The LEGO NXT Robot-based e-Learning Environment to Teach Computer Science Topics

R. Burbaite, V. Stuikys, R. Marcinkevicius

Software Engineering Department, Kaunas University of Technology,
Studentu St. 50, LT-51368 Kaunas, Lithuania
renata.burbaite@gmail.com

Abstract—It is difficult to motivate learners to learn abstract Computer Science topics (e.g., data structures, algorithms and programming) with the adequate level of engagement. We present the process of constructing a LEGO robot, called the DRAWBOT (drawing robot), which enables to create the e-learning environment to demonstrate visually the solution of graph-based Computer Science tasks through teaching programming. Our research has confirmed the importance of using robot-based environments for teaching that was known so far in the literature on e-learning. We have extended the known approaches: a) by providing technical characteristics for the process to create the e-learning environment for the real setting; b) by smoothly integrating different phases of the process and considering it into entirety to support the constructivist learning model.

Index Terms—Computer science e-learning, LEGO NXT robot, learning task visualization, robot control program.

I. INTRODUCTION

Computer Science (CS) deals with abstract topics (e.g., data structures, algorithms and programming, etc.). It is difficult to motivate learners (especially those in secondary schools as well as in universities who lack the ability in abstract thinking) to learn such topics with the adequate level of engagement. The difficulties are also due to the restrictions to visualize computer programs. As a result, understanding the basics of CS concepts can be challenging for students in a learning environment which is centred heavily on lecture-based teaching. The way to respond to the challenges we present in this paper is based on the idea to combine the CS problem solving (through programming) with the visualization process using LEGO-based robots.

The role of visualization as an extremely important attribute in teaching and learning has been recognized a long time ago by the authorities within the e-learning community [1]. On the other hand, according to Bloom's taxonomy [2], education based on the use of the paradigm "learning by problem solving" is regarded as the most effective process in acquiring knowledge. What yet is needed to realize that is to build an e-learning environment. Though there are many approaches, we have selected the robot-based e-learning environment because it, in nature, enables to smoothly

integrate the problem solving and visualization, thus enhancing the motivation and engagement in learning. Furthermore, the approach has a strong technical support due to the availability of the well established robot kits for teaching such as the LEGO MINDSTORMS NXT [3], [4]. In a wider context, as it is stated in [5], Lego kits now are firing up young minds all over the world with the possibilities of technology.

As LEGO robot-based learning encompass many interrelated tasks (ranging from the robot constructing to its control and connecting to learning tasks), we need to restrict the scope of the topic. To make the topic relevant to the profile of the conference and the journal themes, we focus more on technical aspects of the approach here. Thus, the aim of the paper is to present the process of constructing a LEGO robot, called the DRAWBOT (meaning drawing robot), which enables to create the e-learning environment to demonstrate visually the solution of graph-based CS tasks through teaching programming. Creation of the e-learning environment means a mechanical construction of the robot and adaptation (selection) of its operating environment (OS and the language to design the robot control program) for teaching and learning purposes.

Our contribution is that we focus on design of the whole process meaning (a) DRAWBOT robot constructing, (b) testing its correct functioning and (c) demonstrating ability to solve and visually represent a set of the related graph-based tasks in teaching algorithms and programming.

The rest part of the paper is organized as follows. The next section describes related work. The following sections present the essence of the proposed approach, DRAWBOT constructing and testing for its correct functioning along with a case study of using the approach at secondary school classes to teach CS topics. The final section evaluates the approach and concludes the basic issues.

II. RELATED WORK

In the last two decades, educational robots offer new benefits by implementing the most effective active learning methods and supporting tools for teaching of science, technology, engineering and mathematics (STEM).

As it is stated in [6], most researchers work in this field and deal with problems such as ones: (a) the use robotics as an educational tool, (b) empirical testing of the effectiveness of robots and (c) defining future perspectives of the use of

the robots. A great deal of applications is "descriptive in nature, based on reports of teachers achieving positive outcomes with individual initiatives" [6]. The paper also summarizes the educational potential of robotics in schools and concludes as follows: (a) "most of the studies found are concentrated in areas related to robotics" per se (i.e., robot construction, mechatronics, robot programming); (b) a pre dominance of the use of Lego robots is observed (90%); (c) robotics increase learning achievements in STEM concept areas.

There is a lack of publications about the learning environments associated with the use of robots. We found only one paper, which proposes a learner-centred robotic enhanced environment based on the constructivist approach and a methodology to involve students to knowledge construction [7]. The related e-learning topics that use other tools are analyzed in other references.

