SyRoTek—Distance Teaching of Mobile Robotics

Miroslav Kulich, Jan Chudoba, Karel Košnar, Tomáš Krajník, Jan Faigl, and Libor Přeučil

***Abstract*—E-learning is a modern and effective approach for training in various areas and at different levels of education. This paper gives an overview of SyRoTek, an e-learning platform for mobile robotics, artificial intelligence, control engineering, and related domains. SyRoTek provides remote access to a set of fully autonomous mobile robots placed in a restricted area with dynamically reconfigurable obstacles, which enables solving a huge variety of problems. A user is able to control the robots in real time by their own developed algorithms as well as being able to analyze gathered data and observe activity of the robots by provided interfaces. The system is currently used for education at the Czech Technical University in Prague, Prague, Czech Republic, and at the University of Buenos Aires, Buenos, Aires, Argentina, and it is freely accessible to other institutions. In addition to the system overview, this paper presents the experience gained from the actual deployment of the system in teaching activities.**

***Index Terms*—Educational technology, mobile robots, robot programming, telerobotics, user interfaces.**

I. Introduction

ARTIFICIAL intelligence and mobile robotics play an increasingly important role in everyday life and are becoming an inseparable part of many industrial applications. This creates higher demands on the education of new professionals who must not only operate advanced robotic systems, but also understand their behaviors so as to be able to design them and their particular components. According to current trends in education, even in kindergarten, young children become familiar with the world of science and technology by playing with robotic toys like Bee-Bot [1], [2]. Older children in primary and secondary schools use construction kits like LEGO Mindstorms or Fischertechnik to design and build their own robots and program and control these models making use of a graphical software [3], [4]. These initial training activities introduce them to the basic principles of robotics and motivate them to further study of technical domains.

However, these robotic toys suffer from poor sensor equipment and fragile construction, which means they cannot be used for real-world and long-term experiments. Universities teaching robotics use more powerful platforms like Videre Erratic or Pioneer and build their own robots equipped with state-of-the-art sensors. The main drawbacks of this approach are the initial cost and the necessity for continuous maintenance that can be costly and time-consuming. This is even more significant when more than one robot is used.

An alternative to acquiring and maintaining such equipment lies in sharing the required robotic hardware between several institutions. Altin [5] presents a robotic theater, where a project team equipped with LEGO Mindstorms robots visits schools in Estonia to attract them to join a national educational robotic project. Furthermore, virtual laboratories that allow the control of real robots over the Internet have been investigated and developed by several robotic labs. Early teleoperating systems developed in the 1990s allow a user to view and interact with a single hardware device (either a robotic arm or a mobile robot) [6]–[9]. One of the first integrated, remotely controlled robotic systems is the ARL Netrolab project [10], started at the University of Reading, Reading, U.K., in 1993. It allows control of a robotic arm and provides access to selected sensor equipment (sonars, infrared range finders, and a set of cameras). The e-laboratory project [11] combines a remote robotic platform with a virtual laboratory.

While there are many remote labs with robotic arms, remotely controlled mobile robots are not so common. One of most popular systems was developed in the RobOnWeb project at the Swiss Federal Institute of Technology in Lausanne (EPFL), Lausanne, Switzerland [12]. Five fundamental services of the Web interface were defined: chat, video, robot control, virtual robot representation, and logging. Several setups were developed within the project, differing mainly in the robot platforms and sensors used: TeleRoboLab, AliceOnWeb, Koala on the Web, and Pygmalion on the Web. In the REAL project [13], four frames were used to provide remote access to an autonomous mobile robot. The first frame provides the basic access to the laboratory and reservation system. The second frame realizes the teleoperated access to the robot. The additional frame allows the user's navigation module (written in C programming language) to control the robot. During the autonomous robot navigation, sensor data are collected by the user's module and stored in the dedicated user space for a further processing. The last frame represents a distance learning module.

A Web-based remote laboratory that gives an opportunity to remotely experiment with various navigation algorithms on a mobile robot is described in [14]. The system consists of a Web interface for uploading the user's program to the robot, accessing the on-board video camera, and monitoring various state variables. The software framework provided is built on the Microsoft Robotics Studio.

