ELECTRONICS AND ELECTRICAL ENGINEERING
ISSN 1392 – 1215 2012. No. 3(119)
ELEKTRONIKA IR ELEKTROTECHNIKA

EDUCATION IN ELECTRONICS AND ELECTRICAL ENGINEERING
T 000
STUDIJOS ELEKTRONIKOJE IR ELETROTECHNIKOJE

eRACoL: a DRBL based Tool for Easy Remote Design and Administration of Educational Computer Labs

A. V. Alejos, F. Gil Castineira, P. Rodriguez Hernandez, F. J. Gonzalez Castano
E.T.S.E.Telecomunicación, University of Vigo,
Campus Universitario, Maxwell St, Vigo, 36310, Spain, phone: +34986812195,e-mail: analejos@uvigo.es

crossref http://dx.doi.org/10.5755/j01.eee.119.3.1374

Introduction

At different levels, we can state that education has experienced deep changes in the last two decades powered by the demands of different agents and factors. These changes have been materialized in the development of new learning and teaching methods that try to improve teacher-students interactions, or the way of presenting the large available amounts of existing information in a comprehensive, structured and efficient form.

Regardless of the selected approaches, these targets entail the development of innovative material aligned with new teaching methodologies, which will be a short-term demand in the new study plans, as is the case of the European universities that must incorporate the European Higher Education Area (EHEA) requirements [1–3].

Instructors from all educational levels should experiment with the many options offered in the quest of a winning teaching/learning formula. From this quest, the need for new methods that will allow a reasonable access to the material and software resources should not be excluded. Equal access to educational material is central to providing a homogeneous background to students from all socio-economic strata, various minorities and educational environments. Limited access to resources can seem unfair from the social point of view.

Much literature has addressed the question of how to promote equal access opportunities to higher education. Among the many solutions offered to solve this issue, especially in terms of sharing and increasing resources availability and/or number, we can list three main options: cooperative networks [4-7]; virtual instrumentation [8-11]; and remote software laboratories [12-15].

The option represented by remote software laboratories offers advantages in terms of resources number and access, as well as other technical issues: maximizing resource availability; optimizing the use of the resources that are already available; maximizing computing power while minimizing costs; permitting coexistence with previously existing operating systems.

At present, the number of existing choices to implement a remote tool to design and administrate a computer laboratory is not small: cloud computing, Linux Terminal Server Project (LTSP), Diskless Remote Book on Linux (DRBL), and so on. The selection of either option is given by a trade-off of the available facilities and capacities (network speed, computer, labs…), as well as the needs to be covered (hardware, software, number of users…). However, a decisive element can be given by the benefits derived from its implementation, such as the level of installation complexity, administration and management. The experience described in the present paper has selected the DRBL option as an optimal solution for a computer based undergraduate lab in the Electrical Engineering degree at the University of Vigo [1].

Another equally important aspect to consider in the development of remote administration tools is the impact of their implementation in the cognitive and affective dimensions of the learning experience. It is important that the tool be helpful not only to teachers' work but also to students by incorporating educational add-ons oriented to facilitate the learning/teaching process and increase motivation. Another factor to select remote software deals with how easily these valuable educational complements can be incorporated.

Finally, another similarly significant aspect to consider is whether remote labs offer a good means to attract prospective engineering students to pursue Higher Education [2]. Competition among different universities, which is due to a population decrease factor, options offered by new degrees and the demands from the industry, turn remote laboratories into valuable educational tools and a noteworthy added value to retain students.

Thus, the purpose of this paper is first to introduce a general scope of the existing tools to design and manage remote labs. Within the analysis of the state of the art, we discuss the appealing high-tech option represented by the emerging cloud computing technology. Then, we describe the choice selected for our experience: DRBL. After that, we detail the implemented remote lab administration tool: eRACoL (e-Remote Administration of Computer Labs).

Other complementary tool as iTALC is analyzed too. Moreover, a scalability analysis of remote solutions is presented. We also introduce some additional future add-ons to improve the presented tool. Finally, educational challenges for remote tools are highlighted. Conclusions close this paper.

