# Using Robots as Learning Objects for Teaching Computer Science

**Renata Burbaitė**, *renata.burbaite@gmail.com*
Software Engineering Department, Kaunas University of Technology, Lithuania

**Robertas Damaševičius**, *robertas.damasevicius@ktu.lt*
Software Engineering Department, Kaunas University of Technology, Lithuania

**Vytautas Štuikys**, *vytautas.stuikys@ktu.lt*
Software Engineering Department, Kaunas University of Technology, Lithuania

## Abstract

We introduce a concept of Robots As Learning Objects (RALO) and a framework of methodology for robotics-oriented teaching of Computer Science (CS) topics based on the Constructivist and Empirical Modelling paradigms. Our methodology views robotic technologies not as mere tools, but rather as vehicles of new ways of thinking and reasoning about teaching, learning and education at large. We formulate the pedagogy-driven activities, technology-driven processes, knowledge transfer channels with actors involved, tools and facilities used, and the pedagogical outcome in the robot-based constructivist learning process and describe our experience in using RALOs for teaching CS at high school (gymnasium) and university levels.

## Keywords

education, learning object, robotics, computer science, intelligence.

## INTRODUCTION

Robotics is an exciting multi-disciplinary area that is going to dominate the 21st century. The robotics industry is entering a new period of rapid growth (Shukla & Shukla, 2012). For example, the year of 2011 has been named as the most successful year for industrial robots since 1961 (IFR, 2012). The current high school and university students will live in a highly technologized society surrounded by industrial and service robots at work, educational robots at educational institutions, assistive robots at hospitals and care facilities, and domestic/entertainment robots at home. As we are living in an increasing digital world, children should be taught how it works. Therefore the educational priorities must shift towards teaching students how to manipulate all digital devices (computers, robots, smart TVs, high tech gadgets, etc.) that surround them for their own needs (Brittain, 2011).

Robots can be seen as specialized computers with both computational and mechanical facilities to perform physical movement-oriented tasks. Robots allow demonstrating the capabilities of electronics technology and providing students with opportunities for project-based learning. In the context of e-learning, robots are increasingly seen as a means for enforcing engagement, excitement and fun in learning (Shamlian et al., 2006), promoting interest in mathematics, engineering and science carrier (Rogers & Portsmore, 2004), in-

creasing students achievement scores (Barker & Ansorge, 2007), encouraging problem solving (Mauch, 2001), promoting cooperative learning (Beer et al., 1999).

Recently, with the advance of technology, new technology-based models of teaching and learning are becoming more popular. Learning is being transformed from traditional classroom-centred education to education based on web-based resources (e-Learning) (Wiley, 2000) and mobile devices (m-Learning) (Houser et al., 2002), immersive learning in a context aware ubiquitous learning environment (u-Learning) (Jones & Jo, 2004), a context-aware environment able to offer ubiquitous personalized content (iLearning) (Kim et al., 2011), a context-aware system that overlays virtual educational information on the real world based on the learners location and needs (Augmented learning) (Klopfer, 2008).

New learning models are usually based on the concept of a Learning Object (LO) understood as "any digital entity, which can be used, reused or referenced during technology-supported learning" (adapted from (IEEE, 2000)). In most cases, an LO is a directly usable educational resource or a resource with adaptation for computer-aided teaching, such as an educational applet or a self-teaching module to be obtained from a DVD or a web site. In a wider context, a LO per se is a model to support reusability across large e-Learning communities (Liber, 2005), including mechanisms to support automation of reuse (Santacruz-Valencia et al., 2008). However, with the advances in learning technologies, traditional LO models such as metadata – content or hierarchical models based on content granularity (Balatsoukas et al., 2008), are not enough, because: 1) e-Learning is rapidly advancing, and we need to have more flexible, more adaptable, more personalized, and more contextualized LOs to support advanced e-Learning; 2) e-Learning has very wide choices of IT support (mobility, networking, tools, etc.) which so far have been underutilized.

This is the reason why, in this paper, we focus on Robots as Learning Objects. We introduce the concept of Robots As Learning Objects (RALO), propose a framework and a methodology of using RALO for teaching computer science topics, and describe our experience in using RALO for teaching at high school (gymnasium) and university.

