e-learning, robots simulation, virtual laboratorie, e-experiments

Paweł GAWŁOWSKI, Maciej PERLIŃSKI,
Sławomir WIĘCH, Lech WILCZYŃSKI[*]

VIRTUAL ROBOTICS LABORATORY FOR NON-CONVENTIONAL E-LEARNING SYSTEMS SOLUTION

There is a lot of Learning Management Systems which solve some distant learning problems or which are good addition to stationary lessons. They support wide scope of standards like SCORM [1] or IMS. They are commonly used to publish learning content and to perform tests. They have different types of remedies for „no direct contact" problem, like forums, video/chats etc. but often it's not enough for some types of classes.

1. INTRODUCTION

The problem of distant learning absorbs many teachers for years. There is virtually no direct contact with a student and possibility to reach technical laboratories.

There is a lot of Learning Management Systems which solve some distant learning problems or which are good addition to stationary lessons. They support wide scope of standards like SCORM [1] or IMS. They are commonly used to publish learning content and to perform tests. They have different types of remedies for „no direct contact" problem, like forums, video/chats etc. but often it's not enough for some types of classes.

In this paper we would like to present two non conventional distant learning systems for teaching. First one being a system for distant programming teaching called "Code Lords", and the second one is virtual robotics laboratory.

Code Lords has been launched in 2003 in Polish-Japanese Institute of Information Technology and since then it is used as part of Internet Studies in PJIIT. In assumption Code Lords allow users to automatically test and verify software solutions written for programming and algorithmic classes

The goal of the Virtual Robotics Laboratory is to allow internet students and universities not having robotics laboratory on their own to experience real life problems associated with robotics. Simulated experiences such as Stage, Saphira [8] or Aria will never recreate random scenarios that occur while dealing with live electronics. We, on the other hand, are able to provide a real environment for our robots and associated sensory equipment.

[*] The Polish-Japan Computer Technologies Academy, Warsaw, Poland

2. THE PROBLEM SHOWCASE

Teaching programming and writing optimal algorithms is a difficult task in academic environment. Basically there are two approaches to this problem. First one called “start small”, where students begin writing small programs from scratch. The other is “code provided”, where students are given complex piece of code to modify and extend [12]. Both of these approaches assume direct contact with teacher.

The problem with teaching programming arises more complicated when it comes to distant learning [14]. There is no possibility to allow direct correction of errors made by student. In normal conditions teacher can be next to a student and give feedback and constantly check out problem solving progress, can help and evaluate solutions. Distant learning forces the student to use only schemes, books and tutorials.

Programming is not only "by the book" subject. One to become a programmer has not only read several books and tutorials, but also need to write a lot of source code. Programming looks very complicated at a first sight. Every programming language is different. Each programming languages need different teaching approach than other (C++ and Pascal) [10]. Every language has different semantics and syntax, which programmer has to know in order to write a program. A program should also do something interesting and not trivial and this causes more problems to arise.

It often causes this sense of frustration, because of knowledge, which student must possess and which seems to be enormous issue. Attendance of a tool such as compiler can in some cases become more complicated and it is only small part of knowledge, which must be known by a fluent programmer.

Using teachers' knowledge is more difficult in distant-learning conditions. In traditional programming teaching, student can assign direct question to a teacher. For example a person, who wrote a program which doesn’t compile because of unknown errors can ask teacher a question about behaviour of the compiler , what causes bad compilation, and is not convinced to waste time for personal research.

It decreases frustrations in the beginning of learning, where majority of answers returned by the compiler (as well as interpreter in most languages which can be interpreted) is incomprehensible for students. This is the example of a simplest problem which can be fast and easily solved by teacher. There are much more complicated problems such as: e.g. algorithmic tasks. In their cases, finding a solution or verifying it's correctness by teacher can turn out to be harder to execute without capability of appealing for educator's knowledge. Usually during distance-learning there's no contact between other persons participating in the same course. During traditional occupancy of programming studies, considerable knowledge is adopted by students from other participants of these studies.