Remote educational labs administration: state of the art

The first educational computer science laboratories were formed by a single server in which students did their practices either introducing programmed punched cards or by means of an interactive user terminal.

The rise of personal computers in the 1980s allowed the creation of laboratories with one computer for each student or for a group of them, but the administration of those sets of computers became more difficult. At present, if an educational institution opens a laboratory with entirely new equipment, the first option is to install and configure in each computer the desired operating system as well as the applications that are needed for the proper functioning of the laboratory class. In a medium size laboratory, between twenty and forty equipments, this process can take several days.

On the other hand, the laboratory can have failures and stop working properly if the computers have an operating system with unsettled separation of privileges. That may lead users to modify system configurations, installing/uninstalling applications, which, may finally end up corrupting the environment. In order to avoid these problems, the cloning method has been extended for a few years: the installation of the operating system and applications is only performed for one unit, and a perfect copy of this system is obtained and mirrored usually on the hard-drives of the remaining computers.

This method avoids configuration errors in each installation, but it does not significantly reduce installation time for the entire laboratory. It presents, in addition, a series of other problems:

1. It is usual for laboratory equipments to fail. Hence, they are replaced by models different from the original one. It is quite probable that the copy of the initial installation does not work correctly due to differences in hardware configuration;
2. If a new program needs to be added in the laboratory, the cloning and copy process needs to be repeated for each one of the equipments.

Along with the improvement in the features of local networks and the increase in memory and processing capacity of personal computers, the design of systems called "light terminals," such as LTSP or Edubuntu, initiated. These systems consist of: (1) students' computers or clients - limited to act as remote terminals that show images on their screens as well as to capture the orders introduced by keyboard or mouse-; and (2) one central server - in charge of executing applications, interpreting the received orders and sending the results back to the students' screens.

A great advantage is that this approach makes only necessary to manage one computer: the central server. If it is necessary to install, configure or remove an application, the administrator updates the server and the changes are reflected automatically in the clients.

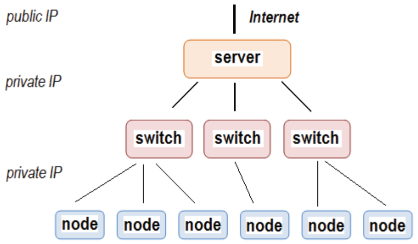


Fig. 1. DRBL concept diagram

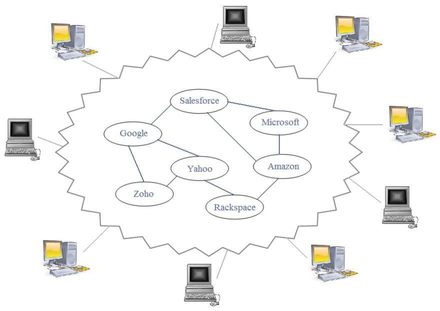


Fig. 2. Cloud computing concept diagram

Another advantage of these systems is that they use old computers that could no longer execute the applications correctly as light terminals.

A main disadvantage this approach presents is that the memory requirements for the server increase considerably depending on the number of users, which makes it difficult to manage laboratories with more than 30 workstations. Furthermore, if we already have a laboratory with modern computers and a system of light terminals is selected, we will be wasting a great calculation capacity.

In this paper we present a laboratory designed and managed by using a system developed with the open source tool DRBL. Thanks to the NFS protocol (Network File System), DRBL enables clients to load the Linux operating system at the start up and to access the applications and personal data stored in the server's file system. Similar to the LTSP, DRBL allows workstations to boot from a server on a network prior to booting the operating system on the local hard drive. This approach means that DRBL can support more clients (usually 30 to 40) per server than LTSP (usually 20 to 30 clients). A DRBL concept diagram is shown in Fig. 1.

Internet connection bandwidth increase has allowed the advent of another laboratory management method, which consists of the utilization of virtual machines that are executed in the "cloud," following the *cloud computing* paradigm. A concept diagram is shown in Fig. 2.