## BACKGROUND OF THE METHODOLOGY

### Methodological backgrounds for using a constructivist approach and robots as LOs

Our approach is based on the Constructivist-Based Learning Model (CBLM) (Leonard, 2002) and the Empirical Modelling paradigm (Harfield, 2007). According to the constructivism, students are most successful in learning when they are given the opportunity to explore and create knowledge dynamically while working with projects that they are interested in, and to explore and test their ideas (Papert, 1993). This style of learning encourages students to create tools and environments that sustain projects that are meaningful for students (DeLuca, 2003). Empirical Modelling is concerned with creating and using empirically developed computer models. The word empirical here means that the learning process is guided by practical experience rather than theoretical knowledge (Harfield, 2007).

Constructivism-based learning is implemented in the context of e-learning through the use of Learning Objects (LO). The basic idea of our approach is that we consider educational robots as LOs. Usually LOs in the e-learning domain are understood as the digital content or learning / teaching material delivered to learners as an IT-based soft product (i.e., any digital entity (IEEE, 2000) such as text file, picture, program, etc.).

On the contrary, in our approach an LO is seen as a physical entity consisting of two parts: hardware (mechanical/electrical parts of the robot) and software (robot control programs). Both parts can be used for learning and knowledge transfer in separation as well as in aggregation. When only programs are considered as LOs, the robot is used an e-learning environment that can interpret programs and transform them into real world tasks

(e.g., draw ornaments on the sheet of paper, cutting vegetables in a kitchen, moving object from one place to another, solving maze tasks, etc.). When only hardware is considered as a LO, the main focus is given first to the construction of robot hardware to illustrate the principles of mechanical construction (e.g., centre of gravity, stability, sturdiness, etc.) and the concepts and laws of kinematics (e.g., forwards kinematics, inverse kinematics, degrees of freedom, steering geometries, etc.). When the entire robot is considered as a LO, the main focus is to specify robot's behaviour (i.e., a correct sequence of control actions) to perform the pre-scribed tasks (e.g., line following, wall following, roaming, collision avoidance, etc.).

### Framework of the approach

Figure 1 outlines 5 basic components of the framework and their interaction. These components can be abstractly identified as pedagogy-driven activities, technology-driven processes, knowledge transfer channels with actors involved, a set of tools and facilities used, and the pedagogical outcome. The latter is a final product that implements the learning goals (objectives) through the use of the framework in the real e-learning and teaching settings (in our case, in different classes at the gymnasium and university levels to teach CS topics). Similarly to any other product, the achieved pedagogical outcome has to be assessed. We anticipate three forms of the assessment: student self-assessment, teacher and expert assessments. Interaction among components is specified through knowledge transfer and feedback channels.

### Robot as Physical Learning Object

The compound process (named as RALO in Figure 1) is based on the new concept of Physical Learning Object (PLO), which extends the notion of an LO beyond the virtual domain (content, web page) to a physical domain (robot hardware and physical processes that are demonstrated by the hardware). Traditionally an LO is seen as "a collection of content items, practice items, and assessment items that are combined based on a single learning objective" (CISCO Systems, 1999). LOs provide self-contained, re-usable units of learning. They typically have a number of different components, which range from descriptive data to information about rights and educational level. The main components, however, are instructional content, practice, and assessment.

We define PLO as follows: a smart thing (as in "Internet of Things" (Haller, 2010)) that has sensors and/or actuators to interact with its environment and content (control program) to control its behaviour. We do not draw a distinctive line between software and hardware domains and postulate a LO as a physical thing that can be used to deploy, demonstrate and validate units of knowledge. We focus on teaching computer science (CS); therefore we interpret the "unit of knowledge" in terms of CS, i.e., an algorithm or a computer (or robot) program. We focus on a proactive construction of knowledge by students when the students themselves design, model and deploy a robotic LO (both its physical (hardware) and virtual (software) parts) and then provide research.

The idea to use smart things as LOs was raised by Specht (2009), who offered an example of a smart ball that "would be full of sensors that can measure the ball's speed, the pressure applied, direction of movement, and absolute location in a room. All this data would be visualised on a big screen and learners could just play with the ball and observe the sensor data in small experiments".

Designing of Robots as components of "Internet of Things" includes: 1) mechanical construction and electrical wiring of a robot; 2) design of the robot control program (Burbaitė et al., 2012); and 3) connection of a robot to web services to allow its virtual images (data representing the state and behaviour of the robot) to be published and shared on the

internet. This process requires a great deal of experiments and testing (in Figure 1 expressed as feedback FB1) that is performed by both teacher and students.