A combination of a simulated environment with a physical setup was applied in the LearnNet project [15]. The VRML technology was used to model the real environment at the user side, while only coordinates of objects are transmitted over the

Manuscript received May 21, 2012; accepted August 07, 2012. Date of publication November 12, 2012; date of current version January 30, 2013. This work was supported by the Technology Agency of the Czech Republic under Project No. TE01020197, the Ministry of Education of the Czech Republic under Projects No. 7AMB12AR022 and 2C06005, and the Ministry of Science of Argentina under Project No. ARC/11/11.

The authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, 166 27 Prague, Czech Republic (e-mail:kulich@labe.felk.cvut.cz; imr@labe.felk.cvut.cz).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2012.2224867

Internet. A set of robots is accessible for users in the Virtuallab project [16]. Several cameras monitored a playing field, and a user can use a combination of several views to get better overview of the robots' movements. The robots can be controlled remotely via the ActiveX technology or by a program in C++, Delphi, or Java. An open-source solution based on the Player/Stage framework [17] was planned in another virtual robotic laboratory project [18], which unfortunately seems to be no longer active. Teleworkbench [19] is a complex system that allows multirobot experiments with Khepera and Bee-Bot robots in an environment automatically built and controlled by a gripper. It provides precise robot localization based on image processing, online video stream and video recording, GUI with augmented reality for robot control, and a tool for experiment analysis. However, this system is currently unavailable.

The design of the SyRoTek[1] system presented here is inspired by the aforementioned projects, and so is similar to them in many ways. Modern technologies, tools, and hardware and software components used in the system increase its robustness and usability and allow it to outperform the foregoing systems at least in the following aspects.

1) Number of robots: SyRoTek offers 13 robots that are permanently (except when being charged) available to users. All these robots can be used simultaneously allowing multi-robot experiments.
2) Sensor equipment: The robots are equipped with a huge set of standard sensors used in robotics. To the best of the authors' knowledge, the SyRoTek system is the only system providing access to robots with laser range-finders. The modular architecture of the SyRoTek robot hardware design means that they can easily be equipped with new additional sensors in the future.
3) Interfaces to robotic frameworks: SyRoTek provides interfaces to Player/Stage and ROS, software frameworks widely used in the robotic community. SyRoTek is probably the only virtual laboratory providing interfaces for these two widely used frameworks. To the authors' best knowledge, no other virtual laboratory connects even to one of these frameworks. The current systems provide their own interfaces or are based on a closed-source and commercial software (Webots).

The rest of this paper is organized as follows. Section II gives an overview of the system and describes its key components. Section III presents courses taught with the support of the SyRoTek system, while Section IV discusses statistics gathered when using the system for educational purposes. Concluding remarks are presented in Section V.

II. System Overview

SyRoTek is a complex system consisting of many hardware and software components that are connected together and communicate using defined interfaces, shown in Fig. 1. The main components are 13 mobile robots operating in an enclosed space, the Arena. The robots communicate through WiFi with the *Control computer* that mediates a user's access

[1]http://syrotek.felk.cvut.cz

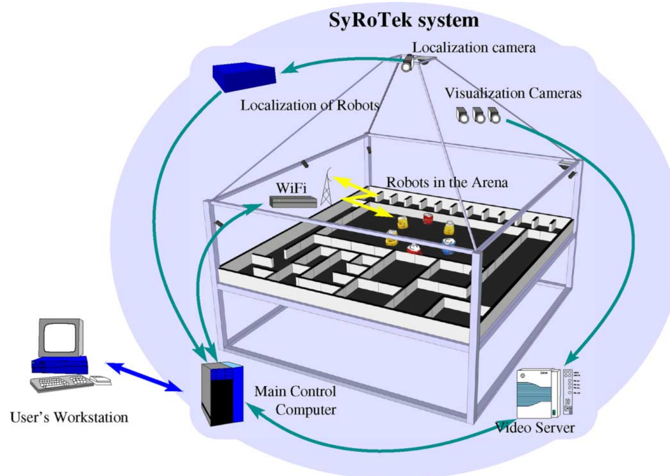

Fig. 1. SyRoTek overview.

to the robots. Moreover, it provides key functionalities of the system: manages the actual state of the robots and the Arena, prepares the robots according to reservations, manages information about users and the tasks they solve, stores teaching material and users' files, and so on. The *Video server* processes video streams from the cameras placed over the Arena, and *Localization* provides information about the positions of the robots in the Arena.

