Addressing technical and organizational pitfalls of using remote laboratories in a commercial environment

Pablo Orduña*†, Luis Rodriguez-Gil*, Ignacio Angulo†, Gabriel Martinez†, Aitor Villar*,
Unai Hernandez*, Paola Buitrago‡, Raúl Camacho‡, Paola Marmolejo‡, Javier Garcia-Zubia†
*LabsLand
Bilbao, Spain
Email: {pablo,luis,aitor}@labsland.com
†DeustoTech - Fundación Deusto,
Bilbao, Spain
Email: {pablo.orduna,ignacio.angulo,unai.hernandez,zubia}@deusto.es
‡ UNAD, Medellín, Colombia
Email: {paola.buitrago,raul.camacho,paola.marmolejo}@unad.edu.co

***Abstract*—A remote laboratory is a hardware and software solution that enables students to interact with real equipment located somewhere else on the internet. This way, students interact with a real laboratory as if they were on a hands-on-lab session. Once the equipment is remote, it is also possible to share it among institutions, so students from one school or university can access a lab in another university. While there is an interest by many universities of sharing their laboratories, and there are several experiences doing so, the impact has been typically limited. One of the reasons for the limited impact is the lack of robustness in most solutions due to technical issues, which leads to a lack of trust and interest by the potential consumers. LabsLand is a spin-off of the WebLab-Deusto research group which sells access to laboratories of universities to other schools and universities. This contribution analyzes through use cases what are the technical and organizational pitfalls that were found in the process of taking real laboratories and making them available commercially and what are the solutions used to tackle the issues arisen.**

I. Introduction

A remote laboratory is a software and hardware tool that enables students to access real equipment located somewhere else on the Internet. For example, students learning how to program a small Arduino-based robot can write the code online. Then, in the web browser, they can connect to a remote laboratory that shows through a webcam how a real robot behaves in a real environment running their code.

In the literature there is a wide range of remote laboratories in many fields (e.g., robotics, electronics, physics, chemistry). Software frameworks have been developed to make the development of remote laboratories more affordable (e.g., Remote Laboratory Management Systems such as WebLab-Deusto[1] [1], Relle[2] [2][3], iLab Shared Architecture[3] [4] or RemLabNet[4] [5]) and tools (e.g., gateway4labs[5] [6]) to provide integrations with educational tools (such as Moodle, Sakai or other LMS, both through ad hoc solutions and through standards such as IMS LTI) or repositories linking remote and virtual laboratories (such as Go-Lab[6] [7], [8]).

In particular the WebLab-Deusto research group has created different open source technologies (WebLab-Deusto[7], weblablib[8]) for developing remote laboratories and many types of laboratories, as well as different remote laboratories for over a decade. However, just like most of the research on the literature, most laboratories have been mostly used by the same university that created the laboratory, or by others but with no reliability guarantee.

For this reason, the group has recently started a spin-off startup called LabsLand[9], which provides access to laboratories from different providers (universities) to different consumers (universities and schools). This initiative has led to the creation new solutions to enable the provision of availability guarantees; which is something not addressed in the literature for professional support. Techniques used in Safety Critical Systems are only covered for developing remote laboratories for students learning the techniques [9]; not applied to other remote labs.

In particular, this contribution covers the strategies taken for two types of laboratories:

- The Arduino-based robot remote laboratory, owned by LabsLand (main part of this contribution)
- A set of laboratories from different universities, located in their own buildings.

The contribution is organized as follows: Section II explains the applied theoretical framework (including remote

[1] http://weblab.deusto.es
[2] http://relle.ufsc.br
[3] http://ilab.mit.edu
[4] http://www.remlabnet.eu
[5] http://gateway4labs.readthedocs.org
[6] http://www.golabz.eu
[7] http://weblabdeusto.readthedocs.org
[8] https://docs.labsland.com/weblablib/
[9] https://labsland.com

