Journal of International Scientific Publication:
Materials, Methods & Technologies, Volume 3

HOME AND REMOTE LABORATORIES – DEVELOPMENT OF WEB-BASED PRACTICAL WORKS FOR HIGHER AND VOCATIONAL EDUCATION.

Oleg Shvets[1], Vello Kukk[2], Martin Jaanus[2], Kadri Umbleja [2]

[1] Virumaa College of Tallinn University of Technology 30322 Jarvekula tee 75 KOHTLA-JARVE, Estonia

[2] Tallinn University of Technology Faculty of Information Technology: Department of Computer Control, Ehitajate tee 5, 19086 Tallinn, Estonia

Abstract

The development of integrated environment providing tools for performing laboratory experiments both in distant and home lab forms. Home labs are based on the so called HomeLabKits consisting of all necessary equipment for experiments. Remote labs use physical equipment located in educational institutions operated over the Internet. In both cases, learning is controlled by server-based software which does not only post tasks to user but also logs all activities, selects problems on the basis of current student state and suggests learning materials. The development process has included two projects: the first was devoted to the design of a set of HomeLabKits inside Tallinn University of Technology and the second one for three higher education institutions and two vocational schools. In this paper, we describe main principles of developed environment and experience of its application. One of conditions of the flexible and qualitative educational process is integration of laboratory works and opportunities of their performance. For this reason it is possible to consider remote laboratories as a new tool in hands of the modern student which will help him/her to improve the quality of knowledge mastering. The presented project is one more step on a way of this integration.

Key words: *e-learning project, lab sharing, distance education*

1. INTRODUCTION

The remote laboratories are, as a rule, complex structure consisting of the equipment and the software. Complexity of this equipment and the software directly depends on target group for which development of the remote laboratories is intended.

Last years significant interest is devoted to distantly operated robots. An important reason is that development of new technologies causes application of robots in a wide range of problems in which remote presence of the operator-expert is required. Remote experiments appear as one of basic technologies in international space station, in application of robots at ecological and technogenic accidents, for research and neutralization of suspicious subjects in mass congestion of people, use of remote-operated objects for military applications, use of robots for assembly of space designs. The new perspective application is the use of robots operated over the Internet, for remote training to robotics [1, 5].

For learning needs, a remote laboratory is important in at least two cases: 1) if equipment is very expensive and therefore it is to be shared by many students, 2) if operational zone is dangerous or cannot be accessed directly by other reasons.

Remote laboratories have a firm place in modern educational process and frequently are good means for economy of time and finance. At the same time, application of the remote training always causes a

problem of reception, so-called, a feedback from the student to the teacher and assessing problem. That is why we applied the same technology as used for home labs: control of learning process is strictly personal and includes some class-works (tests) that must be performed in-site. Results of those must confirm learner's previous work and have the most significant weight in final grading.

The world practice of creation of the remote laboratories is presented on sites

Remote laboratories are considered by numerous authors. Here we refer to [2] where usage of ABB robot used in Labview environment is used over years. In [3] complex project combinening virtual and distant labs are described. Many projects are presented in [4].

In Estonia, three projects have been carried out: 1) design of HomeLabKits (2005-2006), 2) inter-university project for development of home and distant labs (2006-2008), 3) project INTERSTUDY follows (http://www.mehhatroonika.ee) [5].

For increase of efficiency of application of the remote laboratories the main problem is development of the software by means of which educational process will be handled. We offer some aspects and technologies of development of such working environment.

2. FUNDAMENTALS OF HOME AND REMOTE LABORATORIES

2.1. Home Laboratory

Starting from 2007 all labs are based on 'HomeLabKits' – small boxes that include all necessary equipment for experiments. Figure1 shows older verion of kits.



Figure 1. Older verion of kits.



Figure 2. New version of kits.

New version are shown on Figure2. Management of labs is organized as follows. Contents of a kit is individual but not fixed. Components can be moved from one kit into another if needed and this is represented in database which contains all the history of kits. So, even labs performed using previous contents of the kit can be correctly tracked.

Totally 626 lab experiments are set up in database. They may vary by experiment type but also by components used. Experiments are simple, usually from 2-7 measurements and/or calculations. Submitted results are processed and depending on previous student's state the new one is calculated.

