

Development of a Low-Cost Remote Lab Concept for Electronic Engineering Education Based on NI myDAQ and NI ELVIS

<https://doi.org/10.3991/ijoe.v16i14.17007>

Christian Kreiter ^(✉), Thomas Klinger
Carinthia University of Applied Sciences, Villach, Austria
c.kreiter@fh-kaernten.at

Abstract—Lab work and exercises are an essential part of Electronic Engineering Education as it improves understanding of the theoretical concepts. Remote Labs like VISIR (Virtual Instrument Systems in Reality) can supplement the learning process but are limited to a small set of components. Therefore, experiments with VISIR should be combined with prepared and fixed circuits.

This work presents an approach, where in the first step new exercises are developed with the NI ELVIS platform, and later implemented with the much more cost-effective NI myDAQ platform. In general, the entire system is very inexpensive and scaleable, since a single PC can act as a host for a wide number of exercise boards, each of which is connected via a myDAQ.

Keywords—Remote Labs, Online Labs, Engineering Education, Remote Experiments

1 Introduction

In some countries, distance learning was already an essential part of post-secondary education even before the Covid-19 pandemic. For example, in 2018 16.6% of American students were exclusively taking distance education courses [1]. The sudden lockdown situation led to a forced transition towards distance education. Standard tuition, where the students, for the most part, play a passive role, can be migrated to online classrooms more easily. In engineering education (or more generally speaking natural sciences) however, where lab work is a crucial factor for the understanding of the matter, a substitution is not as simple and requires a lot of effort.

For decades, our university offers most study programs in a regular and an evening (also called work-friendly) variant. A few years ago we started to implement Online Labs, like VISIR (Virtual Instrument Systems in Reality) [2],[3],[4], and Pocket Labs using the NI myDAQ [5] into our electrical engineering education. Previous surveys have shown that especially amongst evening students the flexible time management, which is made possible by the Pocket Labs, is well-received (see Q2 of Fig. 1).

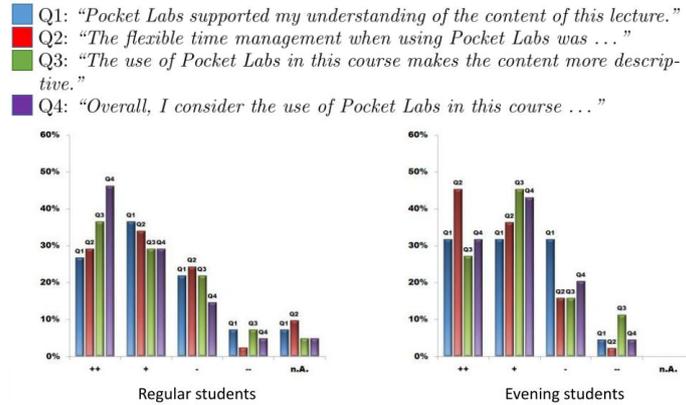


Fig. 1. Survey results on the use of Pocket Labs [5]

Due to Covid-19, the role of Online and Pocket Labs became significantly more important. Remote Laboratory systems like VISIR can be an extremely helpful utility for engineering education, especially in part-time study programs, where time and flexibility of students is limited. However, the number of components that can be used in each experiment limits the functionality of such systems. A helpful approach to expand the number of exercises (e.g. for electronic engineering students) could be to use virtual breadboarding systems like VISIR and combine it with prepared and fixed circuits. With VISIR, students are able to construct and/or modify circuits [6] but are limited to only a small number of components. Prepared and fixed circuits, however, allow for experiments that are more complex.

2 Principle of Development

Complex experiments require diligent and precise circuit development, to ensure reliable functionality over time. Additionally, to simulation systems like PSpice, LT Spice, or Multisim, a prototyping system like NI ELVIS is highly recommended (see Fig. 2).

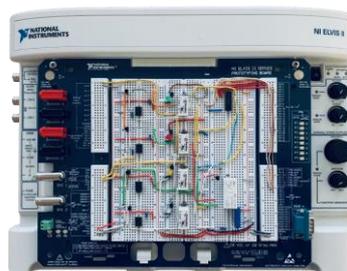


Fig. 2. Developing the prototype circuit with NI ELVIS. The experiment shown here is the single-quadrant multiplier

On the other hand, for engineering education, it is necessary to provide many or at least several exercises to the students, so that a broad variety of topics can be covered. Therefore, the entire Remote Lab system should consist of multiple hardware units with interfacing capability to exercise boards. Although the ELVIS development system satisfies these requirements, it is not applicable to use it in such systems due to its high price.