A long time of expectation for a capability to verify the solution is next problem in programming teaching. Classic approach for algorithmic and teaching programming relies on assigning task to be solved in class, and unless they will be executed, terminating them at home. Task solutions are verified by the teacher during classes. Unfortunately, because of the time-consuming verification, it usually happens on the next classes for most of the students. Frequently, student does not have capability of earlier contact with master. Because of this he must wait for a week to check if his solutions are correct or ask questions about them. In case of erroneous solution student loses another week, because his new solutions can be verified only on next classes.

Wasting time during classes is big problem in such model of programming course. Teacher checks each student solution step by step. Analyzing correctness of anybody source code is, even for expert programmers, it can be an irritating problem which occupies time. Starting up and reviewing each program for correct answers essential. Master must explain the problem exactly, in case of discovering errors. Particularly it is painful in the beginning of the course, when students make many mistakes and solutions must be corrected many times. In time, when instructor deals with one student, rest of the group usually waits for their turn and looses valuable time, which could be take advantage in more productive manner.

We have to deal with another problem beyond time-lost. Lack of engagement of student and discouragement on studied subject is a result of necessity of long expectation. It significantly lowers the capacity of teaching. If there would be a possibility to evaluate immediately solved task, student could become involved to solve task completely until achieving the proper solution.

Teachers have many problems with proper checking programs written by students. It's not only the result of difficulty or task complexity which requires analyzing and testing unknown source code. It can be caused by some prosaic causes like big amount of evaluated solutions.

Automating of this solution process would be invaluable assistance for each programming teacher. It would enable considerable abbreviation of evaluation time, as well as it would solve problem with limited access to a teacher. Considerable simplification of science and teaching process would be a result.

3. THE SOLUTION IDEA

Presented system relies on automatic verification of algorithms. Our solution is based on comparing correctness of program operation results with its correct pattern. We do not analyze essential correctness of source code step by step like it is done in C-Tutor [7], but we only check if it returns correct results for input data which was prepared earlier. Due to selecting proper input data, which will control program behaviour in particular cases, essential correctness is can be forced preparing input data. While compiling, syntax is certainly checked. Such approach has defects for sure, but it enables student to get reflexive information very fast. Simultaneously teacher is not aggravated by necessity of checking solutions manually.

In case of system for aiding traditional course, student weekly participates in classes. Certain algorithmic problem is taught and task for homework is presented and discussed. Student has a week to solve a task using the system and present his solution during next classes. Teacher can look in received solutions before the classes. Due to this he can be better prepared for it. He's getting knowledge about which student had solved task correctly and who had problems. That's why, he will save a lot of time by helping only those who need that or if the same mistake was committed by many students, he can tell them all about it.

In addition he has access to statistics so he can see how many times students were sending solutions and according to this he can evaluate progress in course. It is known that beginners in programming makes many mistakes, in order to swindle fluency in programming after some time and solving task in smaller amount of approach. Besides, there is no necessity for secondary program tests, because system has already done it, and it generated additionally proper report with results.

Test carried by system are more precise because of proper and carefully prepared input which cause smaller possibility of mistake.

3.1. SYSTEM OVERVIEW

It is possible to solve a problem using many approaches described in former point. In this point we would like to present our solution idea. Our system was initially created for aiding standard method of instructing programming. However, it can be used in case of complete lack of "teacher". It doesn't matter where student is, because our system relies on Internet, and is accessible using standard web browser.

3.2. USERS

We have accepted several assumptions during realization of our system. Firstly, we have assumed, that person who will be using our system is an advanced computer user and will not have problems with using of internet browser, browsing WebPages, as well as sending files to our system via WebPages. Secondly, we have assumed, that person who will be using our system have at least basic knowledge of program writing. By basic knowledge we understand ability to write simple program, its compilation, start-up and submitting it. Indeed it is possible to place in our system some fast course of programming basics; however it wasn't our purpose during its creation. We have put a premium on giving fast answer to the learning person and tell if the written program is correct in relation to defined assumptions in task, but not to learn complete basics like how to use the compiler. Certainly, it is possible to place in our system some description, how to use the compiler with proper examples, but it will not be in any case interactive learning.

