Lessons learnt with LEGO Mindstorms: from beginner to teaching robotics

PaedDr. Martina Kabátová
Dept. of Informatics Education
FMFI Comenius University
Bratislava, Slovakia
kabatova@fmph.uniba.sk

PaedDr. Janka Pekárová
Dept. of Informatics Education
FMFI Comenius University
Bratislava, Slovakia
pekarova@fmph.uniba.sk

Abstract **In this paper we describe different lessons of educational robotics at different level of robotics experience. First one focuses on developing basic skills needed to successfully control the robotic model. The other one uses advanced programming skills to provide communication between two programmable bricks. We also describe organization of the project in brief. LEGO Mindstorms kit was used as a learning platform for all activities. Addressing differences between beginners and experienced students, we list several tips and recommendations how to execute quality robotic lessons dealing with the communication and controlling the robot.**

Keywords: robotics education, LEGO NXT, programming, guidance, project

I. Introduction

Robotics presents an attractive introduction to object-oriented programming or higher programming languages (see [2], [5], [10]), but it can be also used in the lower levels of education. Dealing with real robots has high motivational effect – students visualize their robot as a toy [11] which behavior can be set according to scenario where it is used. The experiences with robots are tangible although their design requires much abstract thinking. Finally, they enable rich varieties of interdisciplinary projects. Therefore we consider robotics to be powerful tool for developing thinking. We pay special attention to the preparation of pre-service teachers who can enrich their teaching repertoire by the robots' use. We have been realizing the robotic seminar for them for several years. We try to explore the effective way for constructing students' comprehension of robotics. We often find our methodology similar to different courses worldwide.

II. How robotics is taught

Carnegie Mellon Robotics Academy [3] offers a special robotic course for educators. Here they learn more about:

- MOTION and CALCULUS (What is a robot?; Mindstorms hardware; Movement and rotations; Size, distance and movement; Abstract bridges; Challenge: go as close as possible)
- ROBOTIC SENSORS (Measure – Plan – Execute strategies overview; Touch, ultrasonic, light and sound sensor; View Menu; 'Wait for' block; Limits of the measurements; Tasks with sensors)
- DECISION MAKING (Repetition; Obstacle detection; Cycle, condition and conditional cycle; Line following; Setting the ride through obstacles; Iterative solution of the problems; Challenge: ride through obstacles)
- EDUCATIONAL ROBOTICS (Possibilities; Challenges; Robotics in your school)

Different approach to educational robotics can be found in TERECOP project [1]. Teachers learn how to teach robotics in a constructionist way. Besides the basics of programming the kits, participants also get informed about constructionist learning philosophy and project-oriented classes. They analyze and assess the robotic model, suggest own assignment for their students and think over the organization of robotics projects. In the core lessons of the project they learn to measure with sensors and control the motors of the robot using the basic program blocks. Moreover, they learn how **to check if robot works** in the way prescribed in assignment and how **to modify the program** in order to fit assignment needs.

MIT Lifelong kindergarten applies four principles into their leisure time robotics workshops for children and families [13]: (a) Focus on **Themes** (not just Challenges); (b) Combine **Art and Engineering**; (c) Encourage **Storytelling** and (d) Organize **Exhibitions** (rather than Competitions). Authors aim to make robotics attractive to the widest range of different people.

[9] puts emphasize on creating **cooperative learning environment** where **small groups of students** maximize each other's learning while working on robotics projects. In the proposed curriculum students at first work with ready model, in order to understand possibilities and limitation of robotic kit. They use programs prepared in advance, learn to modify them and after that try to design a functional robotic model on their own. Similar approach is presented in [14]. **Challenges** are taken in the end of each lesson in order to **ensure understanding** to core concepts of the lessons.

Youth Engaged in Technology programme also combines team building activities with demonstration of programming instruction to the robot [6]. This course further contains necessary math to calculate gear ratios. Several exercises are focused on building and mechanical components of robots. Open-ended challenge completes the course syllabus.

[8] has decided to give his robotic course the form of open lab where students can spend **much time** on solving various

robotic related tasks which encourages students to be more creative in their design and robot implementation. The assignments are close-ended and clearly defined, although the author recognizes the potential of **open-ended projects in combination with contests**.

