

Centrobot Portal for Robotics Educational Course Material

Richard Balogh¹, Adrian Dabrowski², Walter Hammerl², Alexander Hofmann³, Pavel Petrovič⁴, Ján Rajniček⁴

¹Institute of Control and Industrial Informatics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, 81219 Bratislava, Slovakia, richard.balogh@stuba.sk,

²Höhere Technische Bundeslehr- und Versuchsanstalt Spengergasse, Spengergasse 20, 1050 Wien, adrian@atrox.at, hammerl@spengergasse.at,

³Department of Computer Science, University of Applied Sciences Technikum Wien, Höchstädtplatz 5, 1200 Wien, alexander.hofmann@technikum-wien.at

⁴Department of Applied Informatics, Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynská dolina, 842 48 Bratislava, Slovakia, ppetrovic@acm.org, jan.rajnicek@st.fmph.uniba.sk

Abstract—This paper describes a prototype built as part of the Centrobot project. We present a web application intended for sharing robotics educational material. The system will allow browsing and editing a large set of projects, which we call **robotivities** (robotic activities). The aim of the portal is to supply information, robotivities and metadata, which can be used by teachers or pupils to educate and to learn. It will be the place, where users can discuss and improve their skills, find useful material and educational procedures: <http://portal.centrobot.eu/>

Keywords - robotics education, portal, robotics projects

I. INTRODUCTION

The amount of experience collected in various applications of robotics technology in education in our and other groups has reached a critical mass, e.g. [1,3,4,5]. However, results from pilot studies, didactic materials, lesson plans, laboratory exercises, instruction sheets, manuals, simple and complex student projects, ideas for activities, links to suitable contests and other activities need to be easily accessible by the educators. Otherwise, the construction sets and educational robotic kits purchased, leased or borrowed by individual educational institutions are likely to finish locked up in a cabinet. Consequently, they would never be used properly and efficiently.

To support this argument, let us compare the number of schools that obtained the robotics sets LEGO Mindstorms Robotics Invention System in Slovakia through the governmental project Infovek from the year 2000 with the number of teams participating in robotics competitions RoboCup Junior and/or FIRST LEGO League. More than 120 elementary and secondary schools received the sets. However, only about 15% of them (18) participated in the competitions. All of the schools were invited, but more than half of them is not using the sets and does not respond to letters and e-mails from the sets distributor.

The reason is obvious. Even though the sets are a wonderful tool for interdisciplinary constructionist learning, there has been a lack of didactic materials and those existing have not been sufficiently accessible. The producer of the sets has made a pensum of projects and materials available, many

of them are contributions of the community, especially in the most recent period. However, a large amount of material and activities is created by other third parties, and particularly by the teachers. Material is produced in local languages, and tailored for the needs, learning style, and common sense of the pupils and teachers from the specific region.

Most authors harness the potential of the modern media, especially the Internet, and they use the local websites, or general-purpose publishing sites to make their achievements available. The missing piece is the integration of these resources into one accessible location, where the resources would be classified, structurally and topically arranged, and provided in a standardized, understandable, and easy to use format. Centrobot portal aims to provide this missing piece. At the first place, it is focused on collecting the materials from our geographical area – Austria and Slovakia. However, the portal is open and all parties interested to share are invited and welcome.

In the following sections, we will describe the functionality provided by the Centrobot portal, explain its structure, and the whole concept. We have also implemented and describe a prototype that provides most of the intended functionality.

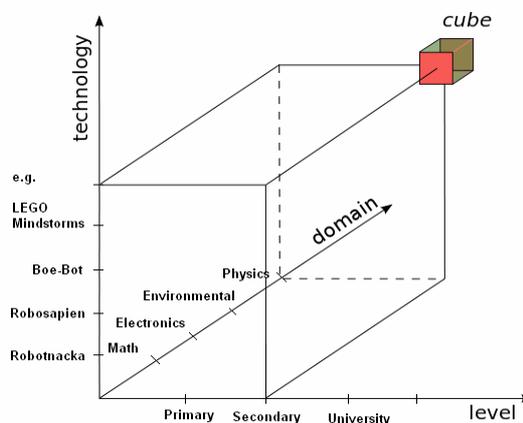


Figure 1. The Centrobot cube concept.