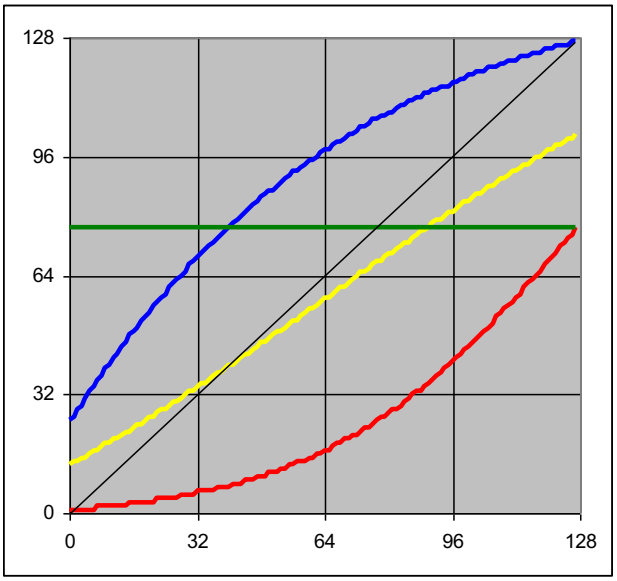


Figure 3. Transition functions. Input - horizontal axis, output - vertical axis. Blue line - correct, red line - wrong, yellow line – neutral.

Lab tasks are grouped into 12 modules that are attached to different courses (4 courses are based on hat set) and every student has his/her state in every module. States are represented by inetger number from the region 0..127. As a result of any experiment (and also other tests) the state is changed as shown in Figure4 where only three cases are shown. Results can be anywhere between wrong and correct and it depends on type of task - sometimes only true and false can be applied as sometimes any real value between those extremes can be obtained (particularily, in experiments). So far, no connection between modules does not exist meaning that result is influencing only that module the task has come from.

2.2. Remote access

The remote user accesses a specialized Web-server via the Internet/Intranet, sets on the site necessary for remote control parameters which are later transferred to the LAB server (Figure 4). These data are used by the LAB server to perform full or partial modeling of the robot's movements depending on a set mode.

It has to be mentioned that as the speed and the quality of connection between the remote user and the Web-server in general is difficult to predict, in order to perform remote control it is necessary to make

a scenario of an experiment by filling a form; and upon the completion of automatic control of defining set parameters within specified limits it will be transmitted to the following computers. Quite naturally, the remote user seems to have fewer possibilities for setting and mode selection than an operator who works at the control panel.

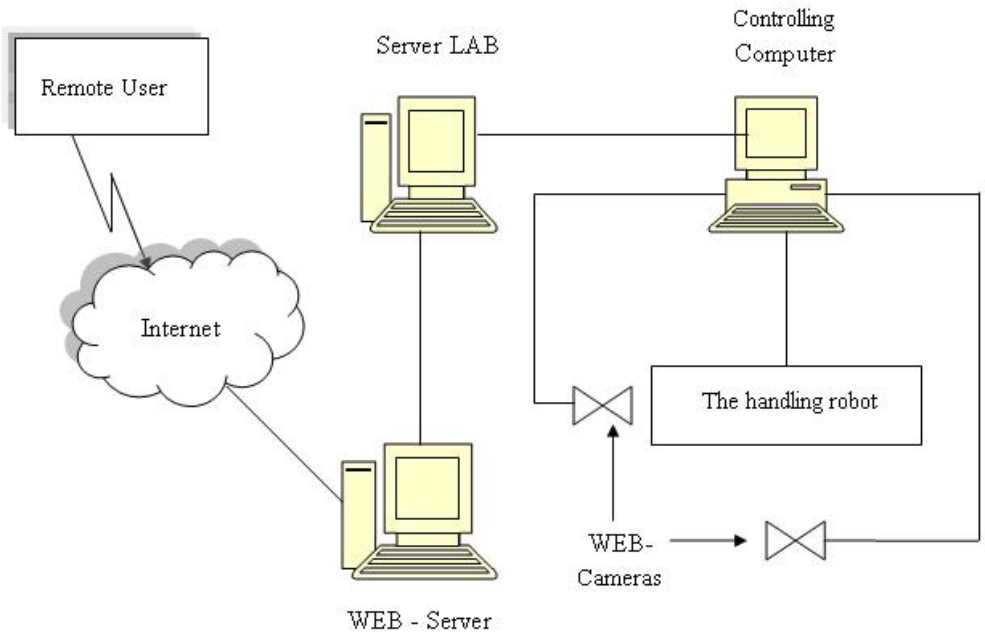


Figure 4. Scheme of remote access.

Equipment: robotic hand, web server controlling the robot, 2 cameras. Up to 10 users can be connected to cameras but usually only one has access – the one whose programme is just running.

Two modes are implemented: 1) direct connection with instant execution of commands sent by the user, 2) time-sharing when users' procedures are scheduled and executed from a queue which is implemented as FIFO. Priorities can be applied to procedures taking long time but it is rather theoretical as usual procedures take time less than 1 minute.