The key feature of the system is the ability to control the robots with applications developed by the user. The Player/Stage framework has been selected as the main programming interface because it is widely used in the robotic community; it supports a variety of mobile robots and sensors and contains a huge library of state-of-the-art algorithms for obstacle avoidance, planning, localization, etc. The Stage is a simulator, which substitutes for real hardware with the same interface. The user can therefore develop his/her code with the simulator, and that same code, with no or few modifications, can be used for a real robot. Player drivers for SyRoTek have been developed that provide access to the individual robots. The user can therefore use these robots in the same way as any other robots/devices via standard Player proxies.

With the increasing popularity of Robot Operating System (ROS) [20], the users also requested support for this framework. The internal communication within the SyRoTek is based on a developed efficient middleware for real-time data collection, which does not add the overheads typically caused by the message marshaling. Therefore, an additional interface can easily be developed, and the ROS interface was added into SyRoTek recently; the robots are thus accessible as ROS nodes providing all robots' sensors.

A. Robots

The SyRoTek robot is called S1R, and its body consists of the main chassis and an optional sensor module [21]. The robot is based on a differential drive with a maximal velocity designed to be 0.35 ms^{-1} and operating time about 8 h. The on-board computer (OBC) is the Gumstix Overo Fire module with the ARM Cortex-A8 OMAP3530 processor unit operating at 600 MHz

Fig. 2. *(left)* SyRoTek Arena and *(right)* S1R robot.

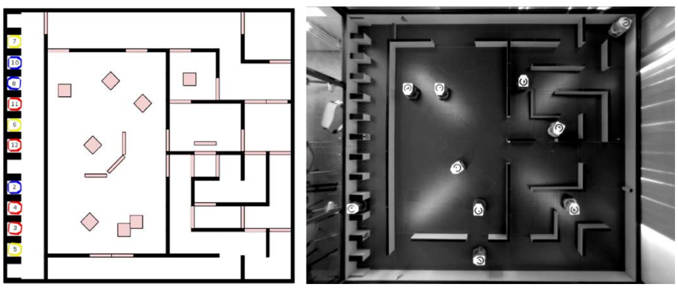

Fig. 3. *(left)* Schematic and *(right)* real view of the Arena. The movable obstacles (squares and rectangles) are light; the fixed obstacles are dark.

and running the Linux kernel. The connection with the control computer is provided by the integrated WiFi module of the Overo board.

The chassis serves as a carrier of basic sensors: five infrared range finders (Sharp GP2D120), three sonars (Devantech SRF10), 12 floor sensors, and a compass (Philips KMZ51). A camera module based on Gumstix Caspa will be added in the next revision. In addition, temperatures are measured in various places on the robot body. The currents to the motors are measured to provide the so-called software bumpers.

Additional sensors can be connected as a so-called front sensor module to the sensor bus or directly to the OBC. Two types of the mountable front sensor module are available. The first module is equipped with three sonars and three infrared range sensors, and it is connected to the sensor bus. The second module is the laser range-finder Hokuyo URG-04LX, and it is connected to the OBC via the USB interface.

B. Arena

The Arena is an enclosed space where the robots operate together with supporting subsystems: charging, lighting, and localization and visualization cameras. The size of the Arena, 3.5×3.8 m^2, is constrained by the space available in the dedicated computer lab where it is located, as shown in Fig. 2. The Arena contains an open area for multirobot exercises (e.g., formation control, flocking behaviors, etc.) and a maze-like area intended for localization, mapping and navigation tasks, as shown in Fig. 3.