II. A GENERAL IDEA

The Centrobot Robotics Educational Course Material Portal is a website for browsing, publishing, editing, and deployment of robotics educational material. It consists of individual units that we call *robtivities*. These are all kinds of activities, lesson plans, projects, quizzes, contests, etc. Robtivities are classified along the three main axes: *level*, *technology*, and *domain*. A particular location described by the three coordinates along these axes typically contains several robtivities. We call such location a *cube*, see Fig. 1. One robtivity can exist in multiple cubes.

By *level*, we mean the age of the robtivity audience. Level spans from the preschool education – kindergardens (up to age 7), through elementary schools (8-10), lower and upper secondary schools (10-14, 15-18) , and colleges (>18). Furthermore, the respective levels are structured further, shaping groups of beginners, intermediate, and advanced learners, depending a) on their level of knowledge and b) their level of expertise with particular technology.

By *technology*, we mean a classification of the various technological robotics hardware and software platforms used in the schools. For instance, LEGO MINDSTORMS NXT sets is one of the hw platforms, yet, it can be used in connection with many different software platforms, such as RoboLab, NXT-G, NXC, Java (Lejos), NXT logo, etc. Other hw platforms may include Parallax BoeBot, our prototyping platforms Sbot [9] and Robotnacka [4], robots Bee-bot, Probot, Roamer, MaVIN, Asuro, Aldebaran Nao, or RoboNova, for instance. Other sw platforms may include Basic-Stamp, AVR Studio, Imagine Logo, Spin, Microsoft Robotics Studio, and many other.

By *domain*, we mean the application area, school subject, or field of study. Examples are physics, mathematics, introduction to programming, mechanical engineering, or artificial intelligence at the coarse level. At several levels of higher detail, the robtivities can be classified into particular subfields, such as constructive geometry, frequency and period, mechanical wave motion, acceleration, search, heuristics, localization, etc.

For each robtivity, the specific coordinates along the three axes form its main classification. However, each robtivity can be assigned a set of general keywords in addition. The keywords can be arranged in hierarchies.

Moreover, a general-purpose material is typically relevant for all robtivities that share the same coordinate along some axis (battery charging procedure, Java language manual, etc.). Such shared materials may be placed to the container associated with the specific axis category, instead of replicating it in each and every one robtivity sharing that coordinate. The shared materials are then directly available when viewing any of the relevant robtivities.

The purpose of the classification of robtivities into the cubes is to allow for a possibility of systematic browsing. A teacher planning to setup a course may want to acquire an overview of the availability of the various robtivities relevant for his course, before dwelling more deeply into details of the planning and the implementation phases. In order to support this scenario, all robtivities must also contain the basic information, we describe below.

III. ROBTIVITIES

A robtivity footprint in our system consists of different information that can be valuable to the teachers when planning for their course, browsing the portal and selecting a robtivity. For lucidity and better comprehension, this meta-information is arranged in a standardized structure with five main categories: Organizational Parameters, Implementation, Technical Parameters, Support, and Resources. Registered users have the option to download all material of one robtivity for off-line use in one archive. The cube coordinates and didactical information are compulsory parts, the remaining elements are optional, but it is highly recommended that they are filled in.

A. Organizational Parameters

The parameters describe the first pieces a user needs to learn when assessing whether a particular robtivity is suitable for their aims.

Cube coordinates

Technology – e.g. LEGO Mindstorms NXT and NXC programming language;

Domain – e.g. Mathematics – Number Theory – Fractions;

Level – e.g. Lower Secondary – Introductory Level – No Previous Technical Skills;

In addition to the three main coordinates, the robtivity also has its *Language* – e.g. English, optional pointers to other language variations, *Version* – e.g. 1.0, and *Author(s)* – who developed it and who are responsible for the entry.

Didactical Information

Content – a short description of the course content and targets;

Required knowledge – a general or special knowledge required by students or instructor ;

Time consumption – for each of the activities: preparation by instructor, theory lesson, practical session, postprocessing.

Related Robtivities – lists other relevant robtivities:

Robtivities with preparational content;

Robtivities with similar content;

Robtivities with advanced content;

B. Implementation

If a robtivity appears suitable, a more detailed examination is needed to learn whether it can be implemented in the prospective lesson or course.

Course preparation

Environment – e.g. classroom, lab, mountain side, ...

Equipment required – hardware, how to source (supplier links), measurement equipment, software development environment;

Presentations – material used for explaining the course content, aim, theory to students (format: PDF, PPT, optionally other formats, such as DOC, ODT);

Papers – any material used by students during the course to reach the targets, forms to be filled-in, etc.