The process of Task selecting for CS topics is important part of the framework because the process should satisfy the requirements of many factors (teaching model, curricula, student involvement and technical possibilities of the developed robot-based learning environment (RALE).
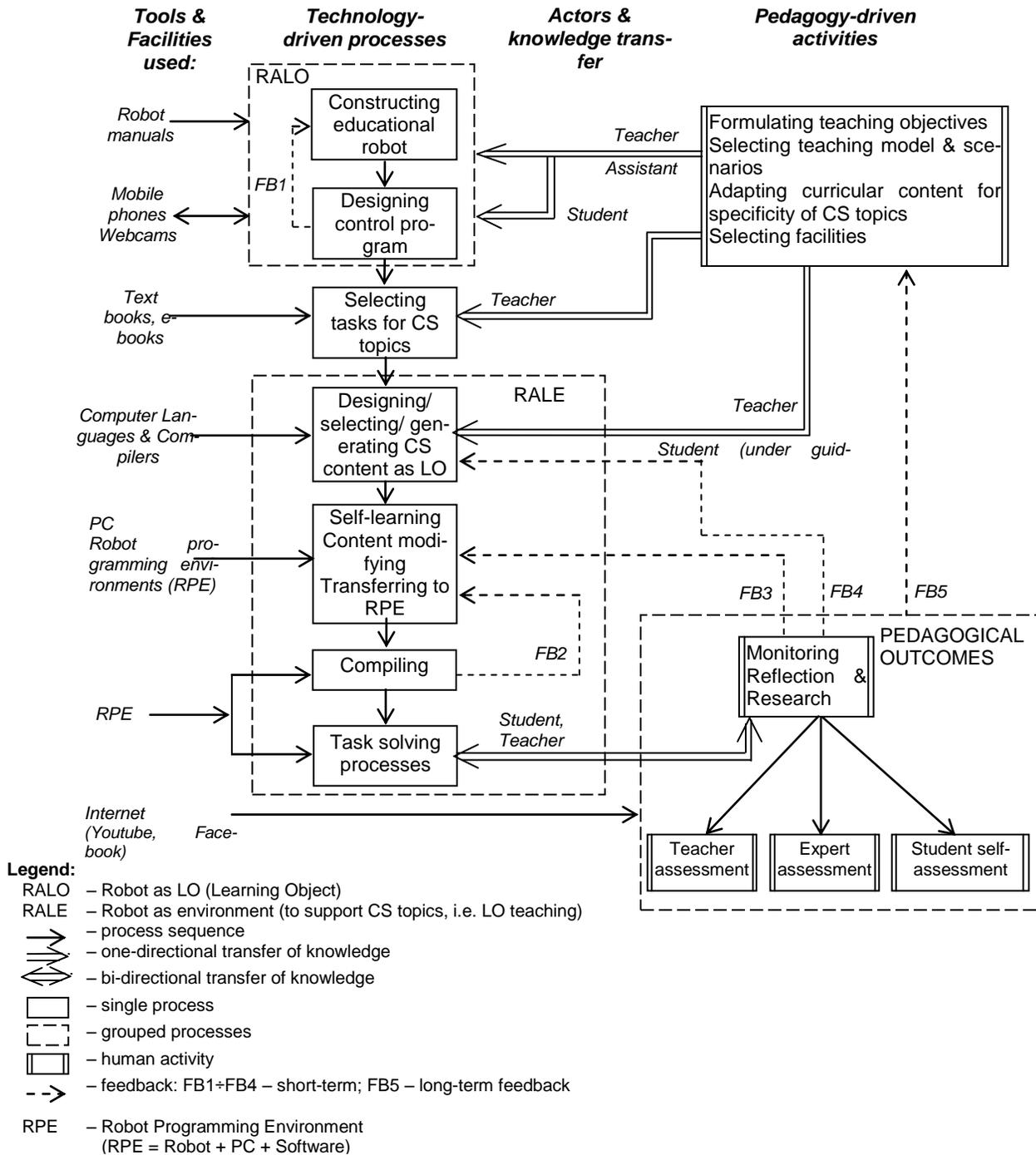