The Arena is designed to be reconfigurable without need of human intervention. This feature is achieved by installing 37 movable obstacles, which can be retracted under the surface in less than 10 s. These movable obstacles allow various configurations of the Arena to be prepared before a user starts solving his/her task and tasks to be defined within a dynamically changing environment. The Arena workspace can also be divided into several separate closed areas to create independent working spaces. The other 50 obstacles are fixed, but can be removed manually, so a major reconfiguration of the Arena is possible by human intervention. The retractable obstacles are controlled through the Player interface, therefore a user can modify the environment even during the experiment (if this is permitted for the current task).

When a robot finishes the user's task (i.e., the user's reservation is expired), it autonomously navigates into a docking station. Thirteen docking stations are located at one side of the Arena, and each dock allows automatic recharging of the robot. A completely discharged robot is fully charged in about 1–2 h.

C. Localization

The purpose of the localization system is to determine the positions and headings of individual robots in the Arena. The system consists of a camera mounted above the Arena, a dedicated PC, and identification patterns attached to the top of each robot. The patterns consist of a black arc used for position and heading determination and a circular binary code used for robot identification. An image processing algorithm is based on convolution filtering and runs at 12.5 Hz. The system has proven to be capable of continuous and errorless operation for extended periods of time (several weeks) with precision of ± 1 cm in robot position and $\pm 3°$ in robot orientation.

Information provided by the localization system is used mainly for system purposes (docking and preparation of robots) and for visualization. Users can access this information only when it is explicitly allowed in the task definition written by the task creator (teacher).

D. Visualization

The visualization subsystem provides information about the actual situation in the Arena and allows online monitoring of experiments. In addition, it facilitates creation of offline videos for demonstration, archiving, and evaluations [22].

The hardware part of the subsystem consists of three IP cameras placed above the Arena. One camera provides an overview, and two cameras scan opposite sides of the Arena. The video streams are transmitted to the video server, from which they are streamed to users over the Internet in several formats and resolutions. Furthermore, the streams are stored on the server for a further download and offline editing and viewing. A user can watch the video online with GStreamer [23] or from the SyRoTek Web pages through an embedded viewer. The last possibility is to use a so-called "schematic view," an html5 canvas on which the robots and obstacles are drawn according to their positions from the localization system and information about the state of the retractable obstacles, as in Fig. 3. This technique avoids the necessity of transmitting large video files and is therefore suitable for low-bandwidth networks.

Moreover, Stage has been modified to be a standalone application, with fully customizable multiview support, able to play embedded video and visualize real sensor data, providing another way for users to watch the current situation in the Arena.

E. User Interfaces

A user can access SyRoTek by three interfaces: Web pages, NetBeans integrated development environment (IDE), and the command line interface. The Web facilitates a user's first steps with the system and helps with administration and pedagogical tasks. It provides links to documentation, news about the system, video files, and live video streams from the Arena and serves as the main source of information about courses offered. At this time, the system offers several standard courses, each consisting of lectures, assignments, and quizzes. The lectures contain supporting material like slides, references, etc., while the assignments specify a particular problem to be solved together with its software and hardware requirements. The quizzes are Web forms consisting of several questions that the students are requested to answer to prove their understanding of the presented topic.

The aforementioned materials are accessible for both nonregistered and registered users. Registered users can enroll in the courses, solve the assignments, take quizzes on these tasks, and above can reserve the Arena and the robots. These users can also communicate with teachers, system administrators, and other users on the forums provided.

The Web also supports teacher activities: course creation and editing, course copying from an existing course, quiz and task creation, and automatic evaluation. A teacher can display a list of students in a course, evaluate particular tasks and quizzes, and manually modify the evaluation. The Web (with information about student's reservations) provides an overview of student activities and progress and gives a teacher a full control of the teaching process.

A substantial part of students' work with SyRoTek is related to writing, compiling, and debugging source codes. Therefore, a set of plug-ins were developed to enhance capabilities of the NetBeans IDE supporting the development process, especially for inexperienced users. The enhancements allow users to easily start developing applications for SyRoTek within the selected IDE. The provided plug-ins bring the comfort of code completing, context help, visual debugging, and access to components of the SyRoTek system.

SyRoTek provides its own project template, which allows the user to choose the course and the task to be solved. The template prepared by the teacher for the given task is loaded to the IDE automatically, as a new project with a skeleton code and the settings of the application to be developed, as well as the Player/Stage configuration files for both simulation runs and experiments with the real robots.