The concept of a "cloud lab" provides some advantages, which other paradigms cannot achieve:

1. Scalability: Since there is no central server but a set of distributed computers, increasing the number of

resources is as easy as adding new computers to the cloud without the need for new configurations or installations. In a central server approach, if more resources are required, some possible solutions are performing upgrades up to the limit supported by the hardware; beyond that point, the only solution is substituting the equipment by a more powerful one;

2. Virtual machines: Clouds provide independence of software from hardware by means of virtual machines. This fact simplifies software installation: now only one virtual machine needs to be installed and configured because the others can be just clones of this first one. Nevertheless, virtual machines consume more resources than real ones and, as a consequence, larger investments in hardware are required, although the resulting benefits justify them.

In spite of all the advantages, Higher Education feels skeptic about cloud computing. The corresponding migration of their sensitive data into remote, worldwide data centers introduces profound legal, compliance, and political issues. Another main benefit provided by the cloud computing is the cost reduction for acquisition of software licenses but also hardware items. The concept of "green technology" also relates to this buzzing model, which will contribute to reduce the required number of computers and hardware infrastructure.

It is clear that regardless of the kind of lab setting, it is very important to have administration tools that help with lab management by facilitating equipment setup and avoiding usual errors. One of these tools, DRBL, has been chosen as the core of the implemented tool.

DRBL on its own is not equipped with educational value, but it can work with complementary tools as iTALC, which could significantly boost the educational dimension of the resulting combination. We will explain in the following section the main used operation modes of DRBL, as well as the incorporated educational add-on, iTALC.

We can also think of using DRBL with cloud computing since it allows remote booting virtual machines in almost any client terminal.

Tool development: description of the software

DRBL was born as a tool to simplify the maintenance of personal computer (PC) clusters. We explained that DRBL was used as the basis to create a diskless/systemless environment and also as a tool to deploy diskfull nodes with the disk-cloning tool Clonezilla. Obviously, a computer classroom is not a PC cluster system, but they have similar needs: both require all computers with the same operating system, tools and applications with access to a common place to store information (where students save their data and where clusters have the information to be processed).

The LTSP was traditionally used in classroom environments to share a server between several thin clients (low powered terminals, without disks or computers running a graphics server). It became a popular approach because with LTSP it is possible to get a high performance server and convert the existing obsolescent computers in thin clients, being this solution cheaper than buying new computers for each student. Furthermore, the configuration, management and maintenance of the computer laboratory are really simplified. Nevertheless, a LTSP based solution presents a drawback: applications tend to become more and more resource intensive and the server can manage only a few clients.

At the moment, personal computers are really affordable and powerful, which means that even relatively old computers are good enough to run the applications used in a computer classroom, but it is really interesting to maintain a centralized configuration and management system like the provided by LTSP. DRBL is a solution, because it is a centralized system able to use local resources (memory, processor, GPU…) in the clients. Newer versions of LTSP also provide this functionality.

In brief, if you want to have a centralized management system for a computer classroom with:

- Very old computers: use LTSP and get a good server, or try to use low resource applications;
- Newer computers: use DRBL or the newest versions of LTSP with support for diskless nodes (also known as *thin-fat clients* or *hybrid clients*).

With DRBL is possible to have computers without any kind of permanent storage in the classroom, saving money and avoiding disk failures. In addition, it is likely to deploy a new operating system in a computer laboratory without modifying the computers, so it results possible to maintain the old configuration to test a new one (and go back to the previous in case of problems) or to keep a multi environment computer classroom: for example Windows installed locally in each computer and Linux loaded over the network with DRBL.

With DRBL the administrator has to install or configure the applications only on the server, and they become immediately available for the clients. Moreover, more computers can be added to the laboratory without making any change (or minimal changes) in the server.

Tool features: operation modes

A DRBL is interesting for us because of its capabilities to create a *diskless* computer classroom, but it has several interesting operation modes which can be chosen in the server with the help of one of the provided tools (dcs):

- Remote Linux for powerful clients: is the DRBL operation mode we have already introduced;
- Terminal: remote display, or the typical terminal mode for thin clients;
- Remote memtest: checks the memory of the machine to detect problems in the RAM;
- Remote FreeDOS: boots FreeDOS from the server;
- Netinstall: installs a GNU/Linux distribution in the local hard-drive, bootstrapping from the network;
- Small Linux: loads in temporal memory a "Small Linux Distribution";
- Clonezilla: used to clone and deploy disk images.