III. ESSENCE OF THE PROPOSED APPROACH

We describe the approach by considering related tasks, processes and results achieved of using robots in teaching and learning. Tasks are as follows:

1) DRAWBOT's structure and functioning algorithm description (space 2D recognition problem, robot coordinates location within 2D space, coordinates of the gripper (pencil in our case) identification);

2) The use of DRAWBOT in teaching programming application task description (drawing of results of the computer programs as learning objects (LOs) that solve CS graph-based tasks).

Fig. 1 explains the whole process in designing and using DRAWBOT-based e-learning environment.
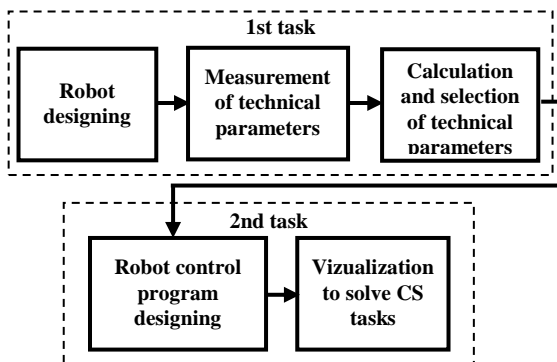

Fig. 1. Sequence of processes to create DRAWBOT-based e-learning environment.

B. Results of task 1

The LEGO NXT robot structure consists of the 4 main components:

1) Servo motors (Fig. 2, a);
2) The NXT Intelligent Brick (IB) (Fig. 2, b);
3) Sensors (Fig. 3, c);
4) The main chassis (not shown).

The NXT IB is a brick-shaped computer with a small 100×64 pixels monochrome LCD display and four buttons that can be used to navigate a user interface using hierarchical menu. The component can take control up three motors and input from four sensors via RJ12 cables. The Brick also contains loudspeaker, which can play sound files

(up to 8 kHz). The Brick is implemented by 32-bit ARM7 microprocessor with 256KB flash memory and 64KB RAM, in addition to a secondary 8-bit AVR microprocessor with 4 KB flash memory and 512B RAM. The Brick also has built-in Bluetooth for transferring programs and commands, and transmitting data of the sensor and motor feedback.
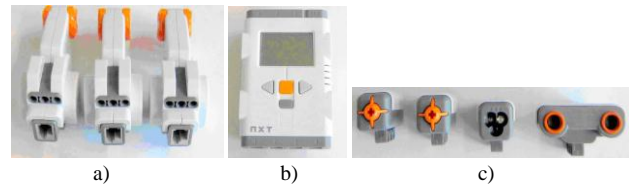

Fig. 2. a – NXT servo motors, b – NXT Intelligent Brick (IB) and c – standard sensors (touch, colour, ultrasonic).

Fig.3 illustrates steps in the DRAWBOT structural design using Lego Digital Designer (LDD) 4.2. Fig. 3, *a* explains the design process when LDD is used from scratch. After that, it is possible to create a building instruction in HTML for more effective constructing the LEGO NXT robots. In Fig. 3, b is shown a full view of DRAWBOT, designed by using building instruction.
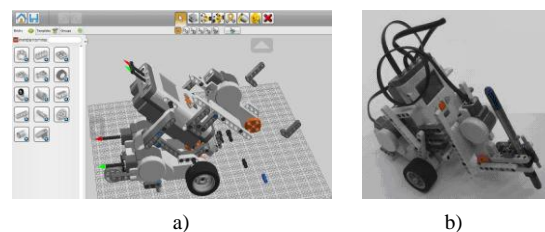

Fig. 3. DRAWBOT designing steps (Lego Digital Designer 4.2): a – model designing fragment; b – a full view of DRAWBOT.

The next important step within the process is the measurement of technical parameters of DRAWBOT, because these parameters are used for robot control program.

Motors are controlled for specifying a power level to apply to the motor. The programming language RobotC uses parameter named "Power level". Power levels range from –100 to +100. Negative values indicate reverse direction and positive values indicate forward direction. For example, to move motor A forward at 30% of full power, we would use the following statement: motor [motorA] = 30;

The distance driven by robot per time depends on the motor's Power level. The movement of the robot depends on robot's construction and technical parameters of the motors. To ensure the smooth movement there are three operating modes: 1) manual adjustment by the motor command "Power level" for the straight robot's move, 2) use of the PID (Proportional-Integral-Derivative) speed control algorithm, 3) use of the motor synchronization to ensure that both motors run at the same speed [8].