The plug-ins integrate Player and Stage into the IDE. The integrated visualization window provided by the modified Stage allows easy monitoring of the experiment. Moreover, the user's application and Player can be run from the IDE either locally or at the control computer with one click, while their output is redirected to the Output window within the IDE. The user can test the code in the Stage simulator first, and after that run the application with real robots.

The plug-ins also provide access to live video streams from the visualization cameras, SyRoTek Web, and the reservation forms. To simplify the reservation process, the course and task are preselected as corresponding to the current project; the user only has to select a reservation time and a required duration.

Advanced users, who prefer a command-line environment, can use standard Unix tools and a set of SyRoTek scripts providing the same functionality as the SyRoTek Netbeans plug-ins.

III. Courses

Since 2011, the SyRoTek system has been used in courses taught at the Czech Technical University in Prague (CTU), Prague, Czech Republic, and the University of Buenos Aires (UBA), Buenos Aires, Argentina. The *Introduction to Mobile Robotics* course, given as a part of the School of Informatics at UBA in July 2011, was attended by 70 students with different *a priori* knowledge of robotics (ranging from beginners to post-graduate students studying robotics). The aim of the course was to introduce students to the main problems, and the available solutions, in controlling an autonomous mobile robot. Eight lessons providing a theoretical overview of the field were accompanied by 3-h practical labs, where students solved wall following, obstacle avoidance, or wandering problems independently in teams of two or three under teacher supervision. Throughout the course, the students worked with SyRoTek as a source of teaching materials, a communication channel with the teacher, to complete quizzes, and mainly as an experimental platform. The work in the lab, scheduled at the end of the course, was performed in the university computer laboratory, where SyRoTek client software (NetBeans with the SyRoTek plug-ins and the extended Player/Stage) was installed.

Although the students had no previous experience with developing a robotic application or even little programming experience, all the teams were able to control the robot in the simulator in the given time. Some applications were not perfect, as the controlled robots were colliding with obstacles. However, applications of three teams were able to control real robots in Prague using the SyRoTek server. A live video showing the real situation in the Arena was projected on the lab wall, letting the students see the real behavior of the robots controlled by their applications.

At CTU, the SyRoTek system was engaged in the *Practical Robotics* course in the winter semester 2011–2012. This introductory course is designed to create an interest in intelligent mobile robotics, and therefore the emphasis is on the supervised individual student's work. The course consists of six 90-min theoretical lectures followed by 14 135-min weekly lab sessions. The task to be solved is an exploration problem, a robot navigation program to create a map of the environment. The course was attended by 12 students split into teams of two or three. The students became familiar with the system after the first few sessions, then they were able to work individually. At the end of the semester, all the students were able to perform the exploration using the real robots.

Moreover, about 10 students used SyRoTek during their Bachelor's or diploma theses in the areas of multirobot exploration, formation control, and swarm robotics.

IV. Evaluation

The users/students of SyRoTek-based courses are encouraged to use Subversion (SVN) [24], a version control system

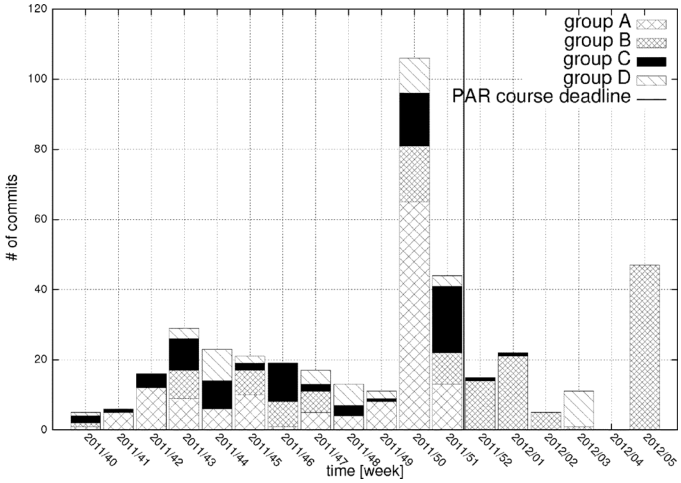

Fig. 4. Subversion activity of the selected teams of *Practical robotics* course in the winter semester 2011–2012.