Once DRBL is configured in the server, any of these modes can be accessed from the main menu. It incorporates also PXELINUX which allows the creation of user menus to choose between different operating modes.

Complementary tools: iTALC, an e-Learning/ remote student assistance tool

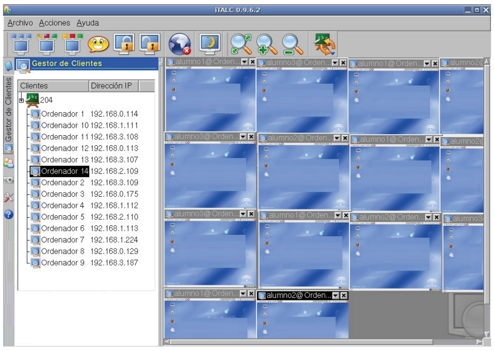
It is necessary the inclusion of some kind of complementary capability that allows the access to each computer in the lab per remote thus facilitating individual student assistance. In order to implement this capability we have selected iTALC.

iTALC (intelligently Teaching And Learning with Computer) is a didactic free open source software for remotely managing, monitoring and controlling computers in a Linux or Windows environment, which is aimed at a educational purposes. The program was developed in Trolltech Qt/C++ as a free alternative to similar commercial software named MasterEye. All functions are based on the RFB (remote frame buffer) protocol used for remote access to graphical user interfaces. iTALC works completely with TCP connections.

The "*per remote"* feature provided by iTALC gives teachers the opportunity to monitor and control the flow of the trainee activities, from the information presentation to the students up to the evaluation process. By this way, it supports the classroom work with modern technique that results compatible with many teaching and learning methodologies approaches. So it can be used either with the master class or with active learning procedures.

Once the application starts, the desktops of all the classroom computers appear on the main window in the teacher's computer. This mode is called the watch mode, and the result is named the classroom map. One teacher's window can monitor several classroom maps. A remote session is opened between the teacher's computer and every computer in the map.

It is possible for the teacher's application to control clients in Windows, Linux or both of them. In Fig. 3 we can observe the teacher's window once the application started, with the classroom map that shows the remote sessions successfully opened.



Clientes	Dirección IP
204	
Ordenador 1	192.168.0.114
Ordenador 10	192.168.1.111
Ordenador 11	192.168.3.108
Ordenador 12	192.168.0.113
Ordenador 13	192.168.3.107
Ordenador 14	192.168.2.109
Ordenador 2	192.168.3.109
Ordenador 3	192.168.0.175
Ordenador 4	192.168.1.112
Ordenador 5	192.168.2.110
Ordenador 6	192.168.1.113
Ordenador 7	192.168.1.224
Ordenador 8	192.168.0.129
Ordenador 9	192.168.3.187

Fig. 3. iTALC main screen at teacher's computer

One of the most attractive options given by the use of iTALC is that it would allow *home-schooling*. This networking tool is not only restricted to a subnet, otherwise it can be accessed from anywhere on the Internet. Therefore students at home can join lessons via virtual private network (VPN) connections just by installing an iTALC client.

However, iTALC shows one limitation: it does not include any support for cloud computing and virtual machines management. Anyway, this fact does not constraint the scalability of a solution that incorporates this software as a complementary tool for remote administration within an educational environment.

Scalability analysis

The solution here introduced as a tool for remote design, configuration and administration of remote computers laboratories is based on DRBL as main core and iTALC as an add-on for educational tasks.

The scalability of the option here introduced depends then on the server machine used to allocate DRBL, in terms of hardware and software. The limit on the number of sub-networks that the server side can manage is given by the number of network cards that can be installed on this machine. The number of network cards that can be simultaneously handled depends strongly on the underlying involved technologies. For a bus as PCI, thinking of a RAID of network cards, the limit can be a large number, but an unsolvable constraint can be found on the operating system side to handle this amount of hardware resources.