The first motor operating mode is an "open loop" configuration and does not result in a consistent speed across all motors; e.g., the speed will become lower as the batteries discharge and different motors will produce different speeds because of minor variations in the motors. The PID speed algorithm is a "closed loop" control algorithm uses feedback from the motor encoder to adjust the raw power to provide consistent speed. Motor synchronization allows us to control

the robot when the most important is the rectilinear movement.

The movement of DRAWBOT was improved by using the PID speed control algorithm and motor synchronization. Fig. 4 presents DRAWBOT's distance-Power level dependencies obtained experimentally. They are needed for correct robot operating to solve a teaching task (e.g., to smoothly make drawings (Fig. 5)).
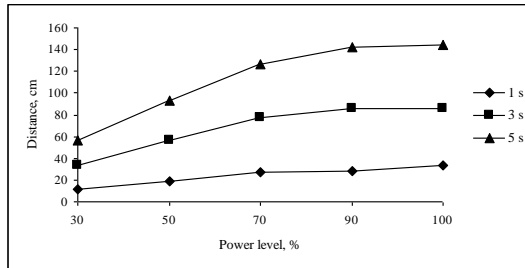
a)
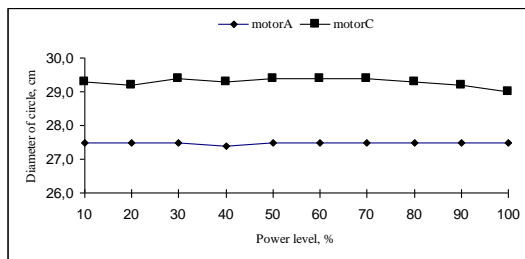
b)

Fig. 4. Drawbot's distance-Power level dependences: a – straight line movement; b – rotation movement (full circle).

IV. DESIGNING OF OPERATING ENVIRONMENT OF THE DRAWBOT

The NXT robot can be programmed: 1) by using the menu of the NXT IB, choosing actions and putting them into a simple flow-chart structure; 2) by using a third-party programming language (PL) (or a subset of a standard language), with the program being downloaded onto the NXT IB to run; 3) by using a third-party PL, with the program being run directly on a client computer and commands transferred to the NXT IB over the Bluetooth connection.

The main features of NXT robot software are presented in [9]. We selected and tested 3 languages: NXT-G, RobotC and LeJOS NXJ (Table I).

TABLE I. NXT PROGRAMMING SOFTWARE (ADOPTED FROM [9]).

Software Features	NXT-G	RobotC	LeJOS NXJ
Language type	Graphic	C	Java
OS	Windows, Mac OS	Windows	Windows Mac OS Linux
Bluetooth: brick to PC brick to brick brick to other device	Yes Yes No	Yes Yes Yes	Yes Yes Yes
Suited for	Beginners	C learners	Java learners
Learning advanced programming	No	Yes	Yes

We have chosen the RobotC for several reasons: 1) it is similar to C++, which is taught in school, 2) it uses the industry standard C PL with additional language extensions for robotic use, 3) it has a big number of sample programs and possibilities to use free tutorials, video lessons and detailed help files, 4) it is suitable to teach advanced programming concepts, 5) it is compatible with Robot Virtual Worlds Software.

V. CASE STUDY: RESULTS OF SOLVING TASK 2

We use the DRAWBOT as a learning environment for CS classes at secondary school. The DRAWBOT was used to teach the CS course topic "FOR loops in computer programs". Students are involved into three related activities under the constructivist learning model: 1) to construct mechanics of the DRAWBOT and control programs; 2) to design an application (programs as LOs); 3) to visually implement the ornament drawing task as a loop-based LO with the aid of DRAWBOT (Fig. 5).
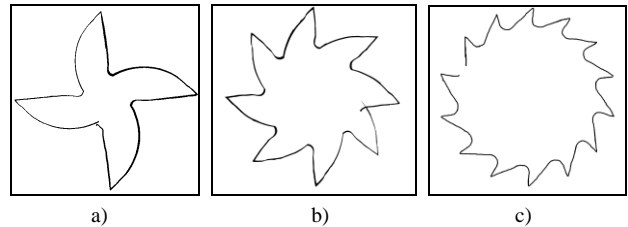

a)   b)   c)

Fig. 5. Ornaments produced by the DRAWBOT (parameters: Power level – 50% of motors A and C; movement duration in ms: a – 1000, b – 500, c – 250).