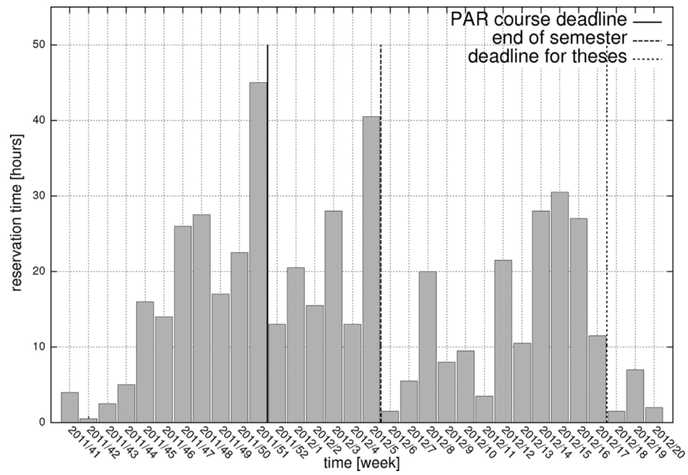

Fig. 5. Arena reservations per week.

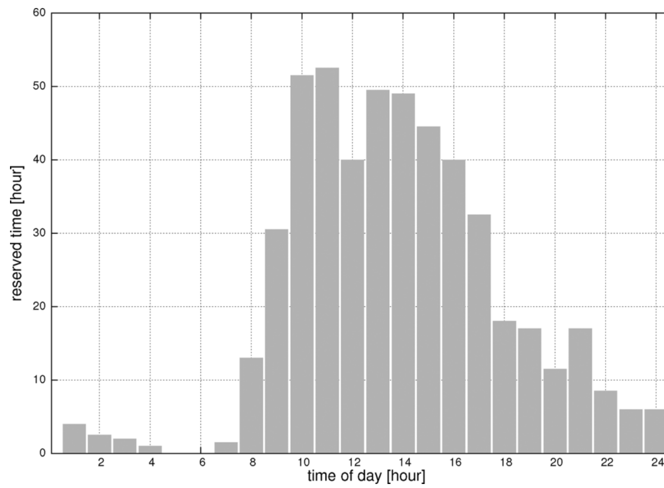

Fig. 6. Arena reservations depending on time of day.

for a comfortable and convenient management of source codes. Besides simplification and organization of a development process, SVN supports code sharing between students working in the same team and with the teacher, who can review the students' code and suggest corrections by adding notes to the source codes. This frees the teacher from tedious management of e-mail correspondence and attachments.

It is also a powerful tool for plagiarism detection and avoidance—it generates various statistics of a user's usage of the system that can reveal nonstandard behaviors. For example, it is suspicious when there is no activity in a student's repository during the whole semester, and a near-final solution appears just before a submission deadline.

Presentation of SVN statistics to students can be used to emphasize an importance of continuous work. Fig. 4 shows the activity in the SVN repository (a number of commits) of selected teams attending the *Practical Robotics* course in the winter semester 2011–2012. It is not surprising that the activity of all teams increased significantly before the deadline. Notice that teams had different habits when working with SVN.

A full operation of SyRoTek started in October 2011 at the beginning of the winter semester at CTU. Data of Arena reservations collected from that time until the middle of May 2012 show other interesting behaviors of all SyRoTek users and their usage of the system. The users are mainly CTU students of the *Practical Robotics* course (48.5% of reservations) and students preparing their bachelor/diploma theses (22.7%), but also include teachers (19.8%), SyRoTek administrators (4.5%), and individual users from other universities (4.6%).

Users made 487.5 h of Arena reservations during the whole monitored period (32 weeks), which is approximately 15.5 reserved hours per week in the average. Several users solved tasks with multiple (from two to six) robots; the number of reserved robot-hours is 1376. The system workload varied during the semester as can be seen in Fig. 5. The users reserved the Arena continuously after a few "warm-up" weeks with increased activity before the deadlines. There was no course scheduled in the summer semester (starting June 2012), and therefore it was mainly students finishing their theses who used the system. Although the total number of reservations decreased, the shape of the reservation curve is similar. A wider peak, just before the deadline, is caused by the complexity of the theses, which is higher than for a seminar work in the *Practical robotics* course.