On the other side, as we mention in the previous section, iTALC as a networking tool can be accessed from anywhere on the Internet, so via virtual private network (VPN) connections could be handled. The limit is given again by the operating system in terms of the maximum number of connections allowed or the achieved quality. The IP address type, VP4 or VP6, also will influence the limit.

The option represented by the cloud computing paradigm is still under observation and shows deep lacks in different aspects for its application in the higher education: network security, access availability, and data protection. The scalability feature is intrinsic to the cloud computing and the large amount of publications found in literature present it as a medium/long-term solution to be implemented in education at different levels, very especially in the universities. Cloud computing gives the illusion of near-infinite resources available on demand.

In any adopted solution, scalability should be transparent to users. From the scalability feature interesting prospective usage can be derived. So, we can think in sharing large-cost or free software for district schools or universities. This would reduce significantly the budget amount yearly spent by educational institutions in software licenses.

Application in the lab: system architecture

Firstly, it is necessary to have the correct hardware elements to create the computer classroom, as show in Fig. 4. Of course, we need a mid-powered PXE capable computer for each student. Usually, it is necessary to enable the PXE boot capability and to select the network boot option in the BIOS.

The network is also very important: the faster, the better. It would be fantastic to have Gigabit Ethernet connections, however Gigabit switches with the necessary number of ports can be expensive. A possible solution is to use Gigabit connections between the server and the switch, and 100Mbps connections between the switch and the clients. If we use a 100 Mbps switch, the performance of the classroom can decrease significantly. The usage of a HUB is strongly discouraged.

In the example illustrated in Fig. 4 we have installed DRBL in two different rooms, but using only one server. In that case a server with two Gigabit cards could be utilized, or several switches could be connected to create a Gigabit backbone. However, if possible, it is recommended to increase the number of Ethernet cards in the server as the number of clients increases, to get DRBL to work more efficiently. The server is also connected to Internet, and acts as a Network Address Translation (NAT) server for the computers in the classrooms.

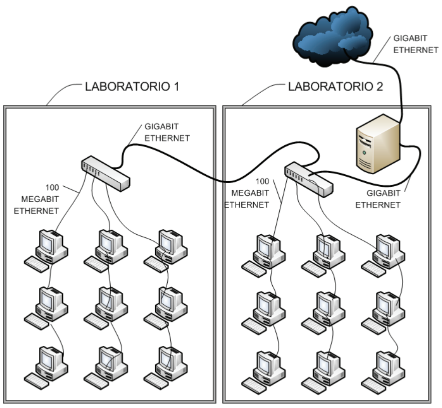


Fig. 4. System architecture

Finally, it is critical that you select a server powerful enough to send the required information from his disks to the clients at a high speed, and with enough storage space to keep all the data for the users and to install all the necessary applications.

Application in the lab: DRBL installation

The DRBL supports several GNU/Linux distributions: Debian, Ubuntu, B2D, Red Hat, Fedora Core, Mandrake, Mandriva, CentOS, Scientific Linux, RHEL, SuSE and OpenSuSE. In brief, it is necessary to install the desired GNU/Linux distribution in the server and configure all the network cards. One of them should be connected to the Internet and the rest to the computers available in the classroom.

Once installed the GNU/Linux distribution, it is necessary to install the DRBL packages adding a new repository. During the installation DRBL will ask you about the network configuration and also about the desired DRBL mode: full DRBL, full Clonezilla, DRBL Single System Image (DRBL SSI), and clonezilla Box.

In the Full modes, a “/etc” and a “/var” directory are created in the server for each client. In the DRBL SSI mode all the clients will have the same “/etc” and “/var”, so they will share the configuration. This mode requires less bandwidth and uses less disk space, but all the computers of the classroom must have the same hardware.

The clients don’t need any installation; it is only necessary to activate the PXE boot option in the BIOS. If the computer is not PXE capable, it is possible to use the GPL network bootloader Etherboot by installing it in the local hard drive, in a floppy or in a CD.

