A Novel Software Architecture for Project-oriented E-learning with LEARN2CONTROL

Marten Völker, Thomas Tometzki, Christian Blichmann, Ernesto Elias-Nieland, and Sebastian Engell
Process Control Laboratory, Universität Dortmund, 44221 Dortmund, Germany
{m.voelker | t.tometzki | c.blichmann | e.elias-nieland | s.engell}@bci.uni-dortmund.de

Abstract – **An important skill of a successful engineer is the ability to structure complex projects into smaller subtasks and to solve each subtask by applying state-of-the-art theories and methods, taking into account the interdependencies between the subtasks as well as the overall goals and requirements. LEARN2CONTROL is a web-based learning environment that can be used to acquire basic knowledge in control engineering by independent and interactive work on design projects that are offered in a uniform setting. The didactic concept aims at teaching the dependencies and the interactions between various methods for modelling, analysis and control system design.**

In this paper, we present a recent redesign of LEARN2CONTROL by which a major disadvantage of the previous version has been overcome: the need for local installations of complex (licensed) programs on the client side. In contrast to the previous version, the present architecture is completely server-based. The use of LEARN2CONTROL requires not more than a standard web browser and Java, and no additional local installations are necessary.

Each LEARN2CONTROL project is represented by a model, the so-called task-machine, that describes the interconnections of the various subtasks. Each subtask is processed on a web page that is generated by Java server technology. A tailored MATLAB web service handles involved mathematical calculations in a multi-user fashion. The production of new LEARN2CONTROL projects is facilitated by specific authoring tools.

Index Terms - e-learning, engineering education, project-oriented learning, software architecture

INTRODUCTION

Computer-based learning via the internet often is provided by lecture notes that are available online, sometimes enhanced by exercises that are however usually not processed and evaluated automatically and online. Integrated web-learning environments have become increasingly popular, but there still is a strong need for efficient concepts and architectures for internet-based distance learning.

The classical control education does very well in providing the theoretical background for solving well-defined subtasks of practical controller design problems, but usually there is little education in solving larger, not pre-structured problems. The necessary skills for the solution of such problems include the partition of the overall problem into smaller parts, the choice of suitable methods to address these subtasks, the choice of the right sequence and the right level of effort invested in solving the subproblems, and the consideration of the dependencies of the subtasks and the overall goals and constraints of the project.

LEARN2CONTROL, [1, 2], aims at providing an opportunity to acquire this knowledge and experience with project-oriented work by means of authentic control system design projects. Seven design projects have by now been realised. Two of these include remote experiments where the users collect process data and test the designed controllers on real plants via the internet. All realised projects of LEARN2CONTROL start with a detailed problem description and a specification of the goals of controller design and the limitations that have to be taken into account. Then, the users have to find suitable strategies to solve the problem at hand. They have to decide which tasks, e. g. modelling, controller design, simulation, etc., have to be performed, and in which sequence the tasks should be tackled. On a subtask level, different methods are offered and the users have the freedom to decide which method they apply. This may lead to a dead end and to the necessity of returning to a previously reached state in case of unsuitable sequences. The return to a previous state in the design process can also be necessary to improve the design or to compare different parameter settings used in a step or to compare different methods to solve the subproblems.

PROJECT-ORIENTED LEARNING AND THE TASK MACHINE

In order to guide the project-oriented learning process, a learning model was developed which describes the hierarchical structure of all subtasks within a project and all interdependencies between these on an abstract level. In LEARN2CONTROL, the learning model is formulated as a hierarchical knowledge graph, the task-machine, which consists of nodes that represent the subtasks and directed edges that represent interdependencies. In particular, nodes can be nested: e. g. a node "Modelling" might contain "First-principles modelling" as well as "Black-box modelling". Each node is associated with one web page on which the subtask can be processed. A directed edge represents the interdependency between two nodes of one hierarchical level. The directed edges indicate possible paths through the graph. A directed edge between two nodes A and B is interpreted as: "The successful completion of node A is a precondition to work on node B". The work on a project can be represented as a path through the knowledge graph, starting from a predefined starting point, where the design problem is specified, to a given end-point, where e. g. the control performance is validated by experiments or simulations. Besides an optimal solution path, less favourable or even infeasible paths may exist. In contrast to classical learning environments in which the user often has to proceed in a predefined linear fashion, our approach supports the free combination of subtasks and the modelling of the interdependencies between theses subtasks.