**Figure 1:** Basic components and their interaction within proposed framework

**Robot-based e-learning environment for teaching CS topics**

The learning process is based on using CS topics-related LOs. The latter can be either selected (e.g., from digital libraries (DLs), text books with or without adaptation), or newly designed (e.g., as solitary instances for each topic, or even automatically generated using generative LO approach (Boyle, 2009; Damaševičius & Štuikys, 2008). This process is also guided by pedagogy-driven activities. However, the student role in this process is minimal because only a small number of students (as apprentices of a teacher) can be involved. The next process (see Figure 1) includes Self-learning (e.g., after teacher's explanation of the LO to be learned), modifying (before and after compilation) and Transferring the LO to RALE. Compiling is a standard process with possible feedback for correction (FB2). The task solving process is at the core of learning because it integrates the knowledge gained through actions performed at the previous stages. The process is also the source of new knowledge gained through monitoring, reflection and research.

Note that the bi-directional knowledge transfer is quite different for high school and university students. The difference is imposed by topics specificity and the level of delivery. For example, university students are directed to knowledge-based robot teaching, where a great deal of research activities is required (see case study 2). High school students typically perform only monitoring, fixation, reflection and discussion with teacher on what they did before and what they see now. But, having in mind a variety of tasks (drawing ornaments, cutting vegetables, sorting (assignment) of coloured bowls into boxes, etc.) and the possibilities to repeat the processes (through feedbacks FB3, FB4), the students are also researching the problems at the level of their knowledge.

Assessment (teacher, expert and student self-assessment) is the result of monitoring, reflection and research activities. By the expert's assessment we mean the evaluation of teaching/learning outcomes by other teachers, e.g., through social networks or members of interest groups (e.g., LEGO Engineering group, legoengineering@googlegroups.com).

## CASE STUDY

The RALO-based teaching methodology (Figure 1) was validated at the high school (70 students) and university (32 students at the 3rd course of software system engineering studies) levels. We have adopted the structured approach (Dimitriou, 2012) towards using RALOs, which conceptually is similar both for high school and university, but differ in the scope and content of the theoretical learning materials presented, knowledge transferred and abilities gained.

The 1st stage consists on the presentation of the teaching topics and learning materials and is driven by the course curricula: at high school – Programming course (basics of algorithm, conditional statements, for and while loops, nested loops); at university – "Fundamentals of Artificial Intelligence" (basic principles of artificial intelligence and intelligent control) and pedagogical requirements.

The 2nd stage consists of introduction of tools and facilities required for solving tasks formulated by the teacher. The tools are Lego Digital Designer for mechanical design of a robot from scratch (high school only, see Fig 2, a) and Figure 3), Microsoft Robotic Developer Studio and its Visual Simulation Environment for simulation of robot control program (university only, see Figure 2, b), Lego Mindstorms NXT 2.0 for implementation of robot control algorithm (both high school and university).

The 3rd stage consists of selecting tasks (problems) for student projects. The students were asked to choose a project that was of interest to them and create it. They watched online videos and looked at pictures of sample projects to give them ideas. The topics of projects are summarized in Table 1 (high school) and Table 2 (university).
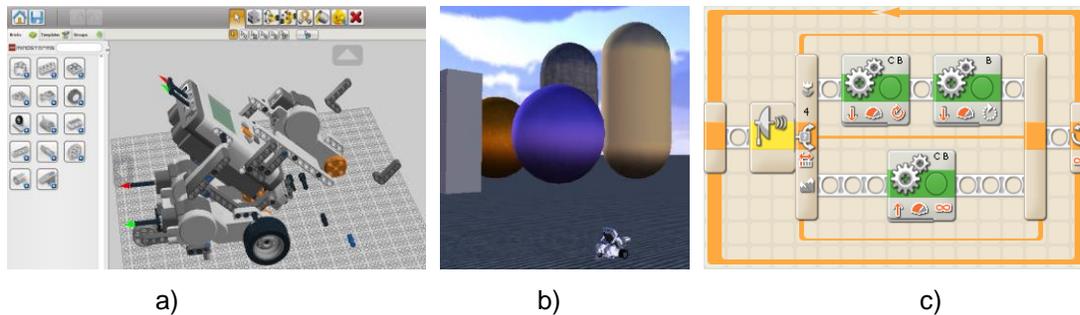
a)                                   b)                                   c)

