RILE – Robotic Interactive Learning Environment

Lakshmi Prayaga
Department of Applied Sciences
University of West Florida
Pensacola, FL 32514
lprayaga@uwf.edu

Chandra Prayaga
Department of Physics
University of West Florida
Pensacola, FL 32514
cprayaga@uwf.edu

Aaron Wade
Department of Physics
University of West Florida
Pensacola, FL 32514
awade@uwf.edu

Niranjan Suri
Institute for Human and
Machine Cognition
Pensacola, FL 32502
nsuri@ihmc.us

Alex Whiteside
University of West Florida
Pensacola, FL 32514
ajw22@students.uwf.edu

Jeff Hawthorne
University of West Florida
Pensacola, FL 32514
jwh36@students.uwf.edu

Miroslav Kulich
Czech Technical University
Prague, Czech Republic
kulich@labe.felk.cvut.cz

Libor Přeučil
Czech Technical University
Prague, Czech Republic
preucil@labe.felk.cvut.cz

***Abstract*—We report the design and alpha-testing of a Robotic Interactive Learning Environment (RILE) to teach introductory one-dimensional kinematics to middle school students. The environment is centered around robots which are controlled via WI-FI, and are equipped with sonar sensors to provide distance vs. time data. A student can remote-login via an application and perform experiments on kinematics on the robots and understand the usually difficult concepts of displacement, velocity, and acceleration. The system was initially tested in a middle school for multiple batches of students at the 8th grade level. Initial reactions show that the students were engaged, interested, and excited. In particular, the excitement of working with real robots kept the students alert to pitfalls in the understanding of kinematics, as shown by their responses to qualitative questions on interpretation of graphical data.**

I. INTRODUCTION

The use of Robotics offers an exciting and engaging environment for teaching STEM (Science, Technology, Engineering and Mathematics) concepts. From the early work with LEGO kits and other robots in the late 1980s with Dr. Seymour Paperts software TC Logo, the use and/or programming of robots by elementary and high-school students has been shown to improve their understanding of STEM topics such as gears, friction, light, heat, movement, velocity, acceleration, slopes, force, etc., together with the mathematical skills required to comprehend these concepts (Weisteider and Brown) [1]. In addition to the skills developed by manipulating the robots to perform the desired actions, robotics also improves the students ability to collaborate with other students and work in teams (WRRF 2003, CAST STEM Institute) [2], a much desired skill. At the same time, it is clear that providing the equipment necessary for students to control and/or program their own robots does not scale to the large number of schools and to the large number of students within the schools, due to the high cost of the equipment needed per student. We present a preliminary version of RILE for middle school physics that facilitates tele-robotics as a mechanism to provide middle school students with the ability to remotely manipulate and control real robots and their environment through a Web-based interface, drastically reducing the cost per student but at the same time providing students the engagement and excitement of working with real robots, including the difficulties of manipulating robots in the real world instead of digital avatars. We built on 1) existing applications of tele-robotics technology in education and on 2) innovative programming paradigms being developed by the researchers from leading institutions.

II. EXISTING WORK

There is considerable research on robots in education. Major, Kyriacou and Brereton (2011) [11] conducted a comprehensive study that reported on the effectiveness of using robots including simulated robots in computer science education. Their case study reviewed 34 papers of which 23 papers reported use of physical robots and 7 papers reported use of simulated robots to teach computer science concepts both at the high school and university levels. Six of the seven papers reported that simulated environments were effective in computer science instruction. The authors Becker (2001) [5] , Borge (2004) [7], Buck (2001) [8], Enderle (2008) [9], and Ladd and Harcourt (2005) [10], reported that a simulated robotics environment was very effective in introductory computer science instruction at the university level.