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

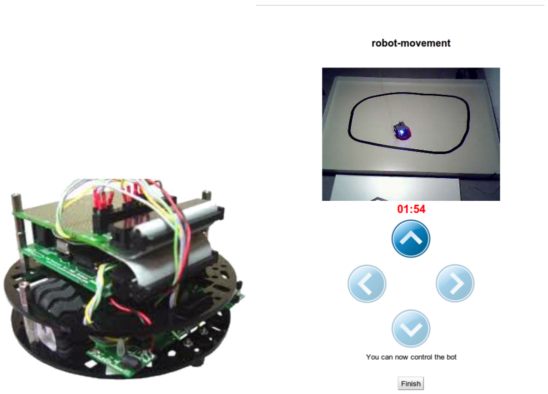

Fig. 1. Robot laboratory [10]. At the left, the mobile robot itself. At the right, the user interface once the program has been submitted.

laboratories, federations of remote laboratories -sharing between different institutions in a transparent way to the student without dealing with creating credentials in each institution-, interoperability and load balancing -both federated and local-). Section III explains the use cases (an Arduino-based robot used by several schools concurrently) and Section **??** the pitfalls found during the process (types of technical -e.g., issues with the robot, networking, field- and organizational problems found). Section V explains the solutions implemented to solve the problems addressed. Finally Section VI provides the conclusions and future work.

II. THEORICAL FRAMEWORK

This section introduces the concepts of remote laboratories, Remote Laboratory Management Systems (RLMS), remote laboratory federations and portals for sharing remote laboratories.

A. Remote Laboratories

A remote laboratory is a software and hardware tool that allows students to remotely access real equipment located in the university. Users access this equipment as if they were in a traditional hands-on-lab session, but through the Internet. To show a clear example, Figure 1 shows a mobile low cost robot laboratory described in [10]. Students learn to program a Microchip PIC microcontroller, and they write the code at home, compile it with the proper tools, and then submit the binary file to a real robot through the Internet. Then, students can see how the robot performs with their program through the Internet (e.g., if it follows the black line according to the submitted program, etc.) in a real environment.

In this line, there are many examples and classifications in the literature [11], [12]. Indeed, remote laboratories started over two decades ago [13], [14], [15], and since then they have been adopted in multiple fields: chemistry [16], [17], physics [18], [19], electronics [20], [21], robotics [22], [23] and even nuclear reactor [24].

B. Remote Laboratory Management Systems

Every remote laboratory manages at least a subset of the following features: authentication, authorization, scheduling users to ensure exclusive accesses -typically through a queue or calendar-based booking-, user tracking and administration tools. These features are common to most remote laboratories, and are actually independent of the particular remote laboratory settings. For example, an authentication and queuing system is valid both for an electronics laboratory and for a chemistry laboratory.

For this reason, Remote Laboratory Management Systems (RLMSs) arose. These systems (e.g., WebLab-Deusto[10], Relle[11], MIT iLabs[12] or Labshare Sahara[13]) provide development toolkits for developing new remote laboratories, as well as management tools and common services (authentication, authorization, scheduling mechanisms). The key idea is that by adding a feature to a RLMS (e.g., supporting LDAP, a Learning Analytics panels [25] or similar cross-laboratory features), all the laboratories which are managed with that RLMS will support this feature automatically.

C. Federating Remote Laboratories

As previously stated in the introduction, a key factor of remote laboratories is that once the laboratory is available on the Internet, it can also be shared with other institutions.

To do this, there are three general approaches:

- Leave the laboratories completely open, so whoever wants to use them can use them. This may reduce the chances of providing proper Learning Analytics or supporting proper accountability mechanisms, in addition to avoiding priorities among students coming from different institutions, leading to a tradeoff between accessibility and advanced features [6].
- Share accounts between the different RLMS: if *University A* want to use laboratories of *University B*, then someone in *University A* will provide a list of usernames to *University B* and students will go to this institution using credentials in *University B*. Ideally, some federated authentication could be used to avoid providing credentials in different domains (such as Shibboleth, OAuth or similar), but it is not typically the case.
- Federate laboratories: if a RLMS supports federation, then if installed in two different institutions (e.g., *University A* and *University B*), students of *University A* will go to the RLMS of *University A* and they will transparently use laboratories in *University B*, working in a institution-to-institution basis (so *University B* does not need to know the list of students of *University A* and simply rely on an existing agreement with that university).