DRBL provides also several tools to manage the installed system, for example to copy files to all the users of a group or to address a command to all DRBL clients.

Performance analysis

To evaluate the performance of the implemented tool we can make an analysis of some parameters which become important from the administrator point of view. Among them we can mention: number of simultaneous connections to the DRBL server, latency, delay, number of subnets, server requirements (RAM, CPU), storage technology, peak data transfer rate, and so on.

From an educational point of view, the performance analysis results more complex since both cognitive and satisfaction levels should be assessed in order to achieve a complete evaluation of the implemented tool.

There are not many references in literature regarding the assessment of remote lab experiences from this point of view. They usually share satisfactory results by comparison with classical hand-on lab implementations [15].

We can infer that the *per remote* presentation of information to the students does not discourage them. On the other hand, it can have an important effect on their motivation for learning.

Future add-ons

Among the main future add-ons in mind we can show up the inclusion of virtual machines option. However, no few problems can be found regarding this choice. The virtual machines should be supported on the server, so this can carry on large delay and latency for the display terminals used by users of these virtual computers.

The processing load of the server can be increased to critical values compromising the performance of the thin or hybrid terminals using the system. Nevertheless, the virtual terminals can limit their usage to a few applications without excessive requirements (priority, delay, data traffic …).

Security appears as a particularly desirable add-on to take into account. Besides of providing safe network communication, the privacy guarantee for stored user data cannot be forgotten. These two last concerns are shared by mostly remote tools, including cloud computing.

Conclusions

In this paper we carry out a review of the different options available to implement a remote configuration system for a computer science educational laboratory. We introduce a system suitable to simplify the administration tasks based on DRBL. It is especially important to count on this type of tools in those cases in which the own professor is the person in charge of the maintenance, so that the effort employed in this task can be reduced and it becomes possible to focus on the really important deal: teaching.

We have outlined some advantages provided by the remote tools. The main of them is to increase the number of hardware and software resources. The remote tools not only simplify the teacher's work, they can contribute to enhance aspects as important as the cognitive level, by increasing the students' motivation whether the software becomes easy and attractive to them.

In last term we cannot forget to mention the social dimension as is the possibility to make available resources for low-budget institutions, and therefore to ensure a fair access to high tech opportunities. Finally, it is not less important to mention the "green technology" dimension that the remote lab option represents by reducing the involved hardware infrastructure and items that can be shared by a larger number of students instead of providing computers in an individual form.

Acknowledgements

We would also like to thank the support received from the People Program of 7th FrameWork Programme (2008 Marie Curie IOF Action).