Our project is novel in that it extends the notion of tele-robotics to K-12 (Kindergarten 12th grade) education and allows direct manipulation of the physical robots located in a real laboratory environment. Our objective is to better motivate students to learn STEM principles and in particular, concepts in physics. Tele-robotics is the notion of being able to control robots from a distance usually via a network (the Internet in our case) to accomplish specific tasks. Example tele robotics project include NASAs Rover, Mercury project, and the Puma Paint project [Stein, (2003)] [13]. Tele-robotics projects in higher education were proposed by A. Bicch, A. Caiti, L. Pallottino, and G. Tonietti (2005) [6], Riyanto Bambang (2007), and by Kulich, Kosnar, Chudoba, Fiser, Preucil (2012) [3] . We use the existing literature as a basis for our experiment. Our hypothesis is that RILE will provide a highly motivating and engaging learning environment for students to achieve the desired learning outcomes. An example learning outcome identified for the experiment on kinematics is described in this paper. We believe that RILE can be leveraged for achieving student learning outcomes in many STEM disciplines including computer science, IT, physics and math.

III. RILE Architechture

RILE complex architecture extends from a basic table-like base to high level user interfaces, incorporating many components in between, allowing a computationally efficient, yet functional system capable of performing physics experiments in a controlled manner. Our architecture is based on the current client-server paradigm and is split physically into three components: Robots, Servers, and External Connections. Each of these entities is both logically and physically separated, relying on network connections both over a local network and the Internet to connect and function. Not only is this necessary to allow portability and expansion, but also essential due to the various local computer and microprocessor architectures within the lab, and the client computers that the students will be using to connect. For this reason, at the software level, we use various languages including C, C++, and Java. The figure below shows a simplified visualization of the environment.

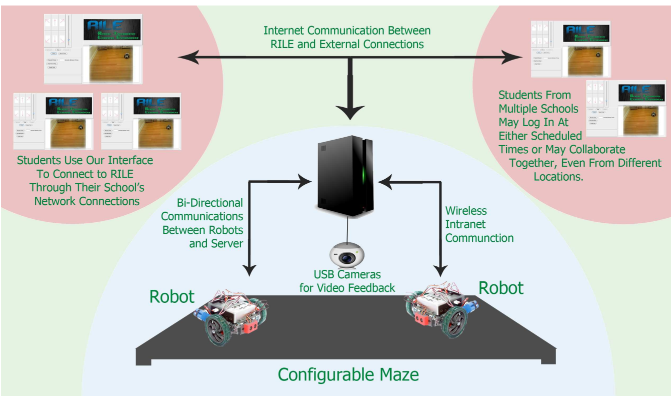

Fig. 1. Visualization of RILE's Architecture

Within the RILE system, we can separate subsystems by location: local interface (Client) and remote system (Server).

A. The Remote System

RILE's remote system includes the experimentation environment, complete with robots, a table-like field, video capabilities and a central server. Its centralization is a key part of the RILE paradigm, allowing one remote system to service numerous local systems independent of location.

At the environmental level, RILE is configured with a simple 9' (2.7 m) square table with raised edges to provide an closed environment for the robots. This field is shown in Figure 2.

Fig. 2. RILE Table

Mounted above the table are 2 high definition USB cameras, which monitor the field.

At the robot level, we have a dynamic number of robots with varying local architectures, each connected to a network via Wi-Fi. At this level, all commands and protocols are specific to each micro-processor and each robot is easily modified though hardware additions. These robots run software specific to their microprocessor which connect to the RILE API on the server and allow local autonomous routines independent of the larger system.

Within the environment, several monitoring programs run to provide information about the system to the server. An example of this is RILETracking, a video streaming service with object detection. This process reads in the multiple camera streams, combines them into one image, and searches that image for robot markers. If a marker is located, the service reports the robot ID, as well as the 4 corner locations of the marker to the server, designating the robot's exact position and orientation. A video stream is also created for external connections to monitor the video feed.

The server program is central to RILE and coordinates all connections and activities. Once devices connect, the server controls all lines of communication and allows various devices, programs and users to communicate with one another in a standardized way. It preforms safety functionality, including keeping robots form colliding; maintenance, including reporting issues to administrators; and security, preventing threats to its users and itself.