In Fig. 6, Arena reservation depending on the time of the day is shown (only users from Central European Time Zone were considered in this case). As expected, the majority of the reservations fell during standard working hours between 8 a.m. and 6 p.m. On the other hand, about 22% of the reservations were made for other times, a few of them even at night. It is also shown that users made reservations according to their preferences and not according to availability of the teacher. Moreover, they worked not only from university laboratories, but also from their homes and dormitories, which is more comfortable for them. This is impossible with standard systems, where the assistance of a teacher is needed and robotic hardware has to be used in dedicated laboratories.

V. CONCLUSION

SyRoTek is a complex robotic e-learning system. Besides standard single-robot tasks, it allows its users to perform multirobot experiments with robots having rich sensor equipment in a high variety of scenarios. The users especially appreciate being able to develop a robotic application rapidly and run it on real hardware without having to solve problems with hardware management, robot charging, and environment preparation. They welcome, too, being able to perform experiments not only from dedicated locations in working hours, but also at the time and place that suits them best. Moreover, NetBeans IDE with the prepared code templates increases the speed of the development process (especially at the beginning of development) as it provides all modern tools for writing, debugging, and running a code and allows the users to focus on solving the task itself.

Unlike the previous mobile robotics courses taught at the CTU, where less than 20% of the students have been able to implement and test their assignments with real robots within a given deadline, all the students using the SyRoTek system entirely completed their assignments. SyRoTek also increased student interest in mobile robotics courses. While the attendance at the *Practical Robotics* course and its predecessors was almost stable (around 10 students) in the last five years, 24 students enrolled the course for the next year.

References

[1] J. Pekárová, "Using a programmable toy at preschool age: Why and how," in *Proc. Int. Conf. Simul. Model. Program. Auton. Robots*, 2008, pp. 112–121.

[2] K. Stöckelmayr, M. Tesar, and A. Hofmann, "Kindergarten children programming robots: A first attempt," in *Proc. Conf. Robot. Educ.*, 2011, pp. 185–192.

[3] A. Kazerouni, B. Shrewsbury, and C. Padgett, "Using the NXT as an educational tool in computer science classes," in *Proc. 49th Annu. ACM Southeast Regional Conf.*, 2011, pp. 67–69.

[4] P. De Cristóforis, S. Pedre, M. Nitsche, T. Fischer, F. Pessacg, and C. Di Pietro, "A behavior-based approach for educational robotics activities," *IEEE Trans. Educ.*, vol. 56, no. 1, Feb. 2013, to be published.

[5] H. Altin, M. Pedaste, and A. Aabloo, "Robotics in education: Methods of getting schools involved in robotics project in Estonia," in *Proc. Int. Conf. Simul., Model., Program. Auton. Robots, Workshop Teaching Robot., Teaching Robot.*, 2010, pp. 421–428.

[6] M. L. McLaughlin, K. K. Osborne, and N. B. Ellison, "Virtual community in a telepresence environment," in *Virtual Culture*, S. G. Jones, Ed. Newbury Park, CA: Sage, 1997, pp. 146–168.

[7] University of Bradford, Bradford, U.K., "Bradford Robotic Telescope," Jul. 27, 2010 [Online]. Available: http://www.telescope.org

[8] W. Burgard, "RHINO-Project," 2012 [Online]. Available: http://www.iai.uni-bonn.de/~rhino/tourguide

[9] University of Wollongong, Wollongong, Australia, "Robotoy," Mar. 7, 2012 [Online]. Available: http://robotoy.elec.uow.edu.au

[10] G. McKee and R. Barson, "NETROLAB: A networked laboratory for robotics education," in *Proc. IEE Colloq. Robot. Educ.*, Apr. 1995, pp. 8/1–8/3.

[11] C. Tzafestas, N. Palaiologou, and M. Alifragis, "Virtual and remote robotic laboratory: Comparative experimental evaluation," *IEEE Trans. Educ.*, vol. 49, no. 3, pp. 360–369, Aug. 2006.