III. Seminar Robotic kits in education in detail

We have been applying constructionist ideas and principles ([12]; [13]) in our seminar practice:

- **learning by doing**, **hands on activities** through own experiences – students build and program robotic model and test its functionality;
- **authentic success in finding problems and their solutions** – students decide how the robot will work and choose way how to achieve this, they select the topic for their project and explore programming possibilities;
- **hard fun** and **playful learning** – robotic kits are in fact toys, but solving some tasks with them can be quite complicated, the atmosphere on the seminar is relaxed and we try to help students learn in an entertaining way;
- **creative learning** by designing and inventing is included in the creating the robotic model;
- **combination of digital technologies** as a building material together **with art materials** – students can decorate their robots, make a costume for their models or produce some coulisses;
- **enough time** – the syllabus of the seminar is quite flexible, we can spend more time on activities that last „too long";
- **freedom to make mistakes** and learn from them – students get space for their own, independent solutions in which they go wrong sometimes, we try to reveal the core of problem in common dialogue and help them to solve it;
- **teamwork**, **collaboration**, distribution of roles within the group, common work on problem solving – students learn how to organize teamwork, some assignments cannot be completed individually (e.g. to prepare the robot for the contest);
- **learning together** – it is not possible for us – teachers – to be prepared for the whole range of troubles that can happen, we also solve novel tasks and learn new things together with our students.

The syllabus of the seminar keeps balance between closed tasks having the only solution and open-ended projects:

TABLE I. The syllabus of the seminar

Week	Topic of the seminar	Tasks, solution, teamwork
1	**Programming without computer?**	Simple closed tasks, one solution, small group
	Creating simple program through NXT brick interface	
2	**Introduction to Mindstorms programming**	Simple closed tasks, one solution, small group
	Move, Display and Wait for block, Cycle, Condition	
3 - 4	**More on Mindstorms programming**	Simple closed tasks, one solution, small group
	Procedures and variables; Parallel processes	
5	**Experiments with sensors**	Single task mixed from small group work and common activity with all students
	Read/Write to file, variables	
6	**Communication between robots**	Advanced tasks for small groups collaboration
	Send/Receive message	
7 – 9	**Our Project 1. Preparation for the robotic contest as an alternative**	Open-ended project, many possible solutions, small group
10 – 12	**Our Project 2. Exhibition and presentation of the models**	Open-ended project, many possible solutions, small group builds single model for the common topic

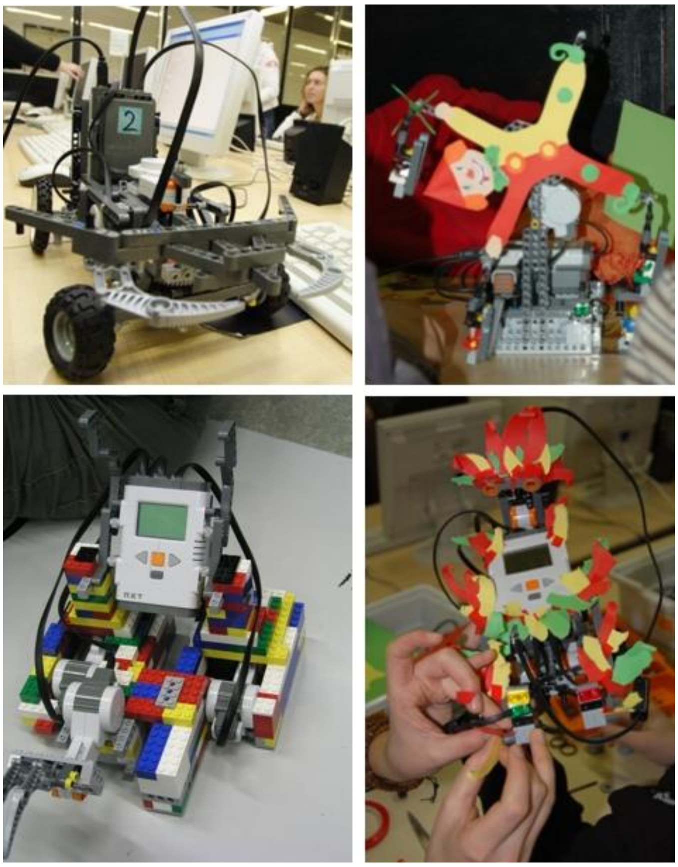

Figure 1. Our Project – examples of outputs

We have touched the way of organizing open–ended projects in [7]. Briefly, during previous terms students:

- built and programmed robotic elevator controlled by touch sensors – this project is described in section V,

- suggested and realized robotic models for Space and Playground topics,
- designed and completed moving pieces of Spooky castle, Amusement park and Intelligent house.

We encourage students to do **pedagogical reflection** of their own robotic design. The part of robotic assignment is also a teacher checklist where they advice fictional teacher how they can realize similar robotic project – how much time is needed, special needs of hardware, previous programming skills, common problems and their solution. Although documentation of the work in this way isn't very popular among students, they are able to produce high quality description of their robotic device.

In the following sections we describe three lessons: (a) initial lesson in Week 2 where students work in the programming environment for the first time; (b) communicating between robots in Week 6 and (c) design of the project-based activity The Lift. We want to show the continual progress from simple close-ended tasks that help the beginner to develop basic programmer skills to change the robot's behavior to more open-ended tasks that require more creativity and open space for more qualitatively different solutions. We think it is necessary to give participants the possibility to **design, build and program the robotic model on their own**. This brings additional effect: the pre-service teachers experience some common problems when designing the robots and can more effectively give advice to their own students in future.

IV. LESSON ONE: CONTROLING THE ROBOT

Students use the standard model of the robot (as presented in Robot Educator section of Mindstorms programming environment). We provide them by the set of clearly defined, simple tasks so they get to know the possibilities of the iconic programming environment used to program the robot. During previous terms we noticed several problems and misconceptions common to various groups of students. Let's have a closer look at some of them.

Robo – archeologist Assignment: Choose one of the letters written on the stone.

- Teach your robot to move in the outline of the letter.
- Teach it to move in the outline of crocodile teeth:

In the very first assignment which should be solved in Mindstorms programming environment students express their high expectations of robot's possibilities. Many ask whether the robot should follow the black outline of the letter and some students even don't ask and try to program it. The main idea of the assignment is in fact much simpler: they should learn to use Move block and use the sequence of this type of block to produce the track of the robot in fixed dimensions (we advise the students the blocks needed in solution on the edge of the assignment).

In trying to create the crocodile teeth track students find out that the angle used in Move block isn't the same angle in which the robot turns. The angle stands for the rotations of the motor – 180^{o} means half of one rotation. Solution of the task is then often based on many experiments with the settings of Move block in order to create desired outcome of the task.

Our students are experienced programmers when beginning to attend the seminar. At this point they suddenly find out that programming real robots differs from programming virtual ones (e.g. a turtle in Logo programming language) – they have to consider physical aspects of the model as well as properties of the environment. When they program the robot that should move after whistling, some of them face the problem that robot will start moving immediately after the program launches (as the noise in room is often high). They discover the need to calibrate the sensors.

We always encounter students who try to program a continuous motion of the robot (no matter how outer conditions are). They soon question why the program containing only one Move block set to Unlimited steps doesn't work as they expected. This is the other difference between programming real and virtual creatures that needs to be explained explicitly and perhaps demonstrated in more guided instructional way.

Students learn to set robot's behavior depending on outer conditions – values measured by sensors in several real-life tasks, for instance:

Robo-racer Assignment: Your robot is waiting for the start-the-race signal. When hearing it, it will move forward.

- Teach it to take its run – to increase its speed.
- It will quickly go forward. It will stop when it finds the (black) line marking the finish of the race circuit.
- After achieving the finish line, the robot will turn all around because of the joy from victory.
- Each racer will smile after turning around – there will be a smiling face on its display.

The lesson finishes by the task requiring a partial disassembling of the robot. We were inspired by [4] in its assignment – the challenge was to increase the robot's speed so that it will move faster than the programming environment allows in default:

The Thief Assignment: Try to achieve as fast motion of the robot as possible. Find at least two different solutions.