Proceeding

Description of how the lesson is carried out – how many students, is it a team work or individual work, how to setup and start, how and what is to be observed, measured, reported, questions that are to be answered;

Sample solution – examples of hardware description, examples of code, pictures.

Multimedia artifacts (Audio, Video, Images) – in standardized formats.

C. Technical parameters

Robotics involves a complex technology and thus in addition to organizational and proceeding didactical information, robtivities provide a special place for non-didactical technical information that are or may be needed for successful deployment of the robtivity.

Construction manual

A general explanation of how to build the robot for this course, what is its main function, degrees of freedom, special requirements, modes of operation.

Description of components

All components that the users need to manipulate, e.g. sensors, microcontroller, power supply, software environment. For each relevant component, its user interface, interconnection parameters, and all technical details can be documented.

D. Support

Forum for discussions

Simple forum for exchange of experience, interesting observations during the lesson, problems and typical solutions (input to FAQs), etc. The authors are automatically notified by e-mail, when new questions appear.

Contact

Allows sending a direct e-mail message to the authors.

FAQ

Is a list of typical questions and answers that is maintained by the robtivity authors.

Rating and Feedback

A simple schema for grading the robtivities. The users who downloaded the robtivity as a single package, and those who selected it for their course are notified to provide rating and feedback after one week.

E. Resources

Robtivity resources

This section lists all referred publications, links, and all other resources that are important or useful for the robtivity.

General resources

Information that is useful for all robtivities that share the same technology, level or domain, i.e. user and programming guides, general didactical methodology, scientific resources, handbooks, textbooks, encyclopedia, and other. They appear in this section based on the robtivity coordinate classification automatically. Users may add new items and specify the respective scope of relevance for each item.

IV. TYPICAL USE SCENARIOS

This section demonstrates selected most typical use sequences of actions when working with the portal. Let us first explain the user types the system supports:

guest, usually an unauthorized teacher, student, or another visitor, who can view almost all the published content, search, browse, and participate in forum discussions, the only

exception is viewing the sample solutions, which are accessible only to authorized teachers;

authorized teacher, who can create new robtivities, edit the robtivities he authored, and administer the content contributed by others to these robtivities; authorization is also required for downloading a cube or a robtivity for off-line use, and for uploading media, solutions, and resources;

authorized student, who have the same privileges as unauthorized student, with the exception of viewing the page that was created for his/her class or course by his/her teacher;

administrator, who can perform site maintenance, moderate all discussions, and edit or delete content.

A. Anonymous user searching for useful content

- select coordinates, alternatively select keywords
- view the resulting list of robtivities
- display the robtivity details one by one
- view individual documents, pictures, videos
- find one robtivity that is the most suitable, copy the robtivity URL for use in her course

B. Teacher preparing a course based on several robtivities

- create a teacher account, or login to an existing one
- be confirmed by another teacher or admin, if needed
- create a virtual classroom – an account for student authorization
- search for robtivities, and add them to the virtual classroom
- provide the student account name to the student in her class
- teachers can create copies of their virtual classrooms for repetitive use in multiple classes
- in each virtual classroom, the teacher can view the solutions submitted by the students

C. Teacher adding a new robtivity

- login to the system, or create a teacher account if new
- selects the most suitable cube, and eventually adjust the categories along one or more axes
- add a new robtivity, upload all relevant files, provide the obligatory and optional parameters
- can take a break and come back later to complete
- can add more authors who can edit all the robtivity content
- when the robtivity editing is finished, the author marks it as published

D. A student works with a specific robtivity

- logs into the virtual classroom as instructed by the teacher
- arrives at a list of preselected robtivities and a commentary prepared by the teacher
- can browse the whole site, if appropriate
- may upload his results for later inspection by the teacher

E. Admin performs the site maintenance

- logs into the system
- can run broken links detection, view their list, and respective pages, correct the links manually
- can configure the system backup and recovery

- can administer accounts, forums, edit or delete or robtivities
- can send an information e-mail to authors