From the items in this list, the most advanced mechanism is the federation of remote laboratories through proper protocols oriented to market-like situations. These federation protocols have been used for fostering interoperability between RLMS [1]. These interoperable bridges between different systems can

[10] http://weblab.deusto.es
[11] http://relle.ufsc.br
[12] http://ilab.mit.edu
[13] http://github.com/saharalabs

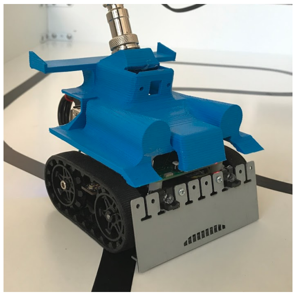

Fig. 2. LabsLand Arduino robot: a Pololu Zumo 32U4 with a 3D-printed shield and a Raspberry Pi

be enhanced if properties such as transitivity or federated load balance are provided [26].

D. Local and federated load balance

A remote laboratory relies on a hardware device (the laboratory itself). In few cases, the same equipment supports multiple concurrent users, such as in the VISIR remote laboratory [20] (supporting up to 60 concurrent individual students). But most of the time, a single instance of the remote laboratory can only support a single user or a single group of users working together [27]. Furthermore, if the equipment is not available (due to maintenance or failure), the service is interrupted.

For this reason, certain RLMS starting supporting remote laboratories load balancing. In its local version (supported by WebLab-Deusto [28] or Labshare [29]), the system supports to have multiple copies of a remote laboratory, balancing the load of users among them. This way, if there are 5 copies of a remote laboratory, and 6 students access the laboratory, they will be distributed among the 5 copies and 1 will be in a queue waiting for any of the 5 current students to finish. In its federated version (supported by WebLab-Deusto [30] and Relle), multiple copies can be available in different institutions or deployments, creating a shared queue. In the case of WebLab-Deusto, this queue is distributed and automatically managed by the different systems through a set of primitives; while in the case of Relle this effort is centralized.

III. Use Case

The following section describes two key use-cases. First, the Arduino Robotics remote laboratory, an example of a complex real-time remote laboratory developed in-house and designed for high reliability. Second, a set of externally-provided remote laboratories, that were to be integrated in the laboratory marketplace.

A. Arduino Robotics laboratory

Nowadays, one of the key goals of policy-makers worldwide is to promote the inclusion of scientific investigations in courses for all ages [31], [32]. Institutions such as the European Union are increasing their efforts to promote STEM, and schools are including subjects that combine programming and robotics [33], [34]. The Arduino Remote laboratory is aligned with those goals.

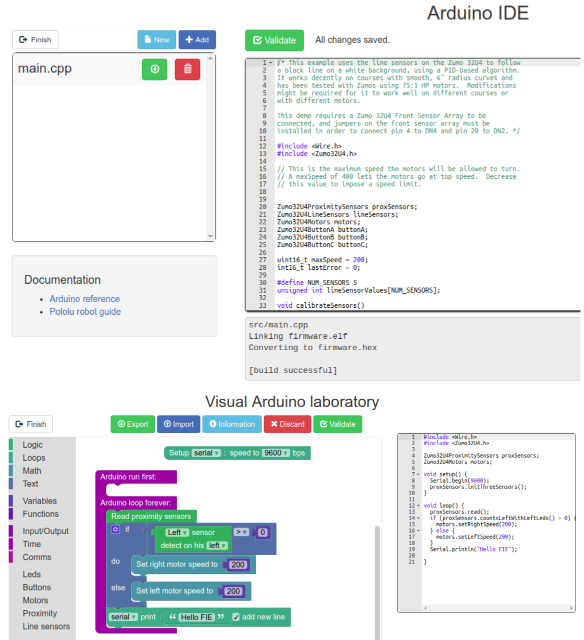

Fig. 3. Programming interfaces for the robotics laboratory: pure Arduino code or Blocks-based

Through this technology, students can learn programming and basic robotics through a visual programming language, by programming and remotely a real robot.

The real robot that students can access remotely is an Arduino-powered Pololu Zumo 32U4 (see Figure 2). LabsLand has modified it to enable remotization. The main challenge has been to integrate an on-board Raspberry Pi[14] to program the Arduino remotely, act as a hardware interface for certain peripherals, and run specific software. The original robot has also been modified to include a custom-designed power cord that can maximize uptime while ensuring that the robot cannot be entangled. The robot is covered in a custom 3D printed chasis that protects the internal components and that is also used for the automated fault-detection system, which is powered by computer vision.