All commands sent to the robot are verified to avoid catastrophic movements and procedures violating restrictions are not allowed to appear in the tasks queue.

2.3. Net control software

Several models of computer software development are considered here. The first scheme is more or less standard in terms of its realization.

To effect the exchange of data between functionally modeling panel and the remote computer it is necessary to develop software written in high-level object data model programming language. The scheme is shown in Figure 5. Software is a complex of programmes which includes CGI-application

(query.exe), executed directly on the Web-server, and console application (take.exe), responsible for data receive from the LAB server. The PHP based Site Core is responsible for the generation of HTML-pages and provides with the help of cookies-files authorised access to the results of the conducted experiments. The data base is meant for users' registration data storage, queries to conduct experiment and their results.

The function of remote access channel is provided, first of all, by CGI-application query.exe which activates two traffics in server's core memory. Via one traffic it sends data to the LAB server, then waits for the confirmation of the data receive and the initiation of a remote experiment. One of CGI-application's functions is the transfer of parameters to the LAB server. The LAB-server receives the control parameters (modeling) according to a set format. The second traffic provides interconnection with the data base – it registers parameters of remote experiment (table QUERY). Storage of remote experiment's parameters is necessary for a more detailed analysis of received results.

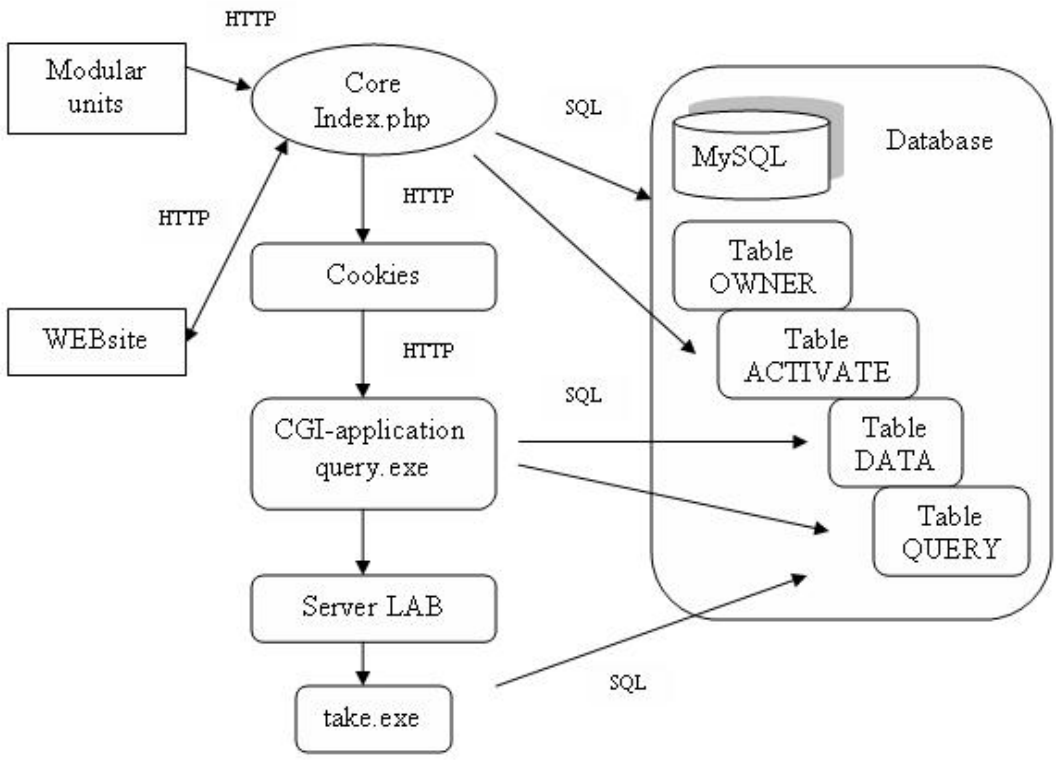


Figure 5. Standard realization of computer software on the basis of the PHP core.

With the help of console application take.exe the results of the conducted remote experiment are entered to the data base, which provides access to them later by means of PHP-technology. By addressing the server, this console application inquires the server at regular intervals. After remote experiment has ended, it creates a typed data file and registers it with the server's data base (table DATA).

Upon completion of a remote control session the user will get an opportunity either to download the file via the net or to construct a manipulation robot's path motion or to use the mode of browsing for the results of remote control.

The second scheme reflects in detail the functional part of required computer software. Figure 6 shows the scheme.

While analyzing the role of the components, let us consider some of them in detail.

Validator(s) – sets the correction of the input at the lower level, i.e. input formats, value of parameters, units.