3.3. USE CASE

After logging to our system, user is given access to several task sets. Task set is connected with concerning subject for example “algorithms and data structures”. Task set consist of several tasks, which must be executed by the user. In next point, appearance of task has been exactly described. Each task can have attributed limits of applications, equal quantitative (maximum amount of submission solution of given task) as well as time (date for when it is possible to report solutions of given tasks).User can report solution for each task, which contains one program source code file solving assigned problem. User obtains reflexive information from the system after verification of correctness of sent solution. He or she can observe his results, as well as compare them with other course user's results. If there were mistakes in the solution, information about types of errors would be send to user. Except standard system functionality like reporting solutions of task, user has access to several other functions which should help with programming learning. Discussion forum enabling exchanging observation and experiences between participant of course, as well as between user and teachers is one of them. Users can use knowledge base existing in system in form of articles and files about programmatic subjects.

3.4. TASK PRESENTATION

Each task consists of descriptive part with problem included. It is presented as amusing story in order not to tire student by boring and tame description of algorithmic problem. Problem is described in accessible manner, picturing situation from real life. Exact specification of input and output program data is placed under each task. It is important to serve them in unequivocal and unawake doubt manner. Exemplary input and output data are also served. Student can test his program before sending, due to it during preparation of exemplary data, we must pay attention, are they are short enough to be analyzed manually, on paper, which will enable better problem understanding, but simultaneously, are they complicated enough that they will enable program testing.

3.5. TESTING SUBMITTED SOURCE CODE

After submission of solution by student, system automatically identifies programming language, which submitted source code is written. Next it is compiled using appropriate compiler. In case when program will not compile correctly, process of testing is interrupted and student is informed about this event. After that for each test set program is started up. Student is credited by point, in case when program will lend proper solutions. Student is informed in opposite case of erroneous solution Program can cause error of execution time during operation which student is properly informed about.

3.6. GETTING FEEDBACK

As it has been already said in former point, student is informed of problems, which have occurred during applied program verification.

Most trivial error is compilation error. It means that program includes syntactical errors, by which it can't be compiled and started up. Certainly this type of error should happen unusually seldom, because students should test programs before submission. If the program has compiled correctly, it is started up several times for every task set and reflexive information is presented to student for every of this start-up. There are 5 possible answers.

Correct passage of test is first and most desirable. It means that program has correctly solved task and it has made it within determinate time limit framework.

Giving of erroneous answer is second. It means that surely program is poorly written and student must correct it. It can mean that student has not forecasted some edge condition and according to this program doesn't work properly for maliciously matched input data's, or simply the whole algorithm is incorrectly written and it requires deepest rebuilding.

Exceeding of time limit can be third possible answer. It means, that program hasn't answered in indicated time limit. It comes out of having algorithm incorrect implemented, and by that it act far too slowly. This type of answer says nothing about correctness of results. It is possible that program has landed good answer, but it has not been optimally written. Possibility which also exist is that program landed erroneous result, or it has not landed it at all, because it is erroneously written and it has runt to infinite loop. In such case student must review his code regarding to a possibility of accelerating it by writing in more effective manner some fragment of program.

Time execution error is next system possible answer. It means that program has compiled correctly, but it has executed in correct operation during operation and it action has been ended. Dividing by zero is proper example of this situation.

Outreaching limit of available memory is last possible error. Each task has amount of memory allocated which student can reserve. Outreaching of this limit means that student used wrong algorithm or he has written it incorrectly by using bad data structure.

3.7. OUR EXPERIENCE USING SYSTEM

During 2 years of work we have notices several advantages of using our system.

First of all time spent during classes is used much better. Teacher knows before classes all the problems which his or her students faced. He or she knows which problems should be mentioned or discussed in more details. Tutor also knows which students need one-in-one explanation.

Secondly students are more involved in learning. They don’t have to wait whole week for verification of their solutions. They can write a program whenever they want and evaluate program in the very first moment after writing it. If the program is wrong, they know it immediately. They can correct the program when they are still fresh with the subject matter.