Students can program and control the robot through a fully web-based interface (see Figure 4), needing only their browsers and an Internet connection. To program it, they can use either Arduino code (a text-based C-like language) or a visual blocks-based programming language known as Blockly (see Figure 3). Blockly is designed to be intuitive and has been used successfully by children [35].

When their program is ready, they can upload it into the robot through the same web-based interface. The robot will run it, and they will be able to see whether it is behaving as they expect in real time, through an interactive live-stream [36] that is provided by a webcam. They can also interact with the robot (and thus with their program) in real-time using various peripherals, including, among others, buttons and a serial terminal.

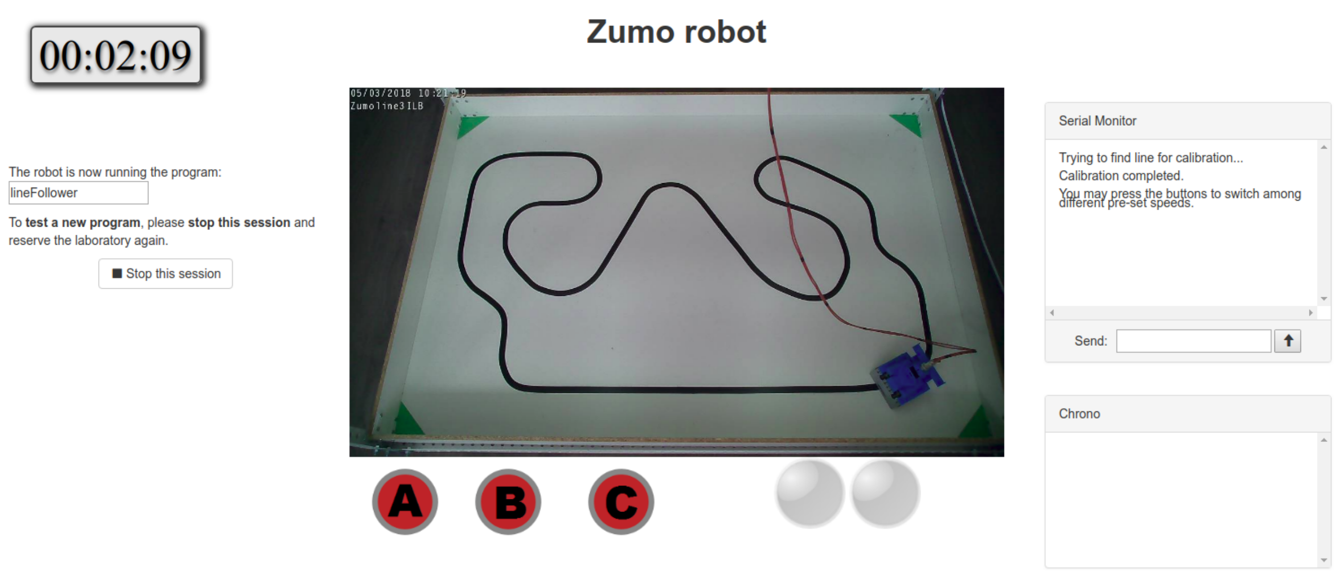

Fig. 4. LabsLand Arduino robot user interface: students select a program they made and then see how the robot interacts in a real environment. Students can use buttons and the serial port at any moment

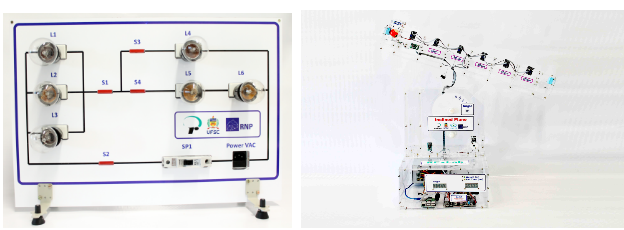

Fig. 5. External laboratories located in other universities (not in LabsLand)

B. External laboratories

Some of the key laboratories provided by LabsLand are developed in-house. A good example is the previously described Arduino robot. However, the goal is to act as a marketplace of many different providers. To fulfill this goal, very different remote laboratories need to be integrated in a coherent way into the platform.