THE PREVIOUS LEARN2CONTROL SOFTWARE ARCHITECTURE

I. Description

The previous LEARN2CONTROL environment was developed based upon the features of the Netscape Navigator 4.x. In this version, the visualisation of the hierarchical knowledge graph of a project is realised using a 3D VRML [3] representation in the navigation window for which a VRML player has to be installed. A task-machine object implements the hierarchical knowledge graph and all interdependencies of the subtasks of a project. A graphical representation of the architecture is shown in Fig. 1.

For numerical calculations and simulations, an installation of MATLAB/SIMULINK on the Client PC was integrated into LEARN2CONTROL. By the use of this powerful computation tool, the algorithmic programming effort is minimised, a large variety of methods are supported and calculations can be carried out numerically as well as symbolically. The data exchange with MATLAB was realised by a Netscape Navigator browser plugin that was developed in [4].

A core component of the previous LEARN2CONTROL version is the LEARN2CONTROL-toolbox which provides methods to store all project data, e. g. signals, system equations, system parameters, project status information, etc., in a structured way. The LEARN2CONTROL-toolbox is implemented as an object-oriented data model in MATLAB and performs all calculations within the projects. This structure is very flexible and can be adapted with little effort. All project data is stored locally on the Client PC in the MATLAB workspace and can be transferred to the responsible instructor via internet for evaluation purposes. The communication between the task-machine, the VRML project navigation, and the HTML project pages is managed by a JavaScript library that is located in the top frame of the user interface.

If a remote experiment is integrated in a project, the student has to book experimentation time using the LEARN2CONTROL portal. The central database system schedules the time slots of the remote experiments and controls the access of the users. Data generated during the remote experiment is transferred to the Client PC and integrated into the local data base. After the experiment, this data can be used for further evaluations and computations within the project.

II. Disadvantages and Limitations

While the aforementioned architecture meets the requirements for a web-based education environment, it has a few disadvantages in practice. First, the original version of LEARN2CONTROL requires the client-side installation of some licensed or non-standard software packages, in particular MATLAB and the VRML player. The complete logic of a project is implemented in client-side JavaScript and embedded in local HTML pages, thus opening a door for

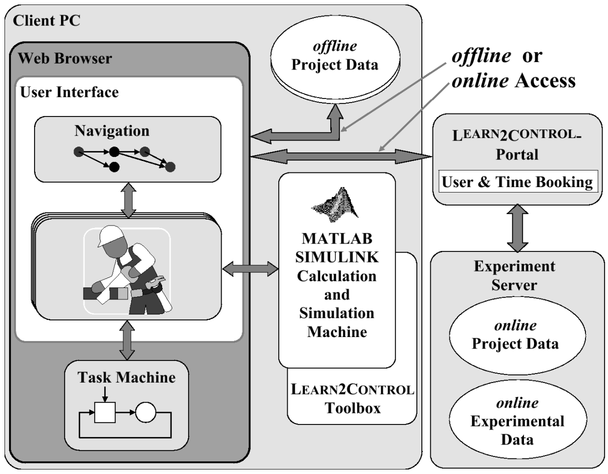

FIGURE 1
STRUCTURE OF THE PREVIOUS LEARN2CONTROL.

unauthorised changes and cheating. The HTML pages can only be used with Netscape Navigator 4.x, since this is the only browser that is compatible with a custom plugin for the communication with MATLAB.