Hint: You may need to modify the construction of your robot slightly. **Notice how the power of the motor transfers to the wheels.**

Besides experimenting with gears and program settings students should find the answers to these questions:

- What will happen if you enlarge the cog wheel connected to the motor and use smaller cog wheel connected to the wheel?
- What will happen in opposite case?
- How much load can the robot push if you use various types of cog wheels?

We motivated students by short movies by [4] and discussed the answers to the questions with them. They had chance to create hypothesis and test it immediately in real conditions. This task was appreciated also by two girls which showed no interest in mechanical issues of robotics and they were proud of themselves that they found the arguments supporting their opinions. Finding solution to this task requires lot of time (because of the need to assembling new driving mechanism for the robot) and the task should be included as the last piece of the lesson because of the need to re-build the robot construction. It inspires future teachers to think in an interdisciplinary way – they think over the math hidden in mechanics and construction of the robotic models.

V. LESSON TWO: ROBOTS COMMUNICATE

As soon as the students manage the basics of programming language, one lesson is dedicated to communication between two (or more) NXT bricks. The bricks have Bluetooth and can be programmed to send and receive various messages. Unfortunately the process of connecting two bricks has some flaws. We found out that creating a connection should be guided as much as possible as the process is nothing near intuitive and detailed guidance can significantly reduce mistakes and errors that have nothing to do with controlling robots.

Once the bricks are successfully connected the teams shall write programs to send and receive messages. In first simple introductory assignment one brick sends a word and the other brick displays the word on its display. We used to give another introductory assignment but we realized that it is not easy if it should be done correctly (the task is to send a Morse signal and the other brick should beep the message out).

The final task is to program a car and its remote controller.

Remote-control car Assignment:

Team A: Create a program that will send control messages to the car. Use NXT buttons on your brick as a remote controller.

Team B: Create a program to receive messages and move the car according to them.

Both teams: Negotiate the message content and how to interpret them. Think about car controlling - what is the desired behavior that will be suitable for a race?

The programs contains loop, if-statement and reaction to NXT buttons or blocks to move the motors. There are several good solutions either with variables or without using them. We encourage the students to find their own solution and we only correct their errors once they ask us. The most common solution for receiving program contains two or more nested if-statements reacting to various messages from the other NXT brick.

Students should also design the behavior of the car, once it is successfully controlled. After few take outs they realize it is important to make the car respond in certain way to be able to navigate it around racing circuit. This task is very closely connected with actual environment and usability of their model.

This activity culminates in a competition. We measure the time needed to finish the racing circuit and give penalty seconds for bumping into circuit borders.

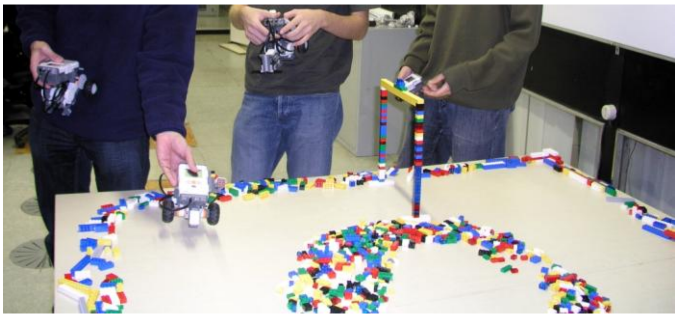

Figure 2. Testing the communication between robots on the racing circuit

Realizing short contest with remote-control cars is our reaction to recent feedback from students who have missed challenges and opportunities for the competitions besides the exhibitions (organizing exhibitions is strongly encouraged in [13]). Still, some groups of students are not interested in the possibility to compare with the others at all. We assume that teachers should provide opportunities for exhibitions as well as for the contests in robotics classes, in order to suit the learning style of most students.

VI. OUR PROJECT: A LIFT AS FAST AS POSSIBLE

In the winter term of 2007 we introduced the open ended project-like activity named The Lift Model. At the end of this term the activity was rated as most popular among the students.

The assignment for the students:

Design the structure and devices of functional lift,

- build the frame tower, the cabin and some mechanism that will pull the cabin,
- program the sensors to act as lift controls (e.g. the touch sensor can be a button).