A first remarkable laboratory is a kinematics laboratory. Developed by the Relle team at UFSC [15], it allows students to specify the inclination angle of an inclined plane. Then, they can drop a ball from the top. A set of sensors registers the time at which the ball goes through each point, and students can thus calculate the speeds and forces involved.

A second remarkable laboratory, by the same authors, is an alternating current one. In this case, students can learn the basics of how current works by remotely testing different circuits. Students can control several switches remotely. The circuits include several lightbulbs, so, depending on the circuits that the students build, more or less current will go through each lightbulb. They can then visually appreciate the differences in light intensity through the webcam, and intuitively grasp how current works in an electrical circuit.

[14] https://www.raspberrypi.org/
[15] http://relle.ufsc.br/labs

IV. Technical and Organizational Pitfalls

This section summarizes the technical and organizational pitfalls that required to be addressed during the preparation of the laboratory to be commercially available.

A. Technical challenges

The first version of the remote laboratory matched all the basic functional requirements: students could send a program, the program was programmed in the Arduino device, the robot would move in the space. It was tested dozens of times successfully. Furthermore, as WebLab-Deusto had experience with other mobile robots, many common implementation problems were already identified:

- The energy cable needed to have certain length to cover the whole area.
- Energy cables tend to cause knots as the robot is used.
- Cameras and certain pieces are typically a problem where certain institution staff (e.g., cleaning staff) might move certain parts when nobody is present.
- Everything can break: cameras, routers, the Raspberry Pi, etc.

However, there are other types of errors that happen, especially after using it hundreds of times. In this particular case:

- After few hundreds of uses, the robot wheels get dust and have to be cleaned or replaced. Otherwise the speed reduces considerably.
- In few occasions, the line of the floor (a black tape) would be removed by the robot itself, causing errors and leaving the robot stuck.
- In some versions of the software, there were errors with the SD card, which if the robot fails to properly

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
08:00							
09:00			86				
10:00			271	25			1
11:00	48	1		257			2
12:00	158		2	4			
13:00			21	15	2		
14:00	2		25	2			
15:00			6	8			
16:00	14	2				2	
17:00	2	2				1	1
18:00							
19:00	1	1	6				
20:00	2		38				
21:00	4		1				
22:00							1
23:00							

Fig. 6. Use of the robot in one of the schools. Each slot is the sum of times it was used in a particular day of the week

reboot, the hardware can be left in a wrong state (e.g., crashed against the wall trying to continue, forcing the engines and causing a major malfunction of all the equipment).

B. Organizational challenges

There are different types of organizational challenges involved in the robot laboratory. In the first version of the remote laboratory, a single instance of the robot was available. And it was designed to be used as assignments (so students could use it at home mainly). However, teachers really preferred to use it in class while teaching, with 30 students at the same time. This created a queue which made the user experience decrease considerably. The problem was not easy to solve: the space required for a robot is considerably big, so having several copies could be a problem.

Also, while primarily using it in class, students and teachers also were using it during the evening and weekend (Figure 6 for one school, but multiple other schools were connecting during the weekend). This required LabsLand to implement complete automatization of the recovery of the robots, and having a strict control of the robots to detect potential errors.

As for the external laboratories, the problems are bigger: LabsLand cannot provide any quality service or fix issues when the equipment in not available. This is a major problem for being able to share laboratories from different institutions.

V. Solution

This section details what solutions were taking into place for addressing the technical challenges.

A. Addressing technical challenges

As the robot laboratory has been built on top of WebLab-Deusto , many common implementation problems were already solved by the RLMS. For example, WebLab-Deusto can be configured to automatically detect certain common errors: a device migth not be reachable through the network due to an error in the device or networking devices, a camera might not be serving proper images, etc.

Also, due to the previous experience as WebLab-Deusto , the energy cable was already connected to the ceiling with a

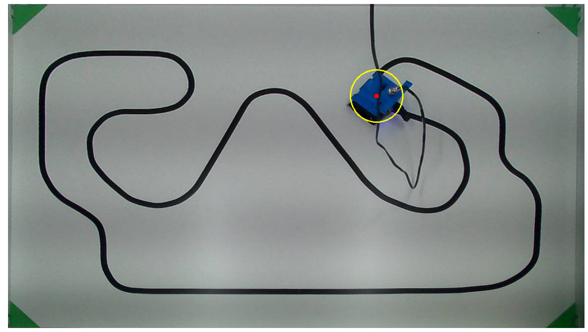

Fig. 7. Automatic detection of the robot while following the line to automatically check for failures

special device that allows it to rotate without causing knots. Also, both camera and the location were fixed and nobody had access, reducing problems of anyone moving by accident the camera and pointing somewhere else.