**Figure 2:** The task steps: a) design (Lego Digital Designer 4.2), b) modelling (Microsoft Robotic Developer Studio 4), c) programming (Lego Mindstorms NXT 2.0)
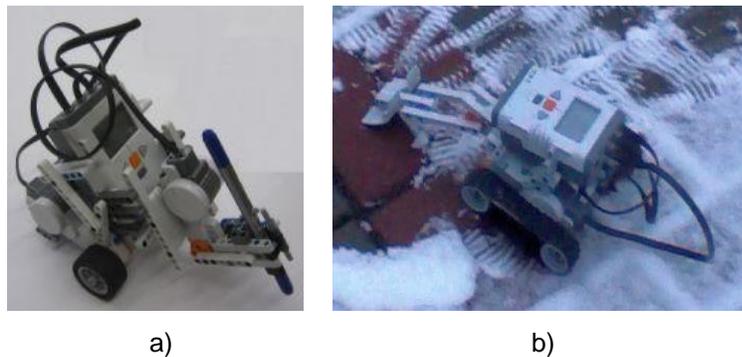


a)                                                   b)

**Figure 3:** View of physical robots used in high school for: a) drawing, b) snow cleaning

**Table 1:** Summary of robotics projects at high school

| No. | Tasks / project topics | Curricula topic covered | Grades/ Age | No. of groups (students) | Gained knowledge level* [1..5] (% of students) |
|---|---|---|---|---|---|
| 1 | Robot calibration | Linear algorithms (LA) | 10-12 / 16-19 y | 5 (12) | 1 – (100%); 2 – (83%); 3 - (75%); 4 – (33-67%); 5 – (17-58%) |
| 2 | Robot calibration | + Loops | 11-12/ 17-19 y | 7 (16) | 1 – (100%); 2 – (88%); 3 - (81%); 4 – (25-58%); 5 – (13-58%) |
| 3 | Snow cleaning | + Loops | 5-10/ 11-17 y | 12 (28) | 1 – (100%); 2 – (89%); 3 - (71%); 4 – (25-79%); 5- (13-58%) |
| 4 | Cutting | + Branching | 5-10/ 11-17 y | 12 (28) | 1 – (100%); 2 – (93%); 3 - (76%); 4 – (0-71%); 5- (0-15%) |
| 5 | Drawing | + Nested loops | 8-12/ 14-19 y | 23 (70) | 1 – (100%); 2 – (76%); 3 - (67%); 4 – (27-40%); 5 – (17-24%) |
| 6 | Sorting | All together | 8-12/ 14-19 y | 7 (20) | 1 – (100%); 2 – (70%); 3 - (55%); 4 – (20-40%); 5 – (3-5%) |

* 1- Viewing; 2 –Responding; 3 – Changing; 4 – Constructing; 5 - Presenting (for details, see Urquiza-Fuentes & Velázquez-Iturbide (2009) and Table 3)

**Table 2:** Summary of robotics projects at university

| No. | Tasks/project topics | Curricula topic covered | No. of groups (students) |
|---|---|---|---|
| 1 | Research of line following | Robot control algorithms | 2 (9) |
| 2 | Collision avoidance | Robot control algorithms | 2 (9) |
| 3 | Overcoming of hurdles | Brook's subsumption architecture | 1 (4) |
| 4 | Mobile room temperature monitor | Robot control algorithms; Robot sensors | 1 (4) |
| 5 | Mobile ultrasonic sensor | Robot sensors | 1 (2) |
| 6 | Stair climbing robot | Robot control algorithms; Robot sensors | 1 (4) |

The 4th stage included analysis of available solutions and selection (under the guidance of the teacher) of an appropriate solution.

The 5th stage consists of compilation and empirical validation (at high school – monitoring, at university - research) of task's solution.

Some examples of projects implemented by students follow (for high school teaching, see Figure 2 a), Figure 3 and Table 1). The aim of the university student project "Research of line following" at the University was to analyze the behaviour of the line-following robot. The students implemented two different line following algorithms (One Bounce, One Inside) (Gray, 2003), and different types of wheel motion control when turning (see Figure 4). The implementation was empirically validated for different levels of motor power (25%, 50%) and with several routes (different number of turns, angles) (see Figure 5).
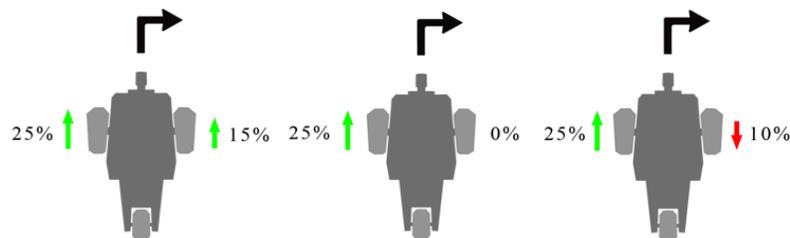