THE NEW CLIENT/SERVER SOFTWARE ARCHITECTURE

I. Overview

To overcome the limitations of the previous version, it was decided to migrate LEARN2CONTROL to a client/server architecture and to make use of newly developed web technologies in this process. Most of the code had to be rewritten and the existing projects were retrofitted to the new environment. Instead of relying on locally installed software, only a standard web browser is now needed to access the LEARN2CONTROL projects. An application server – here Apache Tomcat, but other servlet-containers are possible as well – runs servlets that handle client-authentication, session handling, and the task-machine which models the projects and their status. All servlets in the Java servlet container are programmed in Java using Java Server Pages, offering the full flexibility of Java programming. For computationally extensive operations, a custom server-software is used which allows managing multiple MATLAB workspaces on a single physical server. In the workspaces, calculations can be performed simultaneously, in contrast to the default single-threaded behaviour of the MATLAB Web Server [5] that allows only one thread at a time. Furthermore, the workspaces serve as storages for temporary results. An overview of the new architecture is shown in Fig. 2.

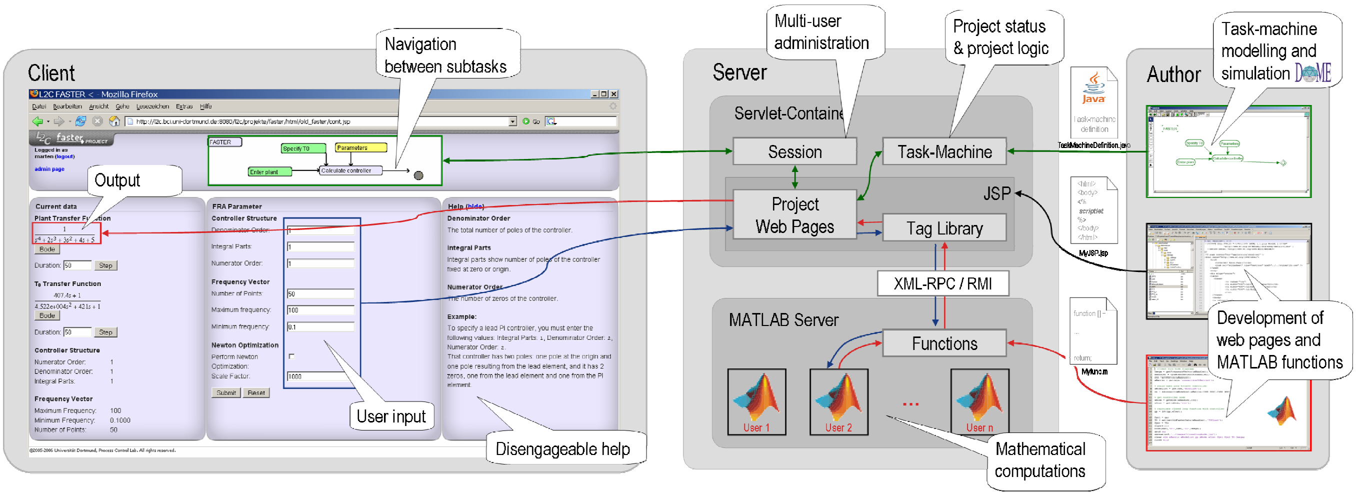

FIGURE 2
STRUCTURE OF THE NEW LEARN2CONTROL.

II. Task-machine

For each user who is logged on to a particular project, a task-machine is created as a servlet on the server side. The task-machine is a Java program that contains the definition and the logic of the learning model for a particular project and the current project status. A local copy of the Java program, an applet in this case, is downloaded to the client if either the project status or the currently active project web page is changed. To visualise the project state and the current subtask for the user, the applet has visualisation capabilities that have been realised in Java2D.

III. Java Server Pages (JSP)

The project web pages are written as Java Server Pages. The distinct feature of JSP is the possibility to hide the Java functionality behind Java Custom Tags that can be used similarly to HTML tags so that the authoring process does not necessarily require Java programming skills. A first Java Custom Tags Library (Tag Library in Fig. 2) has been created which already offers basic functionalities. All project web pages are programmed in Java using JSP. From the markup point of view, writing JSP code is similar to writing HTML. From the business logic point of view however, JSP is very different from HTML + JavaScript in that the complete logic resides on the server. In our projects, we use a model-view-controller (MVC) approach to implementing the logic. Three different JSP pages are used for one subtask: the model JSP describes the processing functionality, the view JSP generates the web page for the client and the controller JSP combines processing and visualisation.