However, the main reason for the rest of the technical challenges was that it was required to have an advanced mechanism of automatic detection of problems. For one part of the problems (detecting if the program was properly programmed in the device), different internal mechanisms were implemented in the Raspberry Pi.

For the external problems, especially the ones that we would not be able to predict, it was necessary to count with a more advanced quality control mechanism. For this reason, LabsLand implemented a software that detected where the robot was in each moment by using computer vision (see Figure 7). With this mechanism, it was possible to periodically (every few hours) send a program that tells the robot to follow the line, and verify if the robot is indeed following the line or not and at what speed. With this data, it became possible to automatically detecting which robots were suffering more problems due to usage and start taking predictive maintenance for many of the problems that had been identified.

Furthermore, this approach also provided us to automatically know if a robot was not functioning correctly or not.

B. Addressing organizational challenges

In the case of the robot, automatically detecting problems allowed LabsLand to implement better policies related to the laboratory maintenance, avoiding certain problems before they happened. However, in certain occasions the problems can not be fixed: the ISP network might have a problem, or the robot or a device might physically break.

For this reason, the approach taken was to create multiple redundant copies of the laboratories, using both local and federated load balance of the robots. As seen in Figure 8, the left image depcts the new version of the robot where two robots are one on top of the other so as to save space. This way, two copies are located in LabsLand offices and other two copies are located in WebLab-Deusto . If there is a major technical issue in one of the two organizations, the system still works since the entry (a Cloud service) will still use the working one.

With these four copies, the laboratory experience increased considerably: given that students were most of the time using

Fig. 8. Robots deployed on LabsLand offices (Spain) -left- and UNAD (Colombia) -right-

the programming tools, and they had only 2 minutes to use the laboratory, the chances of five students using the equipment at the same time were considerably lower. In fact, with classes using the four laboratories at the same time, most of the times there was nobody using the four laboratories so students did not have to wait, and the maximum students had to wait to test their code was 6 minutes.

So as to improve the number of concurrent students, LabsLand is partnering UNAD (Universidad Nacional Abierta y a Distancia; Colombian main distance university) to create a new copy of the robotics laboratory. LabsLand plans to use the same approach with more laboratories, using the federated load balance to ensure that most of the times there are working copies of the laboratories used.

In the case of the external laboratories, however, this solution is not available, since the cost of acquiring many laboratories would be too high. For this reason, for those laboratories where the combinations are limited (e.g., few thousands), LabsLand has developed *ultra-concurrent laboratories*: LabsLand has run each laboratory through all the potential combinations for a number of times, recording all the potential values and videos. With this data, LabsLand has created a software interface that lets the user use the equipment as if the student was using the real-time laboratory, but displaying the proper recording. This approach increases to thousands the potential concurrent students given that it becomes a cloud laboratory. The results with this approach have been very successfully given that it becomes possible to have agreements with more universities to be added to the LabsLand network without maintenance.

VI. Conclusions

Remote laboratories have been used for decades, but their maintenance has always been a problem. And if the laboratories are not well maintained, the trust from potential consumers decreases. This is a critical challenge for one of the main focuses of remote laboratories: enabling different institutions to increase the experiential learning resources by accessing real experience from different institutions.

In this contribution, we have addressed a number of problems that have arisen during the distribution of laboratories through LabsLand, a remote laboratory network where consumers access remote laboratories provided by different institutions. Given that LabsLand sells this service, it must increase the state of the art of the quality of service of the remote laboratories offered. To do this, the contribution has explained technical solutions taken to tackle the problems addressed.

Among them, in the case of the Arduino robot, the contribution has explained the multiple mechanisms at different layers (computer vision, network problem detection, deployment of different copies of the laboratory in different countries) that make the laboratory reliable. In the case of the external laboratories, the contribution also detailed.