V. ROBTIVITY LIFE CYCLE AND SYSTEM MAINTENANCE

Robtivities are prepared by one or several authorized teachers during a period of time, when no other users may access them. Once the robtivity is completed, one of the authors may mark it as published, and it becomes part available at the portal for all the users, who can then rank the robtivity by assigning certain number of stars, add feedback in form of forum comments, or questions, upload their sample solutions, and resources, or send message to the authors. The authors can further edit their robtivity, add, change, or delete the content as appropriate, answer forum questions, appoint other authors who can continue maintaining the robtivity. Administrator always has the rights to remove inappropriate content, delete the whole robtivity, or deny access of some author, if needed. In normal circumstances, the administrator will first contact the author to perform the changes. If all authors do not respond for very long period of time, and the information in some robtivity becomes obsolete, the administrator may choose to add a new author, who will continue maintaining the content. Eventually, outdated robtivities can be marked as outdated or possibly removed from the site.

The system performs a system log of all operations that are performed in the system – for both debugging, tracking and maintenance reasons. The log is saved into database, and can be viewed by the admin.

The system is designed to work in multiple languages, English, German, and Slovak versions are available, other can be added when needed.

VI. EXAMPLE ROBTIVITIES

In this section, we provide two example robtivities as they are published in the Centrobot portal and verified in the classroom. The first one is using the popular NXT construction sets, while the second utilizes Sbot educational robotics platform [9] and used in the exercises of an introductory college course on robotics. For better readability, the information is presented in tables. In Centrobot portal, the information is shown in a structured format and viewed in a webbrowser.

A. Line-Following NXT Robot

| Organizational Parameters | |
|----------------------------------|---|
| Cube Coordinates | |
| <i>Technology</i> | LEGO MINDSTORMS NXT and NXT-G |
| <i>Domain</i> | Introduction to Robotics – Simple applications |
| <i>Level</i> | Lower secondary school – Beginners – No technical skills required |
| <i>Version</i> | 1.1 |
| <i>Language</i> | English |
| <i>Author(s)</i> | Walter Hammerl |

| Didactical Information | |
|---|---|
| <i>Content</i> | Building and programming a robot, which is able to follow a black line |
| <i>Required knowledge</i> | NXT-G basics |
| <i>Time consumption</i> | Preparation by instructor: 1 hour Theory lesson: 30 minutes Practical session: 2 units 50 minutes each Postprocessing: 30 minutes |
| <i>Paths</i> | ◀ Introduction to NXT-G ▶ Passing a maze with NXT ▶ Spatial orientation |
| Implementation | |
| Course preparation | |
| <i>Environment</i> | Classroom with enough space to layout the line follower course |
| <i>Equipment required</i> | - 1 pcs LEGO Mindstorms NXT per student - bright plane with a closed black line (e.g LEGO test pad 8547) - sheet of paper with a black line (2-3cm) - stopwatch to compare the individual performance - 1 PC with NXT-G Software 2.0 per student - 3 light sensors per NXT Alternatively, the lesson can be hold with groups of 2 students per robot |
| <i>Presentations</i> | (link to Line-follower presentation) |
| <i>Papers</i> | (link to Spreadsheet to record the lap times), real-world application of the line-following robots [6] |
| Proceeding | |
| <i>Description of how the lesson is carried out</i> | The instructor may start with a brief introduction to the theory of line following and eventually a review of NXT-G programming. The students shall build their line following robot themselves. In case they start from scratch the manual for building the basic NXT model is helpful. Students shall be encouraged to think about different ways to mount the sensors and to build their individual model. Discuss mechanical parameters for mounting the sensors: <ul style="list-style-type: none"> - distance between the sensors - clearance from the floor - distance from drive wheels, etc. Before starting to program the robot, the students shall provide a flow diagram, which may be reviewed by the instructor. Start programming the robots and test using the test pad. Good practice is testing the correct function of the robot by placing it on top of a support box, the wheels can move freely (the table continues below...) |

| | |
|---|---|
| | (continues from above...) without floor contact, and a piece of black and white paper. Record the time the robot needs to complete a given number of laps. Encourage the students to modify and improve the algorithm. |
| Implementation | |
| <i>Sample solution</i> | 3D model of the robot, example of code: linefollower.rbt, pictures: myrobot.jpg |
| <i>Multimedia</i> | Video: linefollower.avi |
| Technical parameters | |
| Construction material | |
| <i>How to build the robot for this course</i> | main function: Simple robot with 2 degrees of freedom, special requirements: 3 light sensors |
| <i>Description of components</i> | Easy start model, Line sensors LEGO Mindstorms user manual |
| Support | |
| FAQ | |
| <i>Question</i> | When I program the motors to move with unlimited duration they stop after a few seconds. What shall I do? |
| <i>Answer</i> | The program will set the motor operation to unlimited, the motors will speed up, and then the program will carry out the next command. If this is the end of the program the motors will stop. The simplest way to is to put the MOVE block within a LOOP block set to control=forever. See also the presentation material. |