While these software modules handle communications and robot control, RILE also provides a flexible system for monitoring robot sensors, reporting information and presenting it in a graphical format. The flexibility resides in the idea of a module tray, which lies between a user interface and its connected robot. This tray allows numerous modules to be loaded, increasing functionality. Modules currently include a dataset grapher, a position monitor, a velocity monitor, and a data distributer. These modules provide near limitless extendibility for the RILE system as they can be created by users for specific functionality.

At the top of the remote system lies a connection secretary. This program handles all physical connections relating to the external system, both internal and external. These include robot sockets, video streams, user connections and more. This secretary acts as a gate-keeper to the system, allowing one central connection interface.

B. The Local System

The local system within RILE is contained within a single graphical user interface (GUI). This GUI allows students, teachers and administrators to easily connect to the remote RILE system, preform experiments, watch video lectures and more in a user-friendly manner. This system must be installed on the computer the user is sitting at, but requires nothing other than an internet connection to fully utilize RILE.

Upon startup, the local system connects to the designated remote system and authenticates the user via a username and password. From there, the program forks into either student, teacher or administrator mode based on the user's credentials.

Students are forwarded directly to textbook mode, the primary module of the local system. While similar to a typical textbook, RILE's textbook interface (shown in Figure 3) organizes material in a logical, hierarchical system, allowing multiple classes, grade levels and disciplines to use the same textbook, decreasing cost and student acclimation between classes. This novel redesign of the textbook promotes the teaching of topics as a logical piece of a puzzle, instead of a linear chapter in traditional textbooks. We believe that this design will not only improve understanding about the connections between lessons, but also allow independent use of the system without spending time sorting through an index.

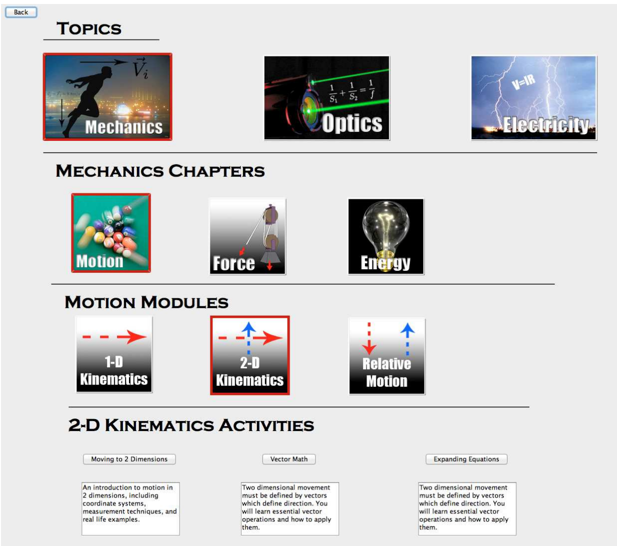

Fig. 3. Textbook Interface

Within this interface, as students select the appropriate levels, lower menus populate with appropriate sub-materials, ending with activities which can be selected.

Once an activity is selected, students are presented with a simple pre-test to estimate current subject knowledge. They are then shown a video lecture of the topic, supported with figures, graphs, and data, providing a stimulating initial presentation. Once finished with the lecture, students are shown a pre-lab document describing the experiment in detail and connecting the experiment to the topic they were just presented. Students may then move on to the experiment itself, the key feature of the RILE system.

Upon entering the experiment panel, students are given the option to preform the experiment or watch a live experiment in progress. This feature allows an entire class to benefit from experimentation, even with limited robot hardware. Viewers receive all video and raw data from the experiment, and are given their own options for graphing and reviewing data, separate from the controller. However, if the preform option is selected, the student is also given access to robot navigation and sensor control panels, customized for the specific experiment, allowing controlled experimentation. A helper block helps walk students through experimentation, connecting live robot data to previously discusses topics. At any point, students can return to the pre-lab or lesson video to revisit information. An image of this interface is shown in Figure 4.

Once the actual experimentation is complete, students continue to the data review pane to analyze their data. Within this pane , students can access all of their recorded data sets and graph them to visualize their results. These graphs and datasets can be exported for lab reports. Finally, the lesson ends with a conclusion and a post test.