Your task is to build a functional model of a lift. You will use LEGO NXT kit, sensors and programming language. You decide how will the frame look like and how the lift will be controlled. Your model should be able to go up and down as the user needs. Note that

- the frame should be high enough,
- the frame should be steady and should not lean to the sides,
- the cabin should not tilt or spin, it should have the

most stable position,

- if the cabin is too heavy, use right gears to lift it although at lesser speed.
- Challenge: try to build the fastest lift possible, is your lift faster than models built by other groups?

The material for students included also some reference photographs of various lifts, pictures of lift that was built by us and list of LEGO bricks we have used in our model. We couldn't bring the actual LEGO model since all kits were used by the students and there was no spare kit for the model. According to [13] for students it is very inspirational to see sample models, they have more ideas and easier identify the problems they are about to solve. As there is often a problem with material and many teachers don't have spare kits we suggest using photographs and videos instead.

Originally we expected the project to take 3 lessons (each lasts 90 minutes), but in the process we realized 4 lessons are needed to finish all the work and present the models.

Three out of four teams finished their models so they were fully functional. Two of those models used double motors at top of frame to pull the cabin. The third solution was a single motor placed at the base of tower. As we anticipated the highest tower was also the most unstable. It is noteworthy that all teams placed some LEGO figures or other decorations on their models (one of the girls even brought her own brick from home). This indicates that the students should have the opportunity to use additional decorative materials to enhance their finished models.

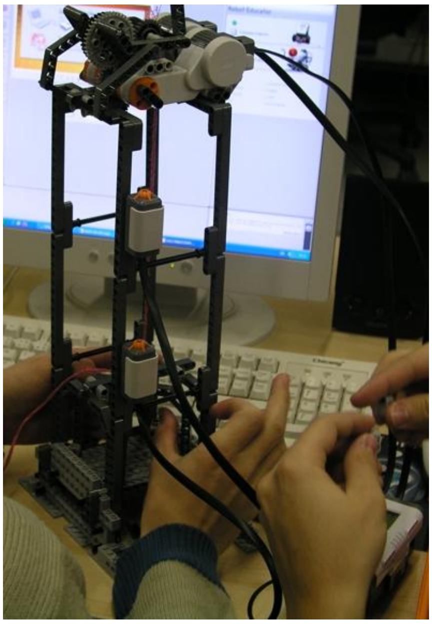

Figure 3. A lift controlled by two touch sensors. The cabin is lifted by two motors placed on the top of frame.

This activity was open-ended but it's nature didn't leave much space for students own inventions and creativity. They also voiced this opinion in the final interview. In next courses we introduced more theme based and even more open-ended activities, though we think there are students that need more guidance and instructions during deciding what are they going to design and build. We suggest that the creative robotics principle "focus on theme" [13] is good way to give students the basic layout of what are they going to do, but the teacher should focus on the less skilled and less creative teams and help them to find more tangible model description - this can be done via guided discussion. We strongly agree that the process of finding the problem is equally or even more important than actually solving the problem.

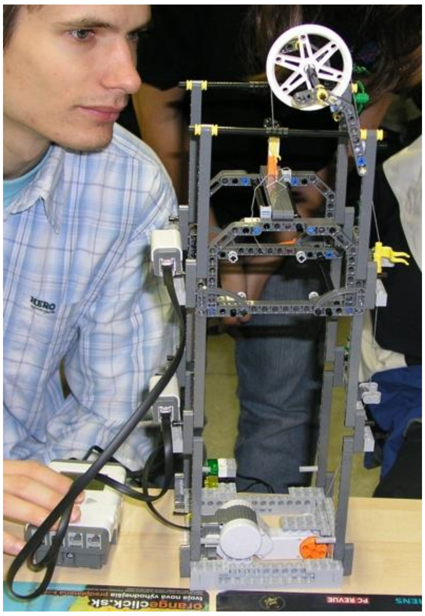

Figure 4. A lift using light signals

At the end of this activity we let the teams to present their models. Unfortunately we were not able to evaluate the challenge for the fastest lift. This should have been done and according to the final interviews students also expect some competition like activities. We think that it's reasonable to include both types of evaluation - competition and exhibition, as they appeal to different students. In case of our seminar the need for competitions is given by specific target group (most of course participants are computer science, mathematics and management students and males).