IV. Multi-user Administration

The new version of LEARN2CONTROL embeds a true multiuser environment. Projects can use custom classes for session handling and user administrations. Currently, there are two kinds of users: administrators and normal users. Administrators are allowed to use the configuration backend, a special administration site written and implemented in JSP, and can add, delete and modify users and project assignments. Normal users have the ability to log on to the LEARN2CONTROL environment and to use the projects they are assigned to. User data and configuration data is stored internally in a MySQL database which in turn can be accessed via JNDI, a Java naming and directory service. The architecture of JNDI also hides the actual database from LEARN2CONTROL, making it easy to replace the database with another solution at a later time if needed.

V. MATLAB Workspace Server

The MATLAB Workspace Server is a platform-independent middleware that provides access to multiple MATLAB workspaces running on a single physical server. Each session has its own workspace that is completely isolated from the others. For an architectural overview of the MATLAB Workspace Server, see Fig. 3.

The session JSP (or any other software speaking either the RMI or the XML-RPC protocol) requests a new workspace session via the programming interfaces and can execute arbitrary MATLAB commands and return results and error messages.

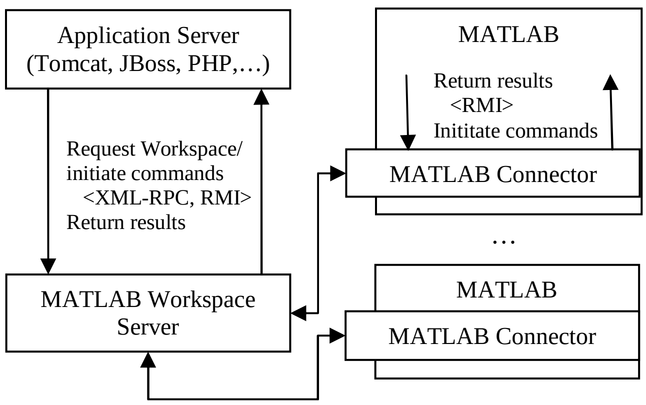

FIGURE 3
ARCHITECTURAL OVERVIEW OF THE MATLAB WORKSPACE SERVER.

Internally, the Workspace Server manages a pool of MATLAB workspaces which are created on demand. Each active workspace corresponds to a running MATLAB process. As, on modern operating systems, the system memory for the required link-libraries is usually shared among multiple processes, this is an efficient approach to provide a multi-threaded, multi-workspace environment. We have tested up to 20 workspaces on a moderately equipped server (Athlon64 3800+ dual-core with 3.5GB of RAM).

AUTHORING TOOLS

I. Task-machine with DOME

The core functionality to author a LEARN2CONTROL project is provided by the software tool DOME [6], which is used to create the task-machine logic. Fig. 4 shows a screenshot of the DOME editor.

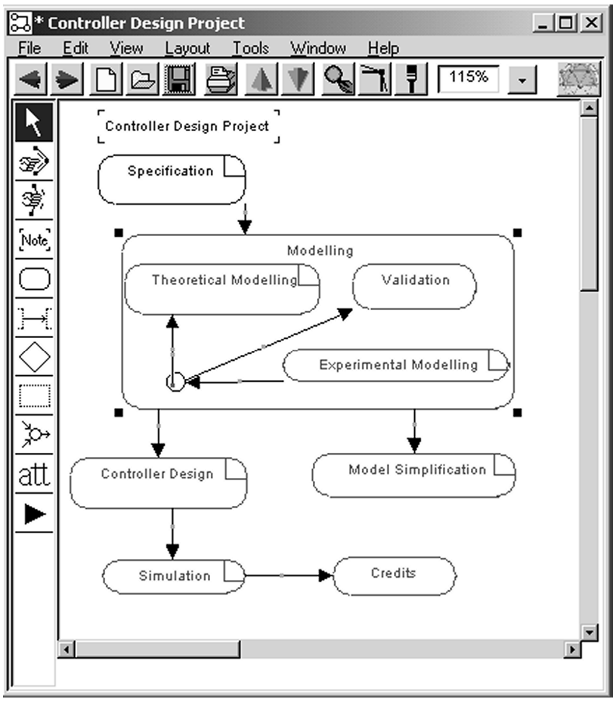

FIGURE 4
THE DOME EDITOR.