Lastly the teachers have less work to do. They do not have to waste vast amount of time analyzing and testing students’ programs. This by the way isn’t very precise. They have more time to discuss problems that have arisen and alternative solutions.

In a nutshell the system we have created is not only great enhancement to standard classes, but also a stand alone system for distant learning.

4. VIRTUAL ROBOTICS LABORATORY

Virtual Robotics Laboratory (VLAB) has started as a part of master thesis of Maciej Perlinski, Michal Wojcik, Marcin Bobowicz, Pawel Osmialowski, Tomasz Bartlomiejski, Piotr Dutt and Marek Jacenko, students of Robotics and Multi-agent Systems at PJWSTK. From the early stages major goal of this project was to share Our Robotics Laboratory hardware to bigger community. After consultations with chief director of Robotics and Multi-agent Systems Department at PJWSTK prof. Lech Polkowski we started research on possible functionalities of Virtual Robotics Laboratory.

After final deployment of the project we'll let other Academic entities without required hardware and software equipment to provide lectures from the field of Robotics and Multi-agent Systems in the meaning of theoretical and practical assignments using VLAB.

During practical assignments students can solve real world problems working with embodied systems surrounded by novel and dynamic environment and not only software simulators.

Practical assignments using embodied systems are giving different results concerning influence of all real world factors. Virtual Robotics Laboratory is hybrid architecture. All agents within the laboratory are sharing the same API and can access other agents readings from their sensors/devices. Whole laboratory is controlled by major server which on one side acts like another agent and have access to all agent's devices on the other side it serves data for all clients from outside of laboratory. In that manner we'll provide all data and sensor readings to VLAB client application installed on the client side machine presented in Fig.1.

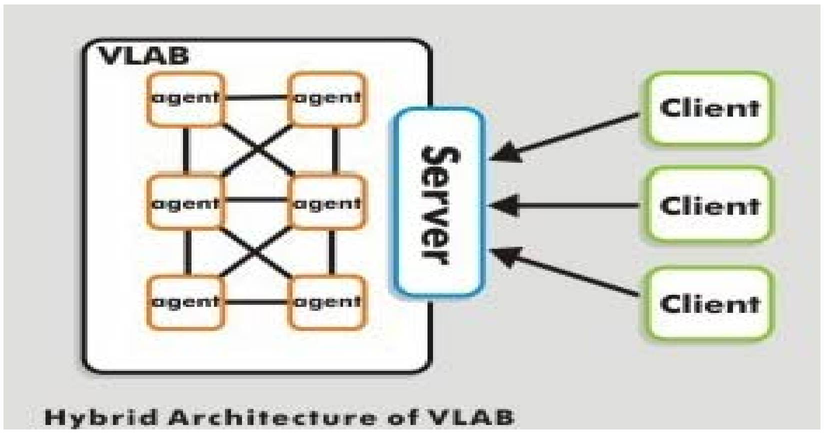

Fig. 1. The hybrid architecture

Practical assignments with Multi-agent system have certain difficulty level due to hardware differences between the agents, different operating systems they require to perform their tasks and different protocols of exchanging data. In order to simplify student's work with such an environment and enhance productivity we have decided to use unified API for all agents in the VLAB. In this way we are providing network functions and access methods for sensors/devices readings.

Unified API allows users to create reusable parts of the code which can be used on the other agent or in case of often repeatable problems for example low level behaviors of the agent or optimization of tasks.

Virtual Robotics Laboratory can provide ideal case study for the research on remotely controlled autonomous systems. For example when creating the robot to perform tasks on Mars we have to take into account that the distance between two planets is so great, that it can take up to 40 minutes to deliver a message from Earth to Mars and receive the answer. Agent sent to Mars must be able to handle emergencies on their own until new orders form the control station arrive [2].

Another advantage of using Virtual Robotics Laboratory is amount of accessible hardware without a care about maintenance. VLAB provides wide variety of sensors/agents such as high resolution cameras, lasers, sonars and mobile robots. Working with such a platform gives opportunity for development of advance systems for example based on sensory fusion.