As for future work, LabsLand envisions a network of real laboratories that is available worldwide, connecting schools and universities with real equipment in different institutions, decreasing the access barrier to laboratories. In particular, LabsLand is already developing new laboratories which reuse many of the techniques presented in the paper to detect problems and has agreed with different universities in seven countries to develop new laboratories to put in the network.

References

[1] P. Orduña, P. Bailey, K. DeLong, D. López-de Ipiña, and J. García-Zubia, "Towards federated interoperable bridges for sharing educational remote laboratories," *Computers in Human Behavior*, vol. 30, pp. 389–395, Jan. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747563213001416

[2] J. P. S. Simão *et al.*, "Relle: Sistema de gerenciamento de experimentos remotos," 2016.

[3] L. Mellos Carlos, J. Bento da Silva, S. Sommer Bilessimo *et al.*, "Estratégias de integração de tecnologia no ensino: Uma solução baseada em experimentação remota móvel," 2017.

[4] V. J. Harward *et al.*, "The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories," *Proceedings of the IEEE*, vol. 96, no. 6, pp. 931–950, 2008.

[5] F. Schauer, M. Krbecek, P. Beno, M. Gerza, L. Palka, P. Spilakov, and L. Tkac, "Remlabnet iii – federated remote laboratory management system for university and secondary schools," in *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*. IEEE, 2016, pp. 238–241.

[6] P. Orduña, D. Garbi Zutin, S. Govaerts, I. Lequerica Zorrozua, P. H. Bailey, E. Sancristobal, C. Salzmann, L. Rodriguez-Gil, K. DeLong, D. Gillet *et al.*, "An extensible architecture for the integration of remote and virtual laboratories in public learning tools," *Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de*, vol. 10, no. 4, pp. 223–233, 2015.

[7] T. de Jong, M. C. Linn, and Z. C. Zacharia, "Physical and virtual laboratories in science and engineering education," *Science*, vol. 340, no. 6130, pp. 305–308, 2013.

[8] D. Gillet, T. de Jong, S. Sotirou, and C. Salzmann, "Personalised learning spaces and federated online labs for stem education at school," in *Global Engineering Education Conference (EDUCON), 2013 IEEE*. IEEE, 2013, pp. 769–773.

[9] M. Ishutkina, E. Feron, M. Casini, and A. Vicino, "An internet based laboratory for control of a safety critical system," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 3. IEEE, 2004, pp. 2707–2712.

[10] O. Dziabenko, J. García-Zubia, and I. Angulo, "Time to play with a microcontroller managed mobile bot," in *Global Engineering Education Conference (EDUCON), 2012 IEEE*. IEEE, 2012, pp. 1–5.

[11] L. Gomes and S. Bogosyan, "Current trends in remote laboratories," *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 12, pp. 4744–4756, 2009.

[12] C. Gravier, J. Fayolle, B. Bayard, M. Ates, and J. Lardon, "State of the art about remote laboratories paradigms-foundations of ongoing mutations," *iJOE*, vol. 4, no. 1, 2008.

[13] B. Carisa, A. Burain, S. Molly H, and C. Lawrence, "Running control engineering experiments over the internet," 1995.

[14] B. Aktan, C. Bohus, L. Crowl, and M. Shor, "Distance learning applied to control engineering laboratories," *Education, IEEE Transactions on*, vol. 39, no. 3, pp. 320–326, 1996.

[15] J. Henry, "Running laboratory experiments via the world wide web," in *ASEE Annual Conference*, 1996.

[16] A. Coble, A. Smallbone, A. Bhave, R. Watson, A. Braumann, and M. Kraft, "Delivering authentic experiences for engineering students and professionals through e-labs," in *Education Engineering (EDUCON), 2010 IEEE*. IEEE, 2010, pp. 1085–1090.

[17] R. Cedazo, F. Sanchez, J. Sebastian, A. Martínez, A. Pinazo, B. Barros, and T. Read, "Ciclope chemical: a remote laboratory to control a spectrograph," *Advances in Control Education–ACE*, vol. 6, 2006.