Selector – offers to the user to choose a task. This part of computer software has to be personalized.

DB – data base which contains information about users, users' budgets, and the information about the laboratory (graphic data, video, etc).

Interpreter (compiler) – acts as a compiler, i.e. translates what user has sent into a set system of commands. Scheduler – is a process administrator (pick up a task from a queue, task's initialization, etc).

Filter – system of filter commands; eliminates errors to protect devices.

Processing – preparation and processing of tasks for sending to the user and their further upgrading.

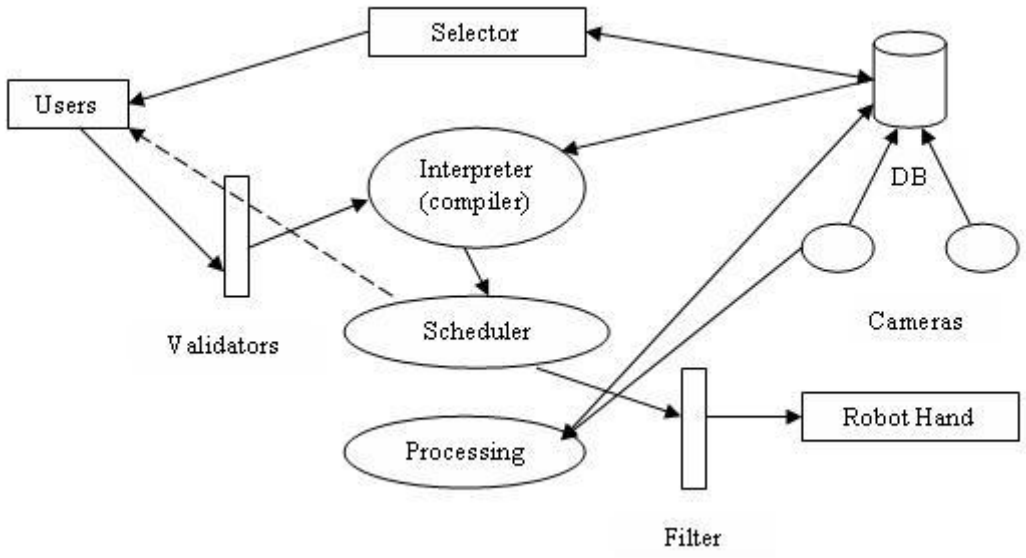


Figure 6. The scheme of computer software realization developed during the project.

Computer software is realized on the Microsoft ASP.NET platform by C# language, and data base uses the MS SQL Server. While creating the software the idea was to make it universal, with possibly a user friendly interface which will be easy for communication without any complicated programming

to update and change software of existing labs. The second aim was to process data from browser to server.

Description of labs is better to take from the paper Student Forgetting Model. I can add something new too.

2.4. Tasks for robotic hand.

Tasks are composed in different levels and modules allowing controlling different knowledge components. Here is a list of basic task types:

1. Identification of robot's servos (direct connection, sliders control)
2. Determination of coordinates (photos of desk with objects, both Cartesian and polar coordinates)
3. Determination of state using Denavit‐Hartenberg matrix (given robot's parameters)
4. Pick-up tasks using macro commands (moveto, grip), combined with determination of coordinates
5. Moving objects using macro commands
6. Programming of given movements (straight line, letters etc, speed may be given)

System tasks for initiation and completing actions:

1. Parking hand (setting robotic hand to safe state and switching off power)
2. Parking objects (collecting objects into their specified positions in special part of desk)
3. Initiation of scene (moving objects from park to prescribed positions on the desk and setting hand to its initial position for current task)

System tasks for control unit:

1. Keeping state information about hand
2. Keeping state information about objects
3. Managing tasks' queue
4. Validation of procedures before moving into queue
5. Processing results of experiments
6. Managing the use of cameras

Note. Managing states of a learner and selecting tasks from database are implemented on the 'learning servers'. Only user id and task id are transferred to robot server. Information about cameras, errors, and experiment results are transferred to learning server which is the only one communication directly with a user.