A LEARN2CONTROL library within DOME enables the modification and creation of nodes and directed edges of the task-machine via drag and drop techniques. Fig. 4 shows the task-graph of the "Heating System Project", the project with the most extensive logic so far. The logic can be understood as follows:

"Specification" is a prerequisite for "Modelling" which in turn must be completed before "Controller Design" or "Model Simplification" can be reached.

Each node in the figure with a rectangle in the upper right corner is a supernode that contains other nodes that must be completed before the supernode can be completed. "Modelling" also represents a supernode, however with a slightly different way of visualisation. In contrast to the other supernodes, the three contained (super-)nodes, "Theoretical Modelling", "Experimental Modelling" and "Validation" can be seen from the viewpoint shown in Fig. 4, while the sub nodes of e. g. "Controller Design" can only be seen after double-clicking on "Controller Design" and thereby changing the view to the sub nodes.

The LEARN2CONTROL library also provides a simulation functionality to test the consistency of the task-machine logic. After this test, the Java code for the task-machine is generated automatically, using the DOME inherent programming language Small Talk for parsing. In this process, the graphical DOME representation is used as a template for the spatial arrangement of nodes in the client-side visualisation of the task-machine.

II. Project web pages

As mentioned before, JSPs are used that enable the use of custom tag markup. So far, all MATLAB interfacing operations are hidden behind custom tags. Furthermore, custom tags for entering and displaying transfer functions are available. Therefore, the HTML markup of a web page can be written using any text-editor; specialised HTML editors can offer convenience and added functionality. To facilitate the design task, graphical development environments can be used.

III. MATLAB data structure

Involved computations as they arise in control engineering such as simulations or optimisations are performed using MATLAB. The m-files in LEARN2CONTROL are integrated in dynamic object-oriented data structures that are tailored to control engineering tasks. The usual data structure for projects is a dynamic list that may contain different model lists, controller lists, and signal lists. The library of functions can be extended easily by user-defined functions.

LEARN2CONTROL PROJECTS

I. Overview

LEARN2CONTROL comprises seven projects. Three out of five from the previous LEARN2CONTROL version have been retrofitted to the new software architecture, and two new projects are being created making full use of the new technology. The projects that were available in the previous version are

- controller design for the heating system of a reactive distillation column,
- controller design for a hydraulic drive,
- controller design for a twin rotor system,
- controller design for a mobile robot,
- controller design for a chemical reactor.

The first three have been integrated in the new framework. The twin rotor project and the hydraulic drive projects comprise remote experimentation where the designed controllers can be tested on real plants or the identification models can be estimated from data recorded by the users via the internet. Two new learning projects are currently being developed using the new framework, an interface to a tool for the synthesis of multivariate low-order controllers, FASTER, and a learning unit to serve our students as a repetitorium for basic elements of the analysis of dynamic systems. Since the 5 former projects have been described extensively in [2], we restrict ourselves to the description of the two new projects here.

II. FASTER - A tool for the synthesis of multivariate low-order controllers

FASTER is a tool for synthesising multivariate low-order controllers by means of frequency response approximation. The original idea has been described in [7] and the extension to the multivariate case has been published first in [8], [9]. In a recent paper, the application of FASTER to a complex controller design problem, the control of a pilot-plant scale reactive distillation column is described [10]. The algorithm minimises the difference (in terms of the Frobenius norm of frequency responses) between the desired closed-loop behaviour and the behaviour of the closed loop with a reduced-order controller of a predefined structure, leading to a non-convex optimisation problem. For certain standard controller structures, optimal controller parameters can be computed by solving a convex problem, this technique can also be used to obtain good starting points for the non-convex optimisation.