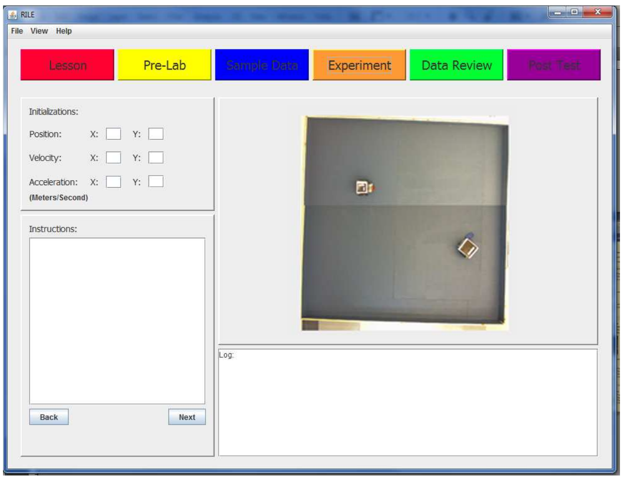

Fig. 4. Experiment Interface

If the user is a teacher, they are given the option to use the RILE system as a student, or to access the teacher controls. These controls allow the creation and manipulation of classes and student accounts within the class. These accounts keep track of student experiments and results as well as test scores for teacher review. Administrators have full access to the server and can create student, teacher and administrator accounts as well as control server behaviors.

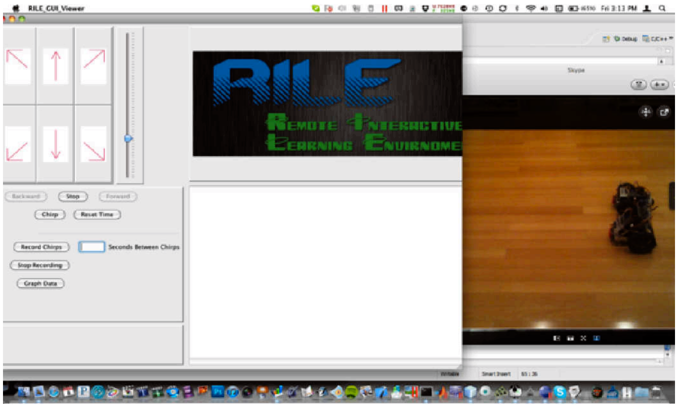

Fig. 5. Alpha Version User Interface

IV. Alpha Testing

Preliminary testing of RILE was performed in a middle school in Florida to evaluate the technology, the impact of the learning environment, and student interest. The experiment was conducted in five 8th grade sections (age group 14 years) with a total of 85 students. Two sections were taught in a traditional manner using chalk and board and the other three sections were introduced to the same concepts using the RILE system.

The concepts taught were part of one-dimensional kinematics. We concentrated on four concepts: distance, displacement, speed, and velocity. This test of RILE was for just one class period, so no attempt was made to teach acceleration, though the system itself includes acceleration as one of the concepts that can be taught. We chose kinematics for two reasons: (1) In any standard course in physics, kinematics is the first topic taught. (2) The difficulties in teaching kinematics concepts are well documented in the literature. Physics education research shows that student understanding of concepts in mechanics (kinematics and dynamics) suffers in a traditional classroom lecture environment, where students learn 20% of the presented material in a traditional lecture course in introductory physics [McDermott, (1993)] [12]. Considerable research exists on student misconceptions in mechanics, including the lack of ability to interpret kinematic graphs. Some of the recurring misconceptions are [Dykstra (2004)] [4]:

Inability to distinguish between distance and displacement
Inability to distinguish between speed and velocity
Inability to recognize the existence of acceleration when only the direction changes (uniform circular motion)
Inability to associate slopes of graphs of position vs time with direction of velocity
Identifying zero velocity with zero acceleration
Associating force with motion, even uniform motion.

Targeted Student Learning Outcomes:

Student demonstrates the difference between distance and displacement
Student distinguishes between speed and velocity
Student demonstrates connections between graphical representation of motion and the physical motion itself.

