the developers of the RoboDidactics Methodology and associated hardware and software establish a dialogue with school users and experts, so as to test their value and obtain feedback that helps improve their content and features.

Initially, the first versions of the RoboDidactics Methodology and associated hardware and software will be tested by a selected number of schools in several European countries: Germany Italy, Poland, and Slovenia. The RoboDidactics Methodology will also be given to an International Panel of Experts (IPE) for comments.

The tests will start with the Conceptual level and, depending on the level of robotic experience of the schools; they can reach and even start at the Physical level. Some schools might go further up in complexity and venture into the Programming level. The results of the tests and comments by the panel of experts should lead to a new improved version of the methodology and hardware/software environments.

This new version should reach a standard of quality sufficient to expand its dissemination beyond a set of pilot schools into the wider school domain. As this happens, more users will be able to provide feedback, thus leading to further improvements in the content and features of the methodology and hardware/software environments.

The process of wider dissemination and interaction with users can also involve relevant players such as the international robotics communities, associated initiatives like ComLab, publication in international magazines and, of course, the organization of some international competitions. Indeed, the planned RomeCup competition during 2008 is a good example, but also are international projects like the Comenius international RoboCup program, the RoboLudens, the German Open competitions planned for 2008, and the 2008 World Championships in China.

Basic research questions on the didactic use of robots in schools

The process of evolution of the RoboDidactics Methodology and hardware and software environments in interaction with users provides also the opportunity to find answers to some basic research questions on the didactic use of robots in schools.

The methodology seeks to show children how interesting technology can be and the main idea behind the RoboDidactics approach is to allow students to gradually learn the principles of designing and building complex working objects, like robots.

A major purpose of this approach is to stimulate children to select scientific and technological education and, more generally, to stimulate the didactic use of robots in all subjects across the entire school curriculum for 9 year old children and older. This raises some important research questions, for instance:

- What motivates children to get involved with technology or to decide not to get involved with it? Can a more interesting and playful approach improve this situation and to what extent? Is the RoboDidactics approach addressing the right issues and is this approach working and to what extent?

- How long does it take children to gain a basic understanding to work with the first basic lessons? Is this different for various age groups. Is it gender dependent? Once learned, how quickly do they get bored and want to move on to the next level?

- A major drawback of learning by exploration is that students may form wrong habits or enter a lengthy path that deviates from the goals. How do we detect that this is happening and how do we solve this situation? Is a guide with set goals a good approach and how are we then going to enforce this?

- How do we know what level of understanding has been gained? What metrics are appropriate to measure and compare this understanding?.

The process of interaction with users during the pilots and wider dissemination will open up the opportunity to explore answers to these and other questions of relevance to the challenge of using robots for didactic purposes at school.

Bibliography

Robotics as an Educational Tool
Orazio Miglino, Henrik Hautop Lund, Maurizio Cardaci

Evolutionary Robotics; A Children's Game.
Henrik Hautop Lund, Orazio Miglino, Luigi Pagliarini, Aude Billard, Auke Ijspeert

AI in Children's Play with LEGO Robots
Henrik Hautop Lund

Programming robots is fun: RoboCup Jr. 2000, Ben Kröse, Rob van den Bogaard, Niels Hietbrink in Proceedings of the twelfth BNAIC conference

What is the Best Environment-Language for Teaching Robotics Using Lego MindStorms? Anthony Hirst, Jeffrey Johnson, Marian Petre, Blaine A. Price, Mike Richards

MinStorms, Children, Computers and Powerful Ideas.
Seymour Papert

Vision for Education: The Caperton-Papert Platform. Seymour Papert and Gaston Caperton.