3. SOFTWARE DESIGNING

3.1. Overview of lab's software design.

Software consists of two main modules (Figure 7):

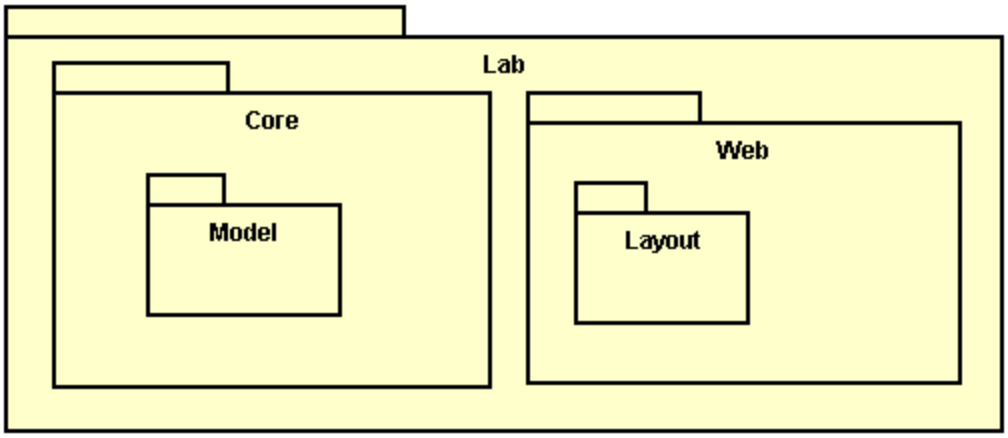


Figure 7.General structure of lab's software.

Core – contains classes of lab's software basic functions. The sub-module Model includes classes which describe the data model of lab's software. Web – realizes classes for users' groups from Web – environment.

3.2. Data Model

Software uses data units which can be divided into three classes:

- Lab's level constants and parameters of kit's constants
- input data units (results of measurements, calculations, comments, formulas)
- derived data units (by taking into account dynamically set constants, input data and other derived from the data base of data units). Each and every unit has to be defined by a corresponding descriptor. Descriptor defines the type of a data unit (real or complex number), scalar value and vector. Simple arithmetic formulas and text comments are also supported. Input and derived data units can have corresponding validators with the help of which it is possible to control data correspondence to various criteria. The number of validators with regard to data units is not limited. To control results there is a notification about an error or a warning. Within the effective connection IScriptSource receives from a validator its functionally represented code (JavaScript) and other names of data units set by a script. Additionally, it is possible to set various validators which can control, for example, a number of combinations of data units. Validators can determine if they are working with a student or an instructor or in both working modes. The data descriptor of the above presented model DerivedDataDescriptor also realizes IScriptSource for the purposes of the assessment of the algorithm in use. The scheme of DerivedDataDescriptor descriptor's work is shown in Figure8.