[12] R. Siegwart, P. Balmer, C. Portal, C. Wannaz, R. Blank, and G. Caprari, "RobOnWeb: A setup with mobile mini-robots on the web," in *Beyond Webcams*. Cambridge, MA: MIT Press, 2002, pp. 117–135.

[13] E. Guimarães, A. Maffeis, J. Pereira, B. Russo, E. Cardozo, M. Bergerman, and M. F. Magalhaes, "REAL: A virtual laboratory for mobile robot experiments," *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 37–42, Feb. 2003.

[14] I. Dinulescu, D. Popescu, and A. Predescu, "Remote learning environment for visual based robot navigation," in *Proc. EAEEIE Annu. Conf.*, Jul. 2008, pp. 26–30.

[15] I. Mas'r, A. Bischoff, and M. Gerke, *Remote Experimentation in Distance Education for Control Engineers*. Bratislava, Slovakia: Virtual Univ., 2004.

[16] P. Petrovic and R. Balogh, "Deployment of remotely-accessible robotics laboratory," *Int. J. Online Eng.*, vol. 8, no. 2, pp. 31–35, Mar. 2012.

[17] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player/Stage project: Tools for multi-robot and distributed sensor systems," in *Proc. Int. Conf. Adv. Robot.*, 2003, pp. 317–323.

[18] "Vlab," Sep. 26, 2006 [Online]. Available: http://vlab.pjwstk.edu.pl

[19] A. Tanoto, U. Rückert, and U. Witkowski, "Teleworkbench: A teleoperated platform for experiments in multi-robotics," in *Web-Based Control and Robotics Education*, S. Tzafestas and S. G. Tzafestas, Eds. Dordrecht, The Netherlands: Springer, 2009, vol. 38, Intelligent Systems, Control and Automation: Science and Engineering, pp. 267–296.

[20] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, 2009.

[21] J. Chudoba, J. Faigl, M. Kulich, K. Košnar, T. Krajník, and L. Přeučil, "A technical solution of a robotic E-learning system in the SyRoTek project," in *Proc. Int. Conf. Comput. Supported Educ.*, 2011, pp. 412–417.

[22] M. Kulich, J. Faigl, J. Chudoba, and K. Košnar, "A visualization system for teaching intelligent mobile robotics in SyRoTek," in *Proc. Int. Conf. Simul., Model., Program. Auton. Robots, Workshop Teaching Robot., Teaching Robot.*, 2010, pp. 494–503.

[23] GStreamer, "GStreamer," Sep. 10, 2012 [Online]. Available: http://gstreamer.freedesktop.org/

[24] M. Pilato, *Version Control With Subversion*. Sebastopol, CA: O'Reilly, 2004.

Miroslav Kulich received the Ph.D. degree in artificial intelligence and biocybernetics from the Czech Technical University, Prague, Czech Republic, in 2004.

His research interests include planning for multi-robot systems, computational geometry for robotics, and education of mobile robotics.

Jan Chudoba is currently pursuing the Ph.D. degree in artificial intelligence and biocybernetics at the Czech Technical University, Prague, Czech Republic.

His research interests include robotic hardware and sensor data processing.

Karel Košnar received the Ph.D. degree in artificial intelligence and biocybernetics from the Czech Technical University, Prague, Czech Republic, in 2011.

His research interests include topological mapping, robot autonomous navigation, and long-term autonomy.

Tomáš Krajník received the Ph.D. degree in artificial intelligence and biocybernetics from the Czech Technical University, Prague, Czech Republic, in 2012.

His research interests include autonomous navigation, reasoning in mobile robotics, and aerial robots.

Jan Faigl received the Ph.D. degree in artificial intelligence and biocybernetics from the Czech Technical University, Prague, Czech Republic, in 2010.

His research interests include multigoal planning, motion planning for high-dimensional systems, and autonomous robotic systems.

Libor Přeučil received the Ph.D. degree in technical cybernetics from the Czech Technical University, Prague, Czech Republic, in 1993.

His research focuses on robot sensing, mapping, and navigation.