We evaluated the mean time required for each task execution (Table II).

TABLE 2. THE MEAN TIME REQUIRED FOR EACH TASK.

Task	Time (hours)	
	From scratch	Using guides or tutorials
Robot designing	4	1
Measurement of technical parameters	1	–
Calculation and selection of technical parameters	2	1
Robot control program designing	2	0,5

What tasks are not considered when using the DRAWBOT as compared to the real world robots? They are as follows: roundabout obstacles are ignored (there are no any obstacles within the 2D working space); the boundaries of working space are specified virtually (in the PC memory), but not physically on the paper's sheet; the precision of locations of nodes within the working space is ignored; the coordinate system of the 2D working space is fixed.

VI. DISCUSSION AND EVALUATION

The DRAWBOT e-learning environment provides sufficient functions to support teaching and learning (using a constructivist model) in programming at the secondary school classes for the advanced students. The e-learning DRAWBOT-based environment is versatile, popular among students, simple to implement and it has low costs. The

versatility means that 1) it is possible to deliver the interdisciplinary aspects of teaching (the tasks considered here are related to mechanics, physics, mathematics, informatics and CS); 2) it is possible to combine the problem solving paradigm with the visualization; 3) it is possible to bring for students the vision of how to model the real world tasks; 4) it is possible to increase the engagement level in learning for all students despite of theirs abilities and social differences as our experiment have shown. Though we have used the approach at school classes, the approach is applicable also at the university level. The university tutors can use the same paradigm for solving more complex tasks and inventing new capabilities in delivering such subjects as Informatics and Robotics (the curricular courses of most technical universities).

Furthermore, we have extended the learning and teaching environment with the automatic generation of simple programs (known also as LOs) for education in CS, which were derived from the so-called generative learning objects [10, 11]. This extension adds the additional value to the approach, but this is beyond the paper's scope.

VII. CONCLUSIONS

Our research has confirmed the importance of using robot-based tools for teaching that was known so far in e-learning. We have extended the known approach: a) by providing technical characteristics for the process to create the e-learning environment for the real setting; b) by smoothly integrating different phases of the process and considering it into entirety. We have identified that the robot–based e-learning environment extends highly the constructivist model of learning and teaching. We have identified also some difficulties to implement the approach. The main social barrier is the teacher's determination and the need of changing the mind in using the approach, and lack of previous knowledge.

REFERENCES

[1] W. Winn, "Visualization in learning and instruction: a cognitive approach", *Educational Technology Research and Development,* Springer US, vol. 30, no. 1, pp. 3–25, 1982.

[2] M. Forehand, *Bloom's taxonomy: Original and revised,* Emerging perspectives on learning, teaching, and technology, 2005. [Online]. Available: http://projects.coe.uga.edu/epltt/

[3] D. U. Ekong, "Resources For Using Robotics To Inspire Interest In Engineering", in *Proc. of the IEEE SOUTHEASTCON 09*, 2009, pp. 273–277.

[4] A.-M. Eubanks, R. G. Strader, D. L. Dunn, **"**A comparison of compact robotics platforms for model teaching", *Journal of Computing Sciences in Colleges*, vol. 26, no. 4, pp. 35–40, 2011.

[5] R. Courtland, "Lego Queen", *Dream Jobs*, IEEE, Spectrum, vol. 49, no. 2, pp. 50–52, 2012.

[6] F. B. V. Benitti, "Exploring the educational potential of robotics in schools: A systematic review", *Computers & Education*, vol. 58, no. 3, pp. 978–988, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.compedu.2011.10.006

[7] S. Frangou, K. A. Papanikolaou, "On the development of robotic enhanced learning environments", in *Proc. of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age*, 2009, pp. 18–25.

[8] M. Carnegie,"RobotC – Improved movement", Robotics Academy, 2007, p. 19. [Online]. Available: http://www.doc.ic.ac.uk/ ~ajd/Robotics/RoboticsResources/ROBOTC%20%20Improved%20 Movement.pdf

[9] *NXT Programming Software.* [Online]. Available: http://www.teamhas senplug.org/NXT/NXTSoftware.html

[10] V. Štuikys, R. Damaševičius, "Towards knowledge-based generative learning objects", *Information technology and control*, vol. 36, no. 2, pp. 202–212, 2007.

[11] V. Štuikys, R. Damaševičius, "Development of of Generative Learning Objects Using Feature Diagrams and Generative Techniques", *Informatics in Education*, vol. 7, no. 2, pp. 277–288, 2008.