**Figure 4:** Different types of robot control when turning
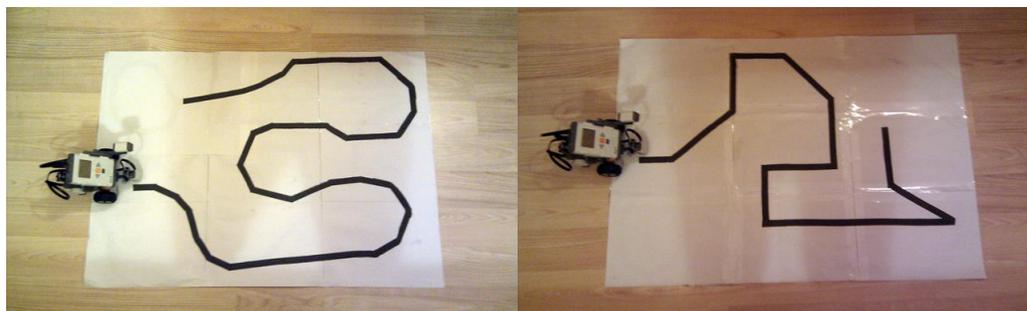


**Figure 5:** Line following using different types of routes

The results of experiments performed by the students allowed them to discover the influence of factors unforeseen in the initial model of a robot such as noisy environment (unevenness of the line, difference in ambient lighting, occasional speckles), faulty sensor data and the unpredictability of results (small changes in the initial position of a robot lead

to different trajectories of robot approaching route angles which in turn lead to robot succeeding or failing the route). The pedagogical outcome was the understanding gained by the student that robot control problem is much more difficult than first anticipated and that "soft" adaptive solutions are required to allow robot to tackle problems unforeseen by the robot programmer.

The final stage of the process includes evaluation of project results and assessment of projects pedagogical outcomes both by students and teachers. This included making oral presentations for other students and a teacher at the end of the course lectures.

## EVALUATION

We have evaluated the pedagogical effectiveness of MPB GLOs using the engagement level-based methodology (Urquiza-Fuentes & Velázquez-Iturbide, 2009) (see Table 3).

**Table 3:** Evaluation of projects based on student engagement level

| Engagement level | % of students at high school (see Table 1) | % of students at university |
|---|---|---|
| *Viewing*: Students view the solutions given by teacher passively | 100 | 100 |
| *Responding*: Students use the solution actively as a resource for answering questions given by teacher | 70-93 | 94 |
| *Changing*: Students *themselves* change the parameters of the solution to achieve better characteristics (robot behaviour) | 55-81 | 88 |
| *Constructing*: Students construct their solutions for solving a problem | 0-79 | 84 |
| *Presenting*: Students present for discussion their solutions | 3-58 | 84 |

Finally, in Table 4 we present the evaluation of robots as learning objects (RALO) based on the properties/attributes of LOs formulated by Wiley (Wiley, 2000).

**Table 4:** Evaluation of robots as Learning Objects

| Property/attribute | Comment |
|---|---|
| Reusability | The same robot can be reused multiple times at multiple learning contexts for multiple purposes (different courses, different pedagogical aims) |
| Generativity | Robot programs can be combined using, e.g., a subsumption architecture (Brooks, 1991) to have complex robot behaviour implemented |
| Adaptability | Robots are modular and can be quickly reassembled and reprogrammed to match different tasks |
| Scalability | Robots are scalable, i.e., after an initial investment no additional significant costs are required to teach different concepts of CS |

## CONCLUSION AND FUTURE WORK

In this paper we have presented a robotics-based approach towards CS education that is based on the interpretation of a robot as a Learning Object per se. Our methodology

views robotic technologies not as mere tools, but rather as potential vehicles of new ways of thinking about teaching and learning at large. As such a robot can be understood as: a) an intelligent system that can 1) act in the physical environment and perform tasks to achieve its objectives, 2) interact with the physical environment  and take decisions, 3) communicate with other robots and/or computer(s) (smart phones, web services, etc.); and b) a learning object that can be used as a tool to 1) engage a student to study, 2) visualize a computer science topic, 3) teach the concepts of computer programming and robot control, 4) demonstrate how the program works, 5) provide feedback to a student, and 6) introduce with the basic principles of scientific research.