A. Trial Methodology

A pretest on these concepts was first administered to all the classes involved. In the robots-based class (the treatment group), there was an initial (10 min) introduction to the concepts, and the students then congregated around the experimental area. In the experiment, a robot was allowed to move in a straight line under commands sent by a student via the RILE interface. The GUI designed for RILE was sufficiently easy and allowed the students to quickly understand how to control the robots from the computer. Less than 5 min was spent on actual acclimatization to the system which is shown in figure 5. This GUI has since been updated to improve usability and promote structured experimentation.

The robots were equipped with a sonar sensor mounted on the rear of the robot that measured the displacement from a fixed wall. This was considered the position of the robot, and the robot was programmed to acquire this data at fixed intervals of time, typically once every second. The client control station was also programmed to produce an on-screen graph of the position versus time. The data and the graph were ported to four other Observation computers in the lab, so a group of 3 - 4 students were stationed with each computer. All the links between computers and robots were through WiFi. The layout for the experimental setup is shown below in figure 6.

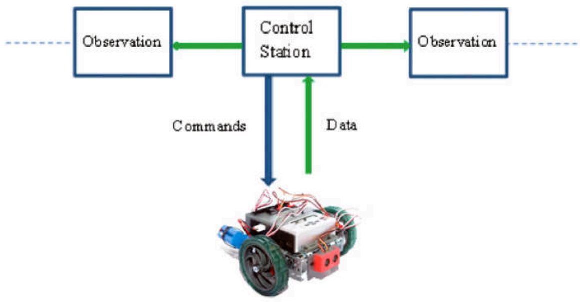

Fig. 6. Alpha Testing Setup

The students experimented with the robot by commanding it to move with constant velocity, and then inspected the graph. The experiment was repeated, with differing velocities, including changes in velocity and velocities in the opposite direction, and the students noted the qualitative features of each graph, such as the fact that a higher speed results in a steeper graph, the slope of the graph changes sign with the direction of the velocity, etc. The data points generated from the sonar sensors are shown in Figure 7 and the graphs generated from the software are shown in Figure 8. During this the course of this experiment it was clear that students began predicting the resulting graphs with more and more certainty.