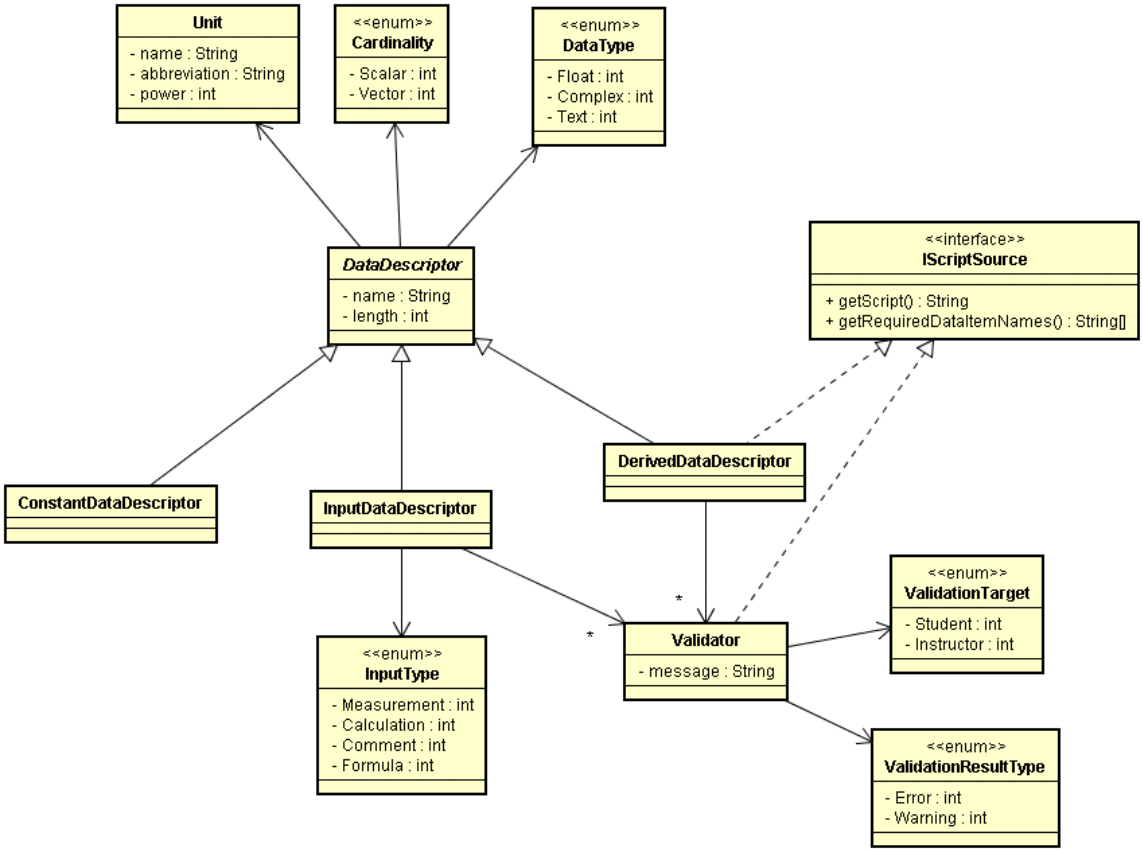


Figure 8. The scheme of DerivedDataDescriptor descriptor's work

In more detail data units are represented by DataItem class with three sub-classes (Figure9). Each data unit is connected with its descriptor, and its value makes the Quantity value.

There are also all numeric values of complex numbers in it (except InputDataItem, which keeps separate all input text values).

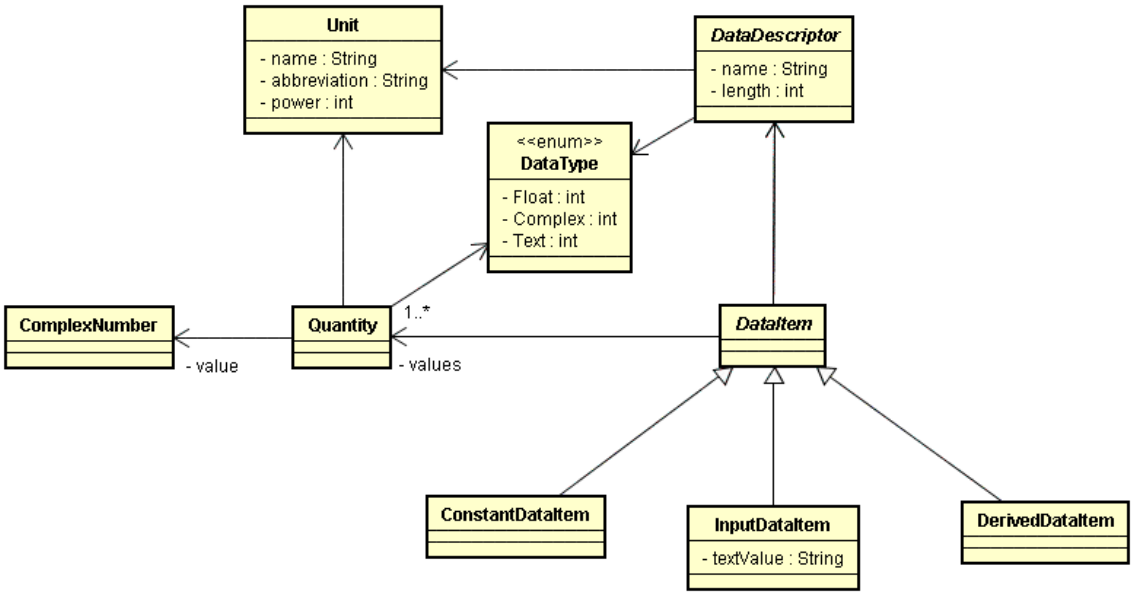


Figure 9. Realization of DataItem class.

Lab data model is generated from dynamic access to the lab with the use of XML-file which contains metadata of the described data model.

3.3. Classes of basic function.

The work of basic function classes can be represented by means of the following scheme (Figure11).

● User – a user of lab's software

● Course – a course during which lab work is performed

● Lab – a particular lab work. This work is linked with data descriptors which set lab constants with regard to lab's input and derived data and lab's validators.

● Kit – a kit

● LabSession – medium class which defines active status of a lab and the required functionality of all data unit

● ScriptBuilder – connects scripts with corresponding data units

● ScriptEvaluator – an auxiliary class with the help of which validators are set and data units are filled up with scripts.

● LabDao – contains data from data base for reading and description of the required functionality .

3.4. Connection of users.

Similarly to the metadata model which is used in XML, each laboratory is connected with its user's model on the basis of which HTML is dynamically generated in the browser. The model of users' connection determines laboratory forms, its structures and rules of its use (Figure10).