Apart from allowing Computer Science and Programming to be taught in a more effective way, this approach also directly addresses a gap in current teaching concerning how to support the learning of practical skills needed in the fast growing market of robotics and embedded computing.

Future work includes a systematic integration of the concept of generative learning objects within robot-based e-learning environment (for high school student teaching) and enforcing of the knowledge-based approaches within robot environments (for university student teaching).

## ACKNOWLEDGEMENTS

## REFERENCES

Balatsoukas, P., Morris, A., & O'Brien, A. (2008). Learning Objects Update: Review and Critical Approach to Content Aggregation. *Educational Technology & Society, 11(2),* 119-130.

Barker, B. S., & Ansorge, J. (2007). Robotics as Means to Increase Achievement Scores in an Informal Learning Environment. *Journal of Research on Technology in Education, 39(3),* 229–243.

Beer, R. D., Chiel, H. J., & Drushel, R. F. (1999). Using autonomous robotics to teach science and engineering. *Communications of the ACM, 42(6),* 85–92.

Boyle, T. (2009, March). *Generative learning objects (GLOs): design as the basis for reuse and repurposing.* Invited presentation in First Int. Conference on e-Learning and Distance Learning, Riyadh, Saudi Arabia.
Retrieved from http://ipac.kacst.edu.sa/edoc/2009/173839_1.pdf

Brittain, N. (2011, September 20). IT education remains mired in uncertainty. *Computing.* Retrieved from http://www.computing.co.uk/ctg/analysis/2110259/education-remains-mired-uncertainty

Brooks, R. A. (1991). Intelligence Without Representation. *Artificial Intelligence 47,* 139-159.

Burbaitė, R., Štuikys, V., & Marcinkevičius, R. (2012). The LEGO NXT robot-based e-learning environment to teach computer science topics. *Electronics and Electrical Engineering, 18(9),* 113-116.

Cisco Systems. (1999). *Reusable information object strategy.* Retrieved from http://www.cisco.com/warp/public/779/ibs/solutions/learning/whitepapers/el_cisco_rio.pdf

Damaševičius, R., & Štuikys, V. (2008). On the Technological Aspects of Generative Learning Object Development. In R.T. Mittermeir & M.M. Syslo (Eds.),  *Third Interna-*

*tional Conference on Informatics in Secondary Schools - Evolution and Perspectives*, *ISSEP 2008 Torun Poland, July 1-4, 2008 Proceedings* (pp. 337-348). Springer-Verlag. doi: 10.1007/978-3-540-69924-8_31

DeLuca, D. (2003). *Robotics and Teaching: Promoting the Effective Use of Technology in Education* (Honors thesis). Tufts University, MA, USA. Retrieved from http://ceeo.tufts.edu/robolabatceeo/references/thesis/DianaDeluca_undergrad.pdf

Dimitriou, K. (2012). A more structured way to teach robotics with robotics. In *Proceedings of 3rd International Workshop Teaching Robotics Teaching with Robotics Integrating Robotics in School Curriculum* (pp. 163-169). Retrieved from http://www.terecop.eu/TRTWR2012/trtwr2012_submission_26.pdf

Gray, J. A. (2003). *Toeing the Line: Experiments with Line-following Algorithms*. Retrieved from http://www.fll-freak.com/misc/01-jgray_report.pdf

Haller, S. (2010). The Things in the Internet of Things. In *Proceedings of Internet of Things Conference*. Tokyo. Retrieved from http://www.iot-a.eu/public/news/resources/TheThingsintheInternetofThings_SH.pdf

Harfield, A. J. (2007). *Empirical Modelling as a new paradigm for educational technology*. (PhD thesis). University of Warwick, UK. Retrieved from http://wrap.warwick.ac.uk/2404/1/WRAP_THESIS_Harfield_2007.pdf

Houser, C., Thornton, P., & Kluge, D. (2002). Mobile learning: cell phones and PDAs for education. In *Proceedings of International Conference on Computers in Education*, vol.2 (pp. 1149-1150).