VII. Discussion

We have described design of our robotics course for pre-service teachers and computer science students. After several iterations of whole course and introducing various types of assignments we propose that at the beginning students should solve smaller close-ended tasks with basic robotic model they do not build. This way they can learn about programming the robotic model and experience basic principles of event driven robot controlling. In later lessons it's reasonable to give students more space to create their own model and follow more constructionist lesson design with plenty of time for experiments.

The three different lessons we detailed suggest that different amount of guidance and instruction is appropriate for various activities.

We have also applied some principles of creative robotics [13] and we argue their relevance for our specific target group.

In conclusion there are some relevant issues that should be considered when teaching robotics computer science students and pre-service teachers, in our course design we try to address them.

REFERENCES

[1] Alimisis, D., Moro, M., Arlegui, J., Frangou, S. and Papanikolaou, K. , "Robotics & Constructivism in Education: the TERECoP project". In Kalas, I. (ed.) Proceedings of EuroLogo 2007. Bratislava, 2007.

[2] Barnes, D., "Teaching Introductory Java through LEGO MINDSTORMS Models." In Proceedings of the 33rd SIGCSE technical symposium on Computer science education, 2002, pp. 147 - 151

[3] Carnegie Mellon University, Professional Development Robotics Academy Classes, available 31.5.2010 at http://www.education.rec.ri.cmu.edu/content/educators/professional_dev/index.htm

[4] Carnegie Mellon Robotics Academy : Introduction to Robotics; Using the NXT. Electronic learning material, 2006

[5] Flowers, T.R., Gossett, K. A., "TEACHING PROBLEM SOLVING, COMPUTING, AND INFORMATION TECHNOLOGY WITH ROBOTS." Journal of Computing Sciences in Colleges archive Volume 17 , Issue 6 , pp. 45 – 55, May 2002

[6] Hoy, P., Webster, P., DiMarzio, C., Batterson, D.F. & Perkins (2006): "ROBOTICS CURRICULUM: USING LEGO® MINDSTORMS® ROBOTICS KITS." Pennsylvania State University: Youth Engaged in Technology (YET) Programme.

[7] Kabátová, M., Pekárová, J., "Learning how to teach robotics." Paper accepted for Constructionism 2010 conference. In press.

[8] Kumar, A. N. "Three Years of Using Robots in an Artificial Intelligence Course – Lessons Learned.". In ACM Journal on Educational Resources in Computing, Vol. 4, No. 3, Article 1, September 2004.

[9] Lau K. W., Tan H. K., Erwin B. T., Petrovic P. (1999) "Creative Learning in School with LEGO Programmable Robotics Products." In Proceedings to Frontiers in Education'99, IEEE CS Press, pp. 12D4/26 - 12D4/31 vol.2.

[10] Lawhead, P.B., Bland C.G., Barnes, D.J., Duncan, M.E., Goldweber, M., Hollingsworth, R.G., Schep, M., "A Road Map for Teaching Introductory Programming Using LEGO© Mindstorms Robots". In ACM SIGCSE Bulletin Volume 35, Issue 2, pp. 191 – 201, June 2003.

[11] Mauch, E. "Using Technological Innovation to Improve the Problem-Solving Skills of Middle School Students Educators' Experiences with the LEGO Mindstorms Robotic Invention System." In The Clearing House, Vol. 74, No. 4, pp. 211-213, Mar. - Apr., 2001

[12] Papert, S., "The Eight Big Ideas of the Constructionist Learning Laboratory". Unpublished internal document.

[13] Rusk, N., Resnick, M., Berg, R., & Pezalla-Granlund, M. "New Pathways into Robotics: Strategies for Broadening Participation." In Journal of Science Education and Technology, vol. 17, no. 1, pp. 59-69, 2008.

[14] Sklar, E., Eguchi, A., "Learning while Teaching Robotics." In The AAAI Symposium Series: Accessible Hands-on Artificial Intelligence and Robotics Education, 2004.