4.1. PHYSICAL PROPERTIES OF VIRTUAL ROBOTICS LABORATORY

Virtual Robotics Laboratory has two sectors. First sector called "Arena" has dimensions of 448x386 cm from each side separate from the rest of laboratory with the borders. Each border has different color and is built from the wooden boards. The inside surface of the borders are covered with the sponge for the safety reasons of agents. In this area we can work with two tank robots and universal robot Pioneer 3-DX.

Over area of "Arena" there are two rails with two mobile wifi cameras and between the rails there is another high resolution, wide lensed camera giving the image of whole "Arena".

Second sector called "Small-Arena" is much smaller. Within the borders of "Small-Arena" we can work with two hemmison type robots. Above this area we placed one static camera to give the image of whole "Small-Arena". Borders of "Small-Arena" are not covered with the sponge because hemmisons are covered with the protection shield (Fig.2).

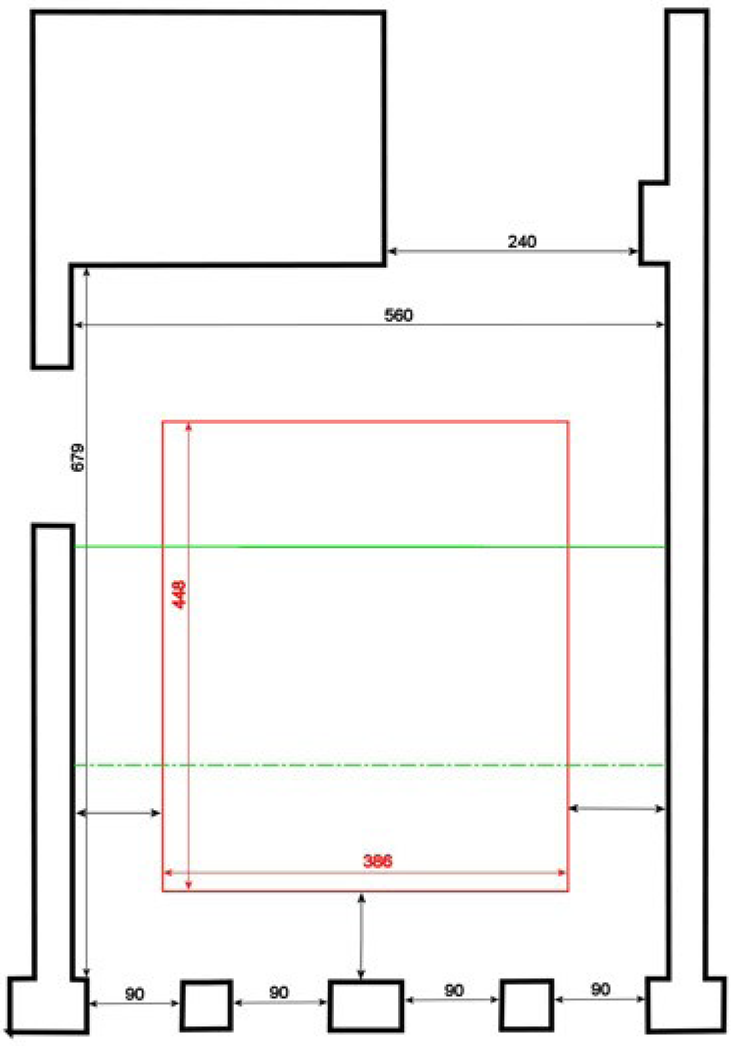

Fig. 2. for the map of Virtual Robotics Laboratory

4.2. HARDWARE DEVICES, SENSORS AND ROBOTS IN VIRTUAL ROBOTICS LABORATORY

Hardware devices:
- 3 Wireless Cameras
- 1 High resolution, wide lensed camera
- 5 GPS sensors

Robots inside of “Small-Arena”:
- 2 hemisson robots

Robots inside of “Arena”:
- Pioneer 3-DX with sensors:
  - Sonars
  - pan-tilt-zoom camera
  - SICK laser measurement system
  - GPS
  - color tag on top of the robot
- Two tank robots with sensors :
  - Sonars
  - camera