In the FASTER project, designing a controller by frequency response approximation is modelled as a three-step procedure. Fig. 5 shows a web page associated to a subtask of the project. Each web page includes the graphic representation of the task-machine at the top, an overview of the data structure on the left, the main section in the middle, and a help section on the right.

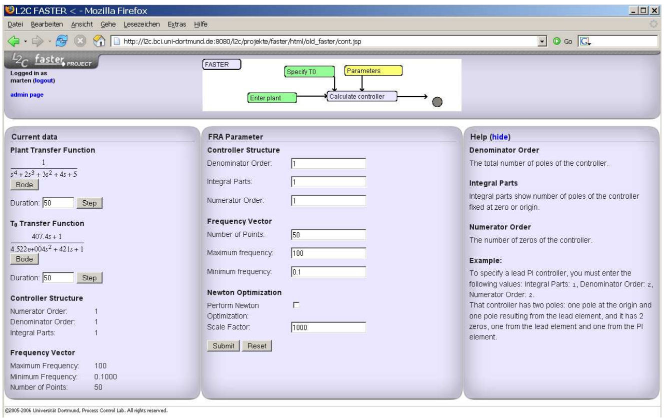

FIGURE 5
A WEB PAGE OF FASTER.

The FASTER project can be best explained by looking at the logic of the task-machine graph at the top in Fig. 5. In the step "Enter Plant", a linear plant model is specified. In the second step "Specify T0", the desired closed-loop behaviour is entered. To simplify the procedure, default models for the closed-loop behaviour such as first-order and second-order transfer functions with delay are offered. In the step "Parameters", the controller structure, the frequency range for the approximation and some other parameters must be chosen. At this point, the subtasks "Enter Plant" and "Specify T0" have already been finished, which is indicated by the green colour. The fact that "Parameters" is the current task is indicated by the yellow colour. Finally, the node "Calculate controller" is shown in blue which indicates that all data has been collected and the controller can be calculated.

In the subtask "Parameters", the parameters of the controller approximation, such as the controller structure (e. g. PI-controller or PID-controller), the frequency range and whether or not the non-convex optimisation shall be carried out are determined. On the left hand side, an overview of the actual data is shown (this is true for all nodes of this project). Therein, the user has an overview of the plant, the desired closed-loop behaviour and the parameters of the approximation algorithm in one figure. Additionally, step responses and frequency responses of the plant and of the desired closed-loop system can be shown (see Fig. 6). On the right hand side, a help window explains the meaning of the parameters for new users. If the help window is disengaged, the extra space is divided between the data overview section and the main section.

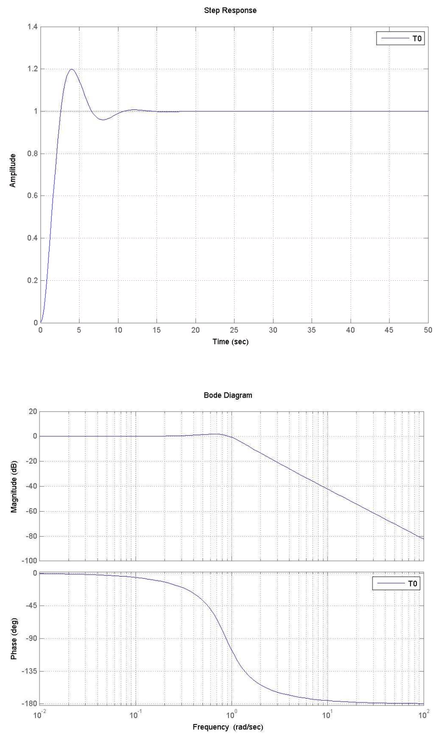

FIGURE 6
STEP RESPONSE (ABOVE) AND BODE DIAGRAM (BELOW).

III. A fundamental course on systems analysis

The goal of this LEARN2CONTROL project is to provide an opportunity to apply theoretical knowledge and to gain experience in basic principles of methods for the analysis of dynamic systems. An illustrative example of a simple continuous stirred reactor system is considered in which an equilibrium reaction of two species takes place.