IEEE (2000). *IEEE Learning Standards Committee, WG 12: Learning Object Metadata*. Retrieved from http://ltsc.ieee.org/wg12

IFR (2012). *International Federation of Robotics. World Robotics 2012 Industrial Robots. Executive summary*. Retrieved from http://www.ifr.org/industrial-robots/statistics/

Jones, V., & Jo, J. H. (2004). Ubiquitous Learning Environment: An Adaptive Teaching System Using Ubiquitous Technology. In *Proceedings of the 21st ASCILITE Conference* (pp. 468-474). Retrieved from http://ascilite.org.au/conferences/perth04/procs/pdf/jones.pdf

Kim, S., Song, S.-M., & Yoon, Y.-I. (2011). Smart Learning Services Based on Smart Cloud Computing. *Sensors, 11(8),* 7835–7850.

Klopfer, E. (2008). *Augmented Learning: Research and Design of Mobile Educational Games*. The MIT Press, 2008.

Leonard, D. C. (2002). *Learning theories, A to Z*. Greenvood Publishing Group, Inc.

Liber, O. (2005). Learning objects: conditions for viability. *Journal of Computer Assisted Learning 21,* 366–373.

Mauch, E. (2001). Using technological innovation to improve the problem solving skills of middle school students. *The Clearing House, 75(4),* 211–213. Retrieved from: http://home.wlu.edu/~kuehnerj/cv/clearinghouse_2001.pdf

Papert, S. (1993). *The children's machine: Rethinking school in the age of the computer*. New York: Basic Books.

Rogers, C., & Portsmore, M. (2004). Bringing engineering to elementary school. *Journal of STEM Education, 5(3&4),* 17–28.

Santacruz-Valencia, L. P., Navarro, A., Delgado Kloos, C., & Aedo, I. (2008). ELO-Tool: Taking Action in the Challenge of Assembling Learning Objects. *Educational Technology & Society, 11 (1),* 102-117.

Shamlian, S. V., Killfoile, K., Kellogg, R., & Duvallet, F. (2006). Fun with Robots: a Student-taught Undergraduate Robotics Course. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation* (pp. 369-374). Orlando, Florida, USA.

Shukla, M., & Shukla, A. N. (2012). Growth of Robotics Industry Early in 21st Century. *International Journal of Computational Engineering Research (IJCER), 2(5),* 1554-1558.

Specht, M. (2009). *Learning in a technology enhanced world*. Open Universiteit Nederland. Retrieved from http://dspace.ou.nl/bitstream/1820/2034/1/Oratio_Specht.pdf .

Urquiza-Fuentes, J., & Velázquez-Iturbide, J. Á. (2009). Pedagogical Effectiveness of Engagement Levels – A Survey of Successful Experiences. *Journal Electronic Notes Theoretical Computer Science 224,* 169-178.

Wiley, D. A. (2000). *Learning Object Design and Sequencing Theory* (Ph.D. dissertation). Brigham Young University, Utah, USA. Retrieved from http://opencontent.org/docs/dissertation.pdf

## Biographies

**Renata Burbaitė** is a PhD student at Software Engineering Department of Kaunas University of Technology (KUT) and a teacher of Computer Science (CS, Informatics) at Gymnasium. She also provides teaching of students in Informatics at KUT. Her research interests relate to the e-learning domain that includes the use of meta-programming-based Generative Learning Objects and Educational Robots to teach CS topics. She is a member of Interest Group of Informatics Teachers.

**Robertas Damaševičius** is a Professor at Software Engineering Department, KUT. He received his PhD (2005) degree in informatics engineering from KUT. Currently he teaches programming, robotics and software engineering courses. He is also the member of Design Process Automation Group at Software Engineering Department. His research interests include program transformation and meta-programming, design automation and software generation, as well as domain analysis methods. Currently he is supervisor of 3 PhD students. He is also member of ACM.

**Vytautas Štuikys** is a Professor of Software Engineering Department at KUT and holds the degree of Habilitate Doctor of Science. His research interests include system design methodologies based on reuse and automatic program generation and transformation. He is author of about 100 papers in this area and co-author of the monograph „Meta-Programming and Model-Driven Meta-Program Development: Principles, Processes and Techniques"published by Springer. His specific interest is the application of the design methodologies to the e-learning domain. Currently he is a head of research group at KUT and supervisor of 3 PhD students. He is a member of ACM.