5. SUMMARY

E-Learning is relatively new field of knowledge which is rapidly growing. Not all aspects of this field can be unified and further research is required. There are many standard solutions in this field which can be successfully implemented in most of e-learning cases. On the other hand there are some fields that require not trivial solutions, dedicated to learning given subjects. One of such problems is learning algorithmic, where it is necessary to be able to perform fast verification of student’s programs. Such problematic subjects are also electronics and robotics, where using only tutoring materials is insufficient and access to laboratory with proper equipment is necessary. We hope that our systems partially solve these problems.

Code Lords system is working for 2 years now helping stationary and remote students to learn algorithmic. As it was said earlier, since its introduction our students are more involved in programming. Teachers also benefits from this system. They have less work to do, because of automating of program verification. They can also prepare better for classes and have more time in classroom to teach new things rather checking students’ assignments.

VLab is meant to be a major breakthrough in distant robotics learning. Learning robotics is impossible without access to laboratory, which is available only at some universities. Maintenance of robotics laboratory required qualified staff and funds which causes this domain of computer science slowly growing and sometimes is hard or impossible to expose for wider community. VLAB is meant to promote this domain of knowledge providing necessary environment for other Academic entities and remote students. It will allow our internet students and students from partner universities to take the full advantage of learning robotics.

BIBLIOGRAPHY:

[1] ADL Technical Team, SCORM 2004 Documentation 2nd Edition, Advanced Distributed Learning(2004)

[2] BACKES P. G., TSO K. S., NORRIS J. S., THARP G. K., SLOSTAD J. T., BONITZ R. G., ALI K. S., Internet-Based Operations for the Mars Polar Lander Mission,IEEE International Conference on Robotics and Automation(2000)

[3] BERQIA A., DIOP A., HARMS J., A Virtual Laboratory for Practical exercises, CUI, Université de Genève, International Conference on Engineering Education August 18–21, 2002, Manchester, U.K.

[4] BURGARD W., CREMERS A. B., SCHULZ D., Architecture of the "Robotic Tele Lab", Dept. of Computer Science III, University of Bonn

[5] COX R., BRNA P., Supporting the use of external representations in problem solving: The need for flexible learning environments, Journal of Artificial Intelligence in Education, 6(2), 239--302 (1995)

[6] FERNANDEZ R. O., KINGUTI E. C., RAMIREZ-FERNANDEZ F. J., A virtual Laboratory to Perform Electronic Experiements by Internet, Polytechnic School of University of São Paulo, International Conference on Engineering Education August 18--21, 2002, Manchester, U.K.

[7] HAHN S. H. , SONG J. S., TAK K. Y., KIM J. H., An Intelligent Tutoring System for Introductory C language Course, Department of Computer Science, Korean Advanced Institute od Science and Technology

[8] KONOLIGE K., MYERS K., The saphira architecture for autonomous mobile robots, http://www.ai.sri.com/konolige/saphira/, 1996

[9] LEE Y., LEE H., MA W., DU D., A Systematic Approach to Design the Network-Based Learning Environment for Home and Office, Distributed Multimedia Research Center, Department of Computer Science

[10] MCIVER L., CONWAY D., Seven Deadly Sins of Introductory Programming Language Design, In Proceedings, 1996 Conference on Software Engineering: Education and Practice. IEEE Computing Society Press, Los Alamitos, CA, USA 1996

[11] MORRISON D., E-Learning Strategies, John Wiley & Sons Ltd., West Sussex 2003

[12] NICHOLSON A. E., FRASER K. M., Methodologies for teaching new programming languages: A case study teaching LISP., Monash University, Victoria

[13] SCHWARZ J., POLZE A., WEHNER K., SHA L., Remote Lab: A Reliable Tele-Laboratory Environment, International Conference on Internet Computing (2000)

[14] STEINEMANN M.-A., and BRAUN T., Remote versus Traditional Learning in a Computer Networks Laboratory, Communications and Computer Networks (CCN 2002)