B. Bayesian Robot Programming with Sbot

| | |
|----------------------------------|--|
| Organizational Parameters | |
| Cube Coordinates | |
| <i>Technology</i> | Sbot and AVRStudio |
| <i>Domain</i> | Artificial Intelligence – Bayesian Robot Programming |
| <i>Level</i> | College – Intermediate students – Intermediate Sbot users |
| <i>Version</i> | 1.0 |
| <i>Language</i> | English |
| <i>Author(s)</i> | Pavel Petrovič |
| Didactical Information | |
| <i>Content</i> | The students will study learning in a robot with a probabilistic model of the world. The robot will learn simple behaviors of pushing and following objects. |
| <i>Required</i> | Theoretical framework of BRP, Bayes |

| | |
|---|--|
| <i>knowledge</i> | formula, basic operation of Sbot robot |
| <i>Time consumption</i> | Preparation by instructor: 1 hour Theory lesson: 90 minutes Practical session: 90 minutes Postprocessing: 30 minutes |
| <i>Paths</i> | ◀ Introduction to SBot ▶ Sbot localization |
| Implementation | |
| Course preparation | |
| <i>Environment</i> | Laboratory with a small arena and rectangular obstacles |
| <i>Equipment required</i> | - 1pcs Sbot per group - flat surface and obstacles of various shapes - a PC with AVRStudio development environment and BlueTooth connection |
| <i>Presentations</i> | (link to presentation Introduction to Bayesian Robot Programming) |
| <i>Papers</i> | (link to a journal paper on BRP) (link to detailed instruction, source, report) |
| Proceeding | |
| <i>Description of how the lesson is carried out</i> | Students first control the robot remotely for a short period of time. The program samples the sensors and builds a probabilistic representation of the behavior policy. When the robot runs in autonomous mode after the learning session, it should successfully push the obstacles. Students repeat the experiment with following the obstacle walls. Students modify the code based on instructions and observe, measure, document the outcome. |
| Implementation/Proceeding (cont.) | |
| <i>Multimedia</i> | Video: sbot_brp.avi |
| Technical parameters | |
| Construction material | |
| <i>How to build the robot for this course</i> | The robot is already built. Students only have to make sure the distance sensors are properly mounted and connected. The robot schematic and software framework is described in the user manual (link). |
| Support | |
| FAQ | |
| <i>Question</i> | If the robot is connected to a port > COM8, the terminal program cannot connect. |
| <i>Answer</i> | This is because the serial ports from COM9 in Windows do not exist. Instead, Microsoft invented the names \\.\COMXX, such as \\.\COM42. Some programs require this format for ports higher than COM8. |

VII. PROTOTYPE DESIGN AND IMPLEMENTATION

Our prototype has been specified and designed by the partners participating in the Centrobot project and the first version was implemented by a bachelor student at FMFI UK in Bratislava, Ján Rajniček [2]. Detailed use-case, sequence, and class diagrams specified in UML can be found there, or in the Centrobot internal technical documentation.

A. Technology

Our prototype is platform-independent, but it runs on a Linux server machine with Apache, MySQL, and Tomcat server installed. It is based on JSP and Java servlets [7], XSL-FO, XML, and MySQL with Hibernate framework, and PHP for forums on the server side, and HTML, CSS, and JavaScript (AJAX) [8] on the client side. It relies on third-party components Log4j for logging, and TinyMC for WYSIWYG editing. The backup is realized with crontab and bash shell scripts. This setup allows for interactive graphical user interface in web browser without being bound to the traditional request-response HTTP model, i.e. the scripts running on the client side are in a permanent communication with the server-side scripts and database without the need to reload the pages.

B. Design

The system architecture diagram is shown in figure 2.