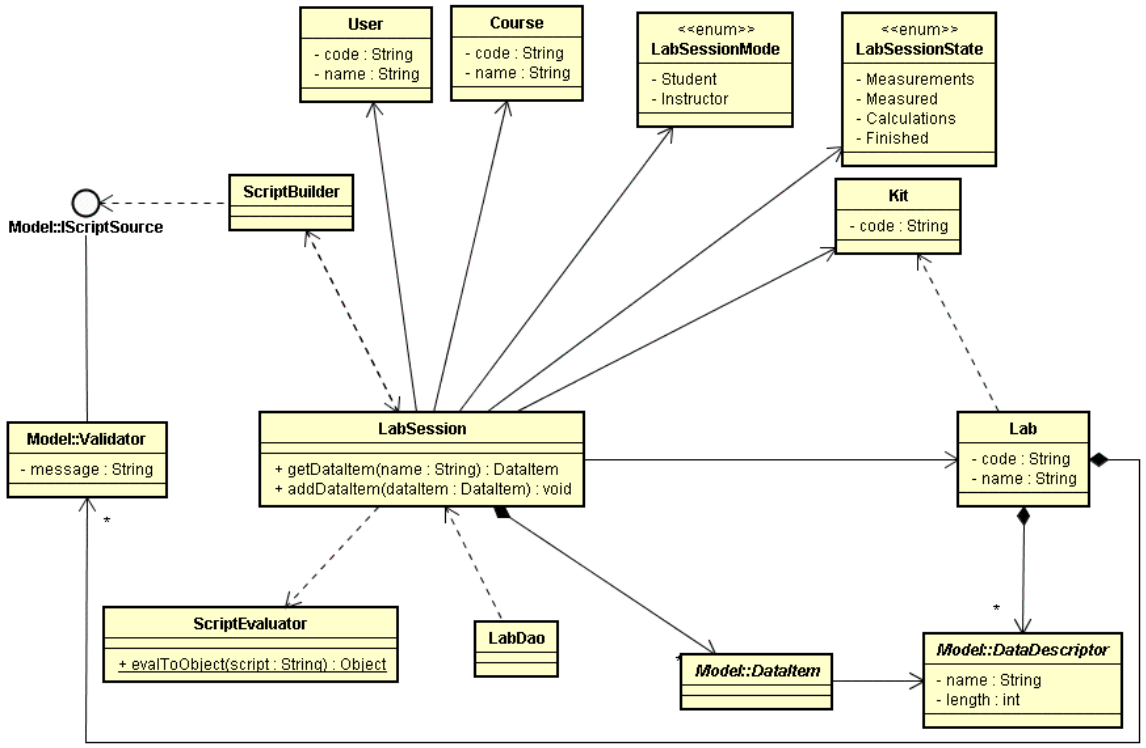


Figure 10.Classes of basic function.

JavaScript is used to present schemes in the laboratory and it enables the browser to show customers all necessary data. The customer-server connection is based on AJAX technology (Asynchronous JavaScript and XML). It allows to enter/modify data while saving it and to control it on the server; it decreases the need for laboratory forms' update and the risk of data errors; it also makes possible to provide user with immediate feedback.

3.5. Classes related with the model of users' connection.

The following scheme shows the model of classes related with users (Figure11).

● LabApplication – LabDao and LabFactory are used to save examples independently from ASP.NET sessions.

● LabFactory – is used to create new laboratory sessions. It reads laboratory related data from data base with the help of LabDao and creates with the help of XmlSerializer an example of LayoutModel class for the model of users' connection. As this action takes time, the data transfer can be performed by access to the memory.

● LayoutModel – is a description of users' connection in XML form.

● LabForm – is a basic screen form of the laboratory. It unites all other programme components and coordinates its actions.