[18] J. Del Alamo, L. Brooks, C. McLean, J. Hardison, G. Mishuris, V. Chang, and L. Hui, "The mit microelectronics weblab: A web-enabled remote laboratory for microelectronic device characterization," in *World Congress on Networked Learning in a Global Environment, Berlin (Germany)*, 2002.

[19] D. Gillet, H. Latchman, C. Salzmann, and O. Crisalle, "Hands-on laboratory experiments in flexible and distance learning," *Journal of Engineering Education*, vol. 90, no. 2, pp. 187–191, 2001.

[20] I. Gustavsson, J. Zackrisson, L. Håkansson, I. Claesson, and T. Lagö, "The visir project–an open source software initiative for distributed online laboratories," in *Proceedings of the REV 2007 Conference, Porto, Portugal*, 2007.

[21] Z. Nedic, J. Machotka, and A. Nafalski, "Remote laboratory netlab for effective interaction with real equipment over the internet," in *Human System Interactions, 2008 Conference on*. IEEE, 2008, pp. 846–851.

[22] R. Safaric, M. Truntič, D. Hercog, and G. Pačnik, "Control and robotics remote laboratory for engineering education," *International Journal of Online Engineering (iJOE)*, vol. 1, no. 1, 2005.

[23] F. Torres, F. Candelas, S. Puente, J. Pomares, P. Gil, and F. Ortiz, "Experiences with virtual environment and remote laboratory for teaching and learning robotics at the university of alicante," *International Journal of Engineering Education*, vol. 22, no. 4, pp. 766–776, 2006.

[24] J. Hardison, K. DeLong, P. Bailey, and V. Harward, "Deploying interactive remote labs using the ilab shared architecture," in *Frontiers in Education Conference, 2008. FIE 2008. 38th Annual*. IEEE, 2008, pp. S2A–1.

[25] P. Orduña, A. Almeida, S. Ros, D. Lpez-de Ipiña, and J. García-Zubia, "Leveraging Non-explicit Social Communities for Learning Analytics in Mobile Remote Laboratories," *Journal of Universal Computer Science*, vol. 20, no. 15, pp. 2043–2053, Dec. 2014, 00000.

[26] P. Orduña, "Transitive and scalable federation model for remote laboratories," Ph.D. dissertation, Universidad de Deusto, Bilbao, Spain, May 2013. [Online]. Available: http://paginaspersonales.deusto.es/porduna/phd/

[27] Z. Nedic and J. F. Machotka, "Remote laboratory netlab for effective teaching of 1st year engineering students," *International Journal of Online Engineering (iJOE)*, vol. 3, no. 3, 2007.

[28] P. Orduña and J. García-Zubia, "Scheduling schemes among internet laboratories ecosystems," *REV 2011*, 2011.

[29] D. Lowe and N. Orou, "Interdependence of booking and queuing in remote laboratory scheduling," in *Remote engineering and virtual instrumentation (rev), 2012 9th international conference on*. IEEE, 2012, pp. 1–6.

[30] P. Orduña, P. H. Bailey, K. DeLong, D. López-de Ipiña, and J. García-Zubia, "Towards federated interoperable bridges for sharing educational remote laboratories," *Computers in Human Behavior*, vol. 30, pp. 389–395, 2014.

[31] N. R. Council *et al.*, *America's lab report: Investigations in high school science*. National Academies Press, 2006.

[32] U. européenne. Commission européenne and U. européenne. Direction générale de la recherche, *Science education now: A renewed pedagogy for the future of Europe*. Office for Official Publications of the European Communities, 2007.

[33] M. D. Merrill, "First principles of instruction," *Educational technology research and development*, vol. 50, no. 3, pp. 43–59, 2002.

[34] J. F. Roscoe, S. Fearn, and E. Posey, "Teaching computational thinking by playing games and building robots," in *Interactive Technologies and Games (iTAG), 2014 International Conference on*. IEEE, 2014, pp. 9–12.

[35] J. Trower and J. Gray, "Blockly language creation and applications: Visual programming for media computation and bluetooth robotics control," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*. ACM, 2015, pp. 5–5.

[36] L. Rodríguez-Gil, J. García-Zubia, P. Orduña, and D. López-de Ipiña, "An open and scalable web-based interactive live-streaming architecture: The wilsp platform," *IEEE Access*, vol. 5, pp. 9842–9856, 2017.