References

1. **Vazquez Alejos A., Muhammad D., Mohammed H. U.R., Garcia Sanchez M., Cuiñas I.** Educational System to approach teaching of bi static noise radar // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No. 6(102). – P. 71–74.
2. **Lopez–Martin A. J.** Attracting Prospective Engineering Students in the Emerging European Space for Higher Education approach // IEEE Transactions on Education. – IEEE, 2010. – No. 53(1). – P. 46–52.
3. **Vazquez Alejos A., Garcia Sanchez M., Gay Fernandez J.A., Cuiñas I.** Platform for teaching of location technologies based on Zigbee Wireless Sensor Networks by learning–through–play theory // Education Engineering. – Madrid, Spain, IEEE, 2010.
4. **Habermann A. N., Notkin D.** Gandalf: software development environments // IEEE Transactions on Software Engineering. – IEEE, 1986. – No. 12(12). – P. 1117–1127.
5. **Hales D.** From Selfish Nodes to Cooperative Networks — Emergent Link–Based Incentives in Peer–to–Peer Networks // Fourth International Conference on Peer–to–Peer Computing, 2004. – P. 151–158.
6. **Macedonia M. R.** A network software architecture for large scale virtual environments. – PhD dissertation. – Naval Postgraduate School, Monterey, California, 1995.
7. **Marill T., Roberts L. G.** Toward a cooperative network of time–shared computers // Proceedings of the AFIPS Joint Computer Conferences, 1966. – P. 425–431.
8. **Durán M. J., Gallardo S., Toral S. L., Martínez–Torres R., Barrero F. J.** A learning methodology using Matlab/Simulink for undergraduate electrical engineering courses attending to learner satisfaction outcomes // International Journal of Technology and Design Education, 2007. – No. 17(1). – P. 55–73.
9. **Ertugrul N.** Towards virtual laboratories: a survey of LabView–based teaching/learning tools and future trends // Int. Journal of Engineering Education, 2000. – No. 16(3).
10. **Hesselink L., Rizal D., Bjornson E.** CyberLab, a new paradigm in distance learning // Stanford University, 2000.
11. **Ma, J., Nickerson, J. V.** Hands–on, simulated, and remote laboratories: A comparative literature review // ACM Computing Surveys, 2006. – No. 38(3).
12. **Balamuralithara B., Woods P. C.** Virtual laboratories in engineering education: The simulation lab and remote lab // Comput Appl Eng Educ., 2009. – No. 17 (1). – P. 108–118.
13. **Colwell C., Scanlon E., Cooper, M.** Using remote laboratories to extend access to science and engineering // Computers & Education, 2002. – No. 38(1–3). – P. 65–76.
14. **Nedic Z., Machotka J., Nafalski A.** Remote laboratories versus virtual and real laboratories // 33rd Annual Frontiers in Education, 2003. – No. 1. – P. T3E1–6.
15. **Nickersona, J. V., Corterb, J. E., Eschea, S. K., Chassapisa, C.** A model for evaluating the effectiveness of remote engineering laboratories and simulations in education // Computers & Education). – 2007. – No. 49(3). – P. 708–725.

Received 2011 07 23
Accepted after revision 2011 10 31

A. V. Alejos, F. Gil Castineira, P. Rodriguez Hernandez, F. J. Gonzalez Castano. eRACoL: a DRBL based Tool for Easy Remote Design and Administration of Educational Computer Labs // Electronics and Electrical Engineering. – Kaunas: Technologija, 2012. – No. 3(119). – P. 103–108.

At this moment, the importance of computers in all the areas of knowledge is undeniable. Students must acquire competencies in computer use during their primary and secondary studies. Later on, during college education they must learn to use basic computer science tools used in their respective disciplines. On many occasions the professor is the person who designs the lab set-up, and also sets the configuration, installs and maintains all the computers. These tasks, especially the maintenance of the software and hardware environment, are not simple because, among other reasons, the design of a laboratory determines who uses it and how. Besides, those tasks are very important, as shown by all the research in the literature. In this paper we present a tool that simplifies this work in a very noteworthy way. That tool centralizes the installation, configuration and maintenance of the applications in a remote single server. Ill. 4, bibl. 15 (in English; abstracts in English and Lithuanian).

A. V. Alejos, F. Gil Castineira, P. Rodriguez Hernandez, F. J. Gonzalez Castano. Lengvo nuotolinio edukacinių kompiuterinių laboratorijų kūrimo ir administravimo įrankis nuotolinio Linux įsikrovimo pagrindu // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2012. – Nr. 3(119). – P. 103–108.

Pirmosios ir antrosios pakopos studijų metu studentai turi įgyti kompiuterių naudojimo kompetencijas. Vėliau mokydamiesi jie turi išmokti naudoti pagrindinius kompiuterinius įrankius, taikomus atitinkamose disciplinose. Daugeliu atveju profesorius sukuria laboratorijos struktūrą, nustato jos konfigūraciją, įrengia ir prižiūri visus kompiuterius. Šie uždaviniai, ypač programinės ir aparatinės įrangos aplinkos priežiūra, nėra paprasti, nes laboratorijos projektavimas lemia, kas ir kaip ja naudosis. Pateikiamas įrankis, kuris supaprastina šį darbą. Jis centralizuoja taikomųjų programų įdiegimą, konfigūravimą ir palaikymą nutolusiame serveryje. Il. 4, bibl. 15 (anglų kalba; santraukos anglų ir lietuvių k.).