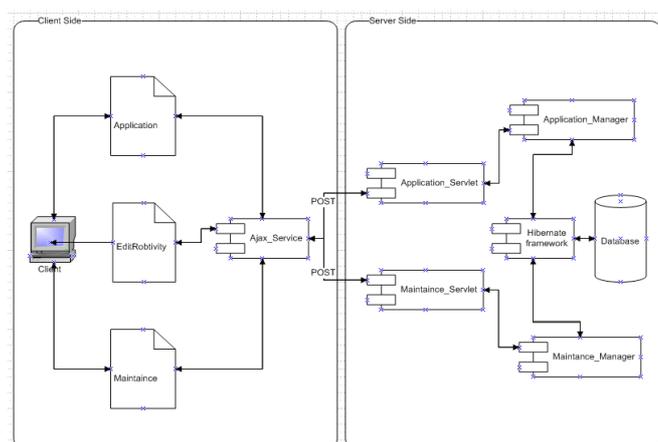


Figure 2. Architecture of the portal.

The system consists of 9 modules and a database. The *Application module* is a webpage with scripts running on the client allowing the users to login, or create their account and edit the profile, create, show, search or rank robtivities, access the robtivity forum, download cubes, upload content, setup, login, view, and delete virtual classrooms, and log all activity. *EditRobtivity module* is based on TinyMCE, and allows editing the various contents of the robtivities. Links to the files that were uploaded can be added to the text. *Maintenance module*, accessible only to administrator provides all admin functionality as described above. The last module that runs on the client side is the *Ajax_Service module*, which mediates the communication with all server modules using the standard XMLHttpRequest object. The server-side resembles this structure. *Application Servlet module* processes the requests from Application, and passes them further to Java package *Application_Manager module*, while the *Maintenance Servlet module* processes the admin's requests and calls the Java

package of the *Maintenance_Manager module* to perform the server-side admin's functionality. Both manager modules utilize the *Hibernate framework module*, a database interface.

VIII. CONCLUSIONS AND FUTURE WORK

We have designed and specified a detailed concept for educational robotics web portal containing set of various robtivities, i.e. projects and activities for classrooms and/or after-school clubs and centers. A prototype of the portal has been implemented and deployed at <http://portal.centrobot.eu/> also available through Centrobot project. Initially, it collects robtivities from our geographical locations in Austria and Slovakia. It is open for a new content provided by third parties.

Our aim is to run ensure a longterm operation of the portal, convert all the materials we collected and created in the past so that they will be available for general public. This involves fixing, debugging and improving the functionality, staying in contact with the content authors, collecting and processing their feedback. For instance, we would like to add a possibility to create on-line questionnaires or simple quizzes that could be filled-in by the students and automatically evaluated by the system based on the data provided by the teacher. The current search capabilities could be extended to provide an advanced and intelligent search options, and automatic filling of terms in the search form

ACKNOWLEDGEMENT

Project Centrobot (ATMOS N_00023) is supported within the Slovak-Austrian cross-border cooperation programme 2007-2013 'Creating the Future' financed by the EU European Regional Development Fund.

REFERENCES

- [1] T. Kraft, A. Hofmann, "Vienna Cubes – Robocup F180 Small Size League, Team", ÖGAI Journal 03/2003, Österreichische Gesellschaft für Artificial Intelligence, pp 14 – 19, 2003.
- [2] J. Rajniček, "Portal of educational robotics for the Centrobot project", bachelor thesis, FMFI UK Bratislava, 2010 (in Slovak).
- [3] R. Balogh, "Basic Activities with the Boe-Bot Mobile Robot", DidInfo 2008, 14th International Conference. FPV UMB, 2008.
- [4] P. Petrovič, R. Balogh, A. Lúčny, R. Weiss, "Using Robotnačka in Research and Education", Eurologo 2007 Conference, FMFI UK, 2007.
- [5] P. Petrovič, A. Lúčny, R. Balogh, D. Ďurina, "Remotely-Accessible Robotics Laboratory", Acta Mechanica Slovaca, Vol. 10(2-A), pp. 389-194, 2006.
- [6] J. Svetlík, M. Sukop, "Robocar - mobilný robotický vozíkový systém," Acta Mechanica Slovaca. vol. 12, no. 2-a pp. 617-624, 2008.
- [7] N. Z. Zakas, Professional JavaScript for Web Developers, Wiley, 2009.
- [8] B. Burd, JSP: JavaServer Pages, M&T Books, 2001.
- [9] Gušťařík, David and Krasňanský, Pavol: Robot SBot 2.0. Student scientific conference, FEI STU, 2008, (in Slovak).