The first task is to describe the dynamic system by means of three differential equations for the state variables for the concentrations for both substances and for the volume of the content of the reactor. The next step is to calculate the steady state of the system as a prerequisite of a subsequent linearisation at this point. By consideration of the eigenvalues of the resulting Jacobian matrix, a statement about the stability of the system can be made. Thereafter a transfer function for the system at hand must be computed from the previous linearisation. For the resulting transfer function the root locus diagram has to be computed. The graph of the root locus and step responses can also be plotted. In the given example a zero-pole cancellation can be regarded.

The design of this project web page is similar to the design of the previous FASTER project, using the same stylesheets (CSS).

CONCLUSIONS AND FURTHER WORK

The emphasis of the learning environment LEARN2CONTROL is on web-based project-oriented learning of methods that can be applied in solving practical control engineering tasks. The projects are realised using a learning model which describes all possible solution paths as well as the prerequisites for their completion by a hierarchical knowledge graph. The data generated during the execution of a project is stored and saved in an object-oriented data structure that enables the users to store more than one solution. The differences in the performance that result from changes of the parameter sets or from different design methods can be experienced and the effects of different controller settings can be studied. Possibly, certain design steps have to be repeated in order to meet the specifications. In contrast to traditional learning environments, LEARN2CONTROL helps to understand the dependencies and the interactions between the various methods and tasks for control system design and therefore enables the users to gain experience in structuring control engineering projects.

To increase the usability of LEARN2CONTROL, a completely new client/server architecture was devised. These modifications concern a server-side integration of MATLAB, the task-machine, the programming of the project web pages in JSP, and the multi-user administration with a data base connection. Furthermore, the client-side visualisation of the task-machine was changed by using Java technology instead of the outdated VRML technology.

LEARN2CONTROL is continuously used in our engineering education within the Master Programme “Automation and Robotics” at Universität Dortmund. Therefore, LEARN2CONTROL is continuously updated and improved.

ACKNOWLEDGMENT

The authors would like to thank the ”Universitätsverbund MultiMedia des Landes Nordrhein-Westfalen” for the financial support of the project ”Multimediale Lernumgebung für projektorientiertes Lernen der Methoden der Regelungstechnik”. We would also like to thank our former colleagues Andreas Liefeldt, Manuel Remelhe, Kai Dadhe, Sabine Pegel and Thomas Löhl for their most valuable contributions to LEARN2CONTROL. Special thanks to Bastian Steinbach who is currently working on the implementation of a new LEARN2CONTROL project.

REFERENCES

[1] LEARN2CONTROL URL: http://l2c.bci.uni-dortmund.de:8080/l2c/

[2] Völker, M.; Liefeldt, A.; Engell, S.; Schmid, C., “LEARN2CONTROL: A Project-oriented Approach to Teaching Control Engineering”, *Proc. IEEE Conference on Computer-aided Control System Design - CACSD* Taipei, Taiwan, 2004, 184-189.

[3] VRML URL: http://www.karmanaut.com/cosmo/player/ [checked May 2nd, 2007]

[4] Dynamit URL: http://dynamit.esr.ruhr-uni-bochum.de, 2004.

[5] The MathWorks, Inc., “MATLAB External Interfaces”, *Natick, MA*, 2002.

[6] DOME URL: http://www.htc.honeywell.com/dome [checked April 27th, 2007].

[7] Engell, S., “Compensator Design by Frequency-Weighted Approximation”, *Proceedings of the IEE International Conference on Control*, 1988, 253-258.

[8] Engell, S.; Müller, R., “Controller Design by Frequency-weighted Approximation: the Multivariable Case”, *Proc. of the IEE International Conference CONTROL-91* Edinburgh, 1991, 581-586.

[9] Engell, S.; Müller, Ralf, “Fast and efficient selection of control structures”, *Computers & Chemical Engineering*, Pergamon Press, 1992, 157-164.

[10] Völker, M.; Sonntag, C.; Engell, S. “Control of Integrated Processes: a Case Study on Reactive Distillation in a Medium-scale Pilot Plant”, *Control Engineering Practice (Special Section on Award Winning Applications)*, to appear 2007, available online via http://www.elsevier.com