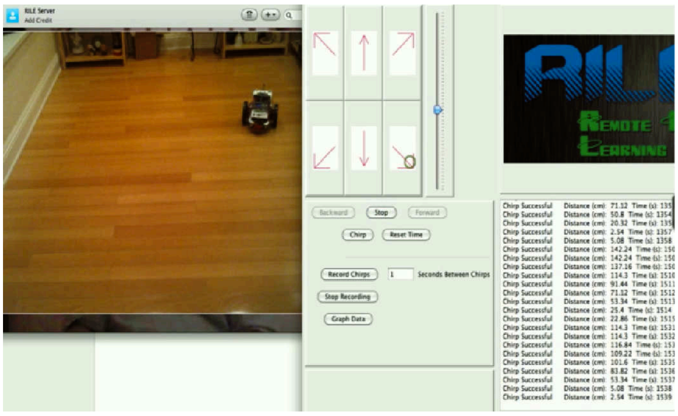

Fig. 7. Text Based Sensor Feedback

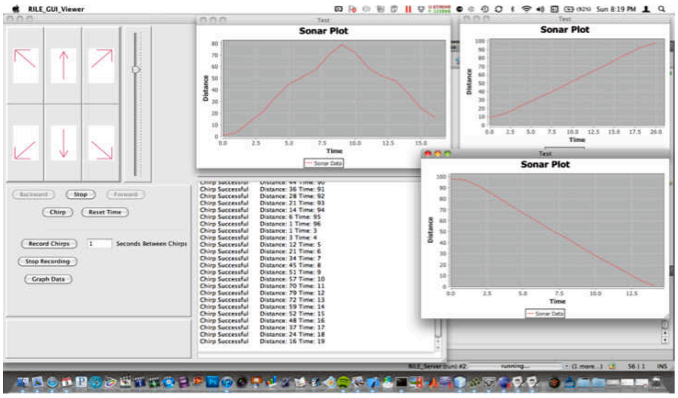

Fig. 8. Graph Based Sensor Feedback

Within the control group, the same concepts were covered in a traditional method using the blackboard. The theme of robots moving under student control was used as the background for introducing the concepts. The students were introduced to the ideas of position, displacement, velocity, distance and speed. A few exercises were done under supervision, where position vs. time data was given in a table and the student was asked to calculate the displacement and velocity during specific intervals of time. The students were also shown qualitative graphs of position vs. time, and were asked questions such as, In this region, is the robot moving faster or slower?

B. Results

The alpha tests were evaluated by observing the amount of student participation compared to the traditional lecture, the learning curve of student manipulation of the robots, the students ability to orally interpret the physical meaning of graphs, and how well the students were able to make the robots behave according to a position versus time graph.

The active learning environment when utilizing the robots resulted in the class being highly engaged when compared to the traditional lecture where the instructor had to break from lecture every 5-7 minutes to obtain the students' attention. This time spent controlling the class far exceeded the 5 minutes necessary to train the class in how to use the client GUI. This resulted in more available class time to discuss the course content. In addition, the short length of time necessary to train the students in how to use the GUI is a testament to the ease of use of the interface.

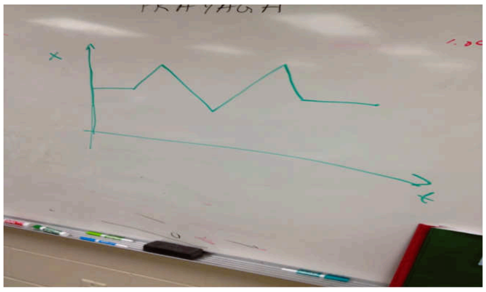

Fig. 9. Graph Based Sensor Feedback

When the instructor drew several graphs of position vs. time on the board (Figure 9) and asked the students to evaluate the graph and make the robots move with the same relative motion as the graph. Most students were able to drive the robot and compare the system generated graphs, which were pretty close in achieving the required result. Qualitatively the students were able to:

1) Associate the horizontal axis to the physical notion of time. When presented with a graph that had the time flowing backwards (see Figure 10 below), several students in each group were able to recognize that time cannot flow backwards even before attempting to control the robot.

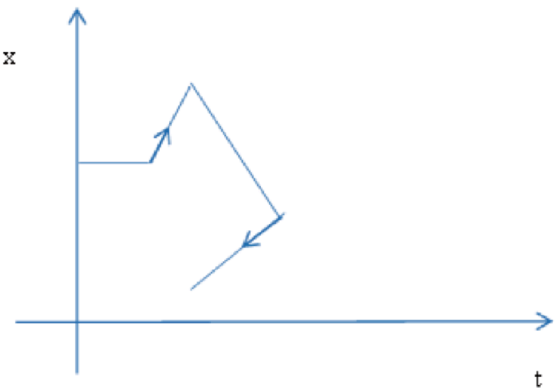

Fig. 10. Graph Illustrating Backwards Time Travel

2) Understand that position can be a positive or a negative value and that it is relative to an associated origin.
3) Some students, not the majority, could say that both backwards and forwards velocity could have the same speed and were able to associate the steepness of the graph to higher speed.
4) The instructor drew an arbitrary graph of position versus time on the board, with different sections of the graph showing higher or lower speeds, and some sections with the robot not moving (velocity =0) The students were

then asked to make the robot move according to the graph.

By comparing the graphs drawn on the board to the graphs generated by the client GUI when manipulating the robots, the number of errors quickly decreased and the students were able to reproduce the original graph immediately.

The response from the students from all the five batches of the class was uniformly encouraging, and the teachers were enthusiastic about using the system for future classes and also on other topics in physics.