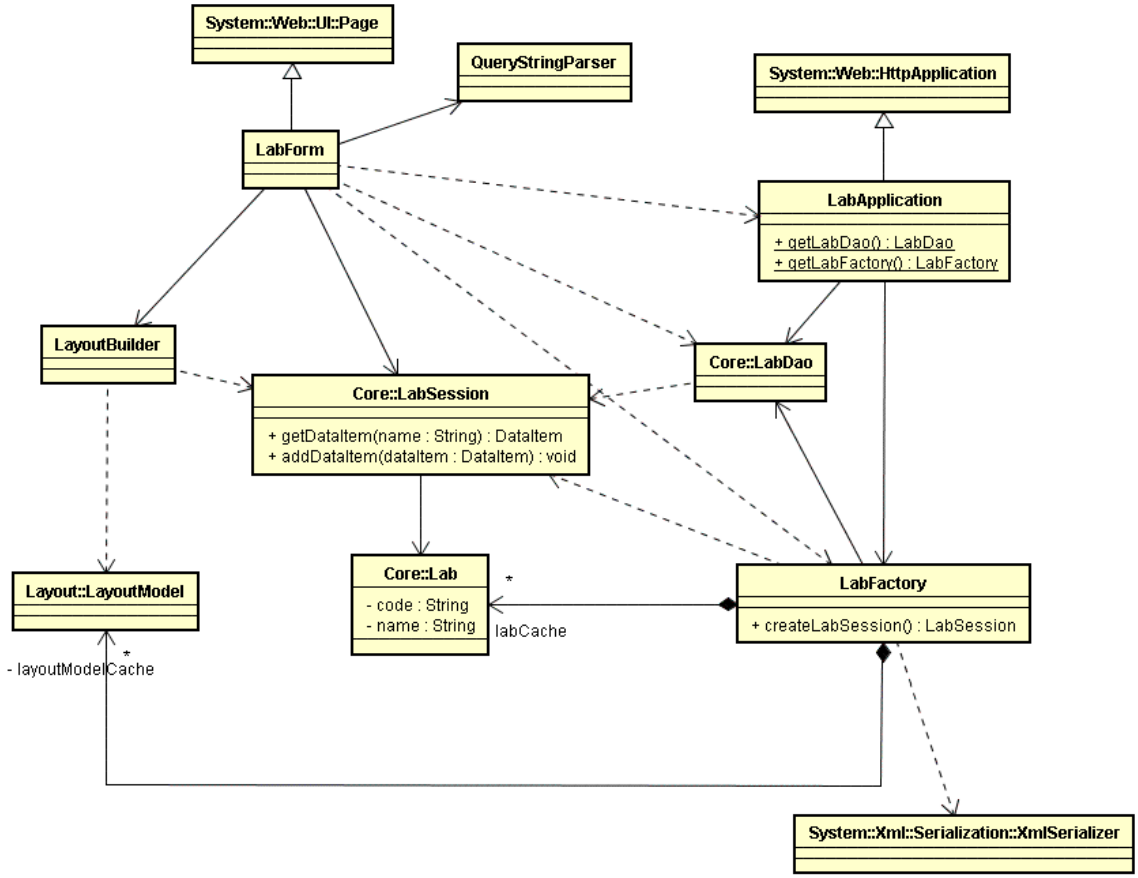


Figure 11. The model of classes related with users.

● LayoutBuilder – is a class which realizes actions of dynamic generation, it contains the description of users' connections in LayoutModel and keeps current data in LabSession.

● QueryStringParser – is an auxiliary class of HTTP parameters interrelated with QueryString.

4. RESULTS AND DISCUSSIONS

HomeLabKits have been tested for several years in different configurations and using different controls. Main conclusion is that students use them both in school laboratory as well at home. In most cases, a student first visits school lab, generally about 3 to 6 weeks and then is ready to take the kit to home.

Important result is that due to the control that demands positive results, a student has to work much more with experiments than in classical labs. As environment provides fully individual learning track the students can use just as much attempts as they need and this number may be large (dozens of

experiments). Students appreciate instant reaction from the system which matches expectations of 21st century's learners.

All experiments have been logged and analysis of that data shows that it is possible to extract much more information for student model than it was expected initially. In fact, results of an experiment allow to insert changes not only to the item where the task came from but also to other components of student model. This is the main topic for further development activities.

Combining robotic hand with the same environment used for experiments and other tests have shown this simple device can provide large number of different tasks concerning very different areas of knowledge, including math, physics, mechanics, programming etc. Developed software enables to use the device in time sharing mode making it available to large number of users as no preregistration is needed.

In both cases, homelabs and distant labs, the most important learning effect is high motivation – environment itself with instant reaction has created enthusiasm, the most important for successful learning.

5. CONCLUSIOS

Internet based remote laboratories are of great importance for the work of contemporary educational institutions. Realization of such laboratories is a complex problem of how to integrate equipment, software and learning materials. Students can conduct experiments and acquire skills while physically being present in various places in the world by using equipment resources of one university. This approach reconsiders radically the traditional approach to teaching; it increases the quality of learning and develops communicative thinking.

The presented project is one more step forward on this difficult but interesting way.

REFERENCES

1. Kukk, V.; Jaanus, M. (2008). Student Forgetting Model: Practical Experience. In: EAEEIE 2008 proceedings CD-ROM: 19th EAEEIE Annual Conference, Tallinn, Estonia, June 29 - July 2,2008; 112 - 117.
2. http://telerobot.mech.uwa.edu.au/Telerobot/
3. http://www.sira.diei.unipg.it/principal-fields-of-research/robotics-2.html
4. http://www.distancelab.org/projects/
5. Reiner Dudziak, Carsten Köhn, Raivo Sell. Integrated systems and design, 2008; 15-40.