V. Conclusion

A remotely located Robotic Interactive Learning Environment has been designed for purposes of teaching one-dimensional kinematics to middle school students. Initial testing shows that the system does provide an engaging environment for the student to learn the traditionally difficult concepts of kinematics, the foundation of physics. The system allows remote login and robot control through an internet connected Wi-Fi network. Students recieve live data from the robots over the internet, and also visually interact with the robots via a video link. This system provides a central experimentation environment capable of supporting numerous classes, making it a cost effective alternative to classroom robot sets. Within the RILE system, new technologies and innovations in educational methods provide a complete educational experience through a logically organized textbook based interface; while in-system quizzes, instructions and structured experimentation allow teachers to focus on teaching material and answering questions instead of lab setup and materials distribution. Overall, it seems to be a viable implementation of robotics in education.

VI. Future Work

The system is in the initial stages of development. Further developments involve systems which can be used to teach more complicated concepts in physics, including high-school level physics - such as 2-dimensional kinematics, force concepts, energy and power considerations, concepts in electricity and magnetism, and optics. Many interfaces are being updated to provide a more user friendly and dynamic experience while keeping the system logical and easy to use for both students and administrators. External applications are also being developed to provide more accurate environmental information such as robot position, lighting and connection statuses. This knowledge will lead to more autonomous behaviors, including autonomous charging and pre-lab setup.

In order to test these improvements, we are reaching out to other schools within the region to preform beta testing as well as promote the importance of robotics education.

References

[1] Weisteider & Brown (2008) (Virginia Westheider, Patrick Brown, Enhancing K-12 STEM Education with Engineering, Proceedings of the 2008 American Society for Engineering Education, Annual Conference, Pittsburgh, PA. .

[2] Western Region Robotics Forum, http://wrrf.org/wrrf/backgrounder/.

[3] Kulich, M. - Košnar, K. - Chudoba, J. - Fišer, O. - Přeučil, L.: User's Access to the Robotic E-Learning System SyRoTek. In Proceedings of the 4th International Conference on Computer Supported Education [CD-ROM]. Porto: SciTePress – Science and Technology Publications, 2012, vol. 1, p. 206-211. ISBN 978-989-8565-06-8. .

[4] Dykstra, D. I. (2004). Why Teach Kinematics ? An Examination of the Teaching of Kinematics and Force – Part I. Physics Retrived from: http://web1.boisestate.edu/physics/dykstra/.

[5] Byron Weber Becker. Teaching cs1 with karel the robot in java. *SIGCSE Bull.*, 33(1):50–54, February 2001.

[6] A. Bicchi, A. Caiti, L. Pallottino, and G. Tonietti. Online robotic experiments for tele-education at the university of pisa. *Journal of Robotic Systems*, 22(4):217–230, 2005.

[7] Richard Borge, Annita Fjuk, and Arne-Kristian Groven. Using karel j collaboratively to facilitate object-oriented learning. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, ICALT '04, pages 580–584, Washington, DC, USA, 2004. IEEE Computer Society.

[8] Duane Buck and David J. Stucki. Jkarelrobot: a case study in supporting levels of cognitive development in the computer science curriculum. In *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, SIGCSE '01, pages 16–20, New York, NY, USA, 2001. ACM.

[9] Stefan Enderle. Grape – graphical robot programming for beginners. In Achim Gottscheber, Stefan Enderle, and David Obdrzalek, editors, *Research and Education in Robotics — EUROBOT 2008*, volume 33 of *Communications in Computer and Information Science*, pages 180–192. Springer Berlin Heidelberg, 2009.

[10] Brian Ladd and Ed Harcourt. Student competitions and bots in an introductory programming course. *J. Comput. Sci. Coll.*, 20(5):274–284, May 2005.

[11] L. Major, T. Kyriacou, and O.P. Brereton. Systematic literature review: teaching novices programming using robots. *IET Seminar Digests*, 2011(1):21–30, 2011.

[12] Lillian C. McDermott. Guest comment: How we teach and how students learn—a mismatch? *American Journal of Physics*, 61(4):295–298, 1993.

[13] Matthew R. Stein. The pumapaint project. *Autonomous Robots*, 15:255–265, 2003. 10.1023/A:1026216520523.