Following these presumptions, we decided on a different solution. The NI myDAQ is a cost-efficient hardware platform, which is frequently used at our University for Pocket Labs [7], can be equipped with extension boards, and, most important, is fully compatible with NI ELVIS regarding in- and outputs and the necessary measurement specifications. The now cost-efficient entire system can be seen in Fig. 3, whereas Fig. 4 shows the myDAQ together with an experiment board.

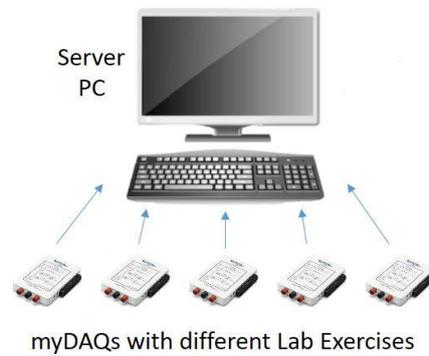


Fig. 3. Remote-Lab system using the cost-efficient NI myDAQ instead of NI ELVIS. The myDAQs are connected to a single PC acting as a measurement server.

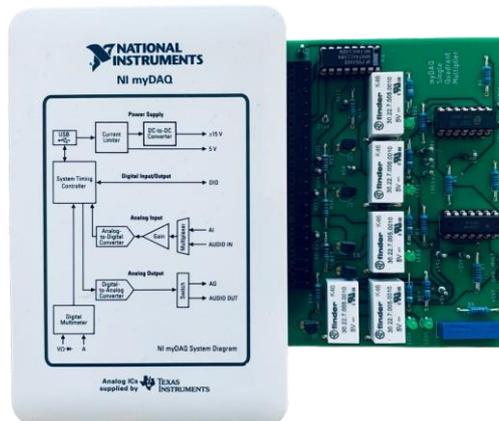


Fig. 4. Final lab PCB connected to a NI myDAQ. The experiment is again the single-quadrant multiplier

The experiment boards can be created on a PCB after intensive testing of the circuit using the breadboard of NI ELVIS. As the hardware control software runs under LabVIEW™, the measurement device can easily be changed to a compatible one, which allows for the later use of a cost-efficient measurement device such as the myDAQ.

3 Three Examples of Remote Lab Experiments

To explain the possible functionality of the mentioned laboratory exercises, two examples are now shown. The first one is a board for measurements on operational amplifiers (Op Amps), which is shown on the right of Fig. 5, and the other one is an analog multiplication circuit for positive values, therefore also known as a single-quadrant multiplier (left board in Fig. 5).

3.1 Experiments with operational amplifiers

The first board enables the students to perform experiments with different operational amplifier circuits, which are provided in functional pairs:

- Inverting and non-inverting amplifiers
- Summing and differential amplifiers
- Differentiator and integrator
- Logarithmic and anti-logarithmic amplifiers
- Comparator and variable-gain amplifiers

The various circuit configurations can be selected through the digital port of the myDAQ.

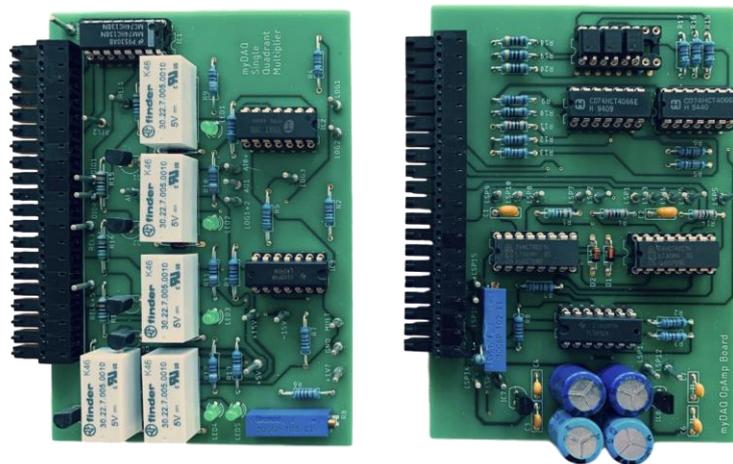


Fig. 5. Single-Quadrant Multiplier Board (left) and Operational Amplifier Board (right)

3.2 Single-quadrant multiplier

The second board developed shows the functionality of an analog multiplication circuit. The principle is based on analog multiplication making use of the fact that

$$\frac{xy}{z} = \exp(\ln x + \ln y - \ln z) \quad (1)$$

The entire circuit can be seen in Fig. 6; it realizes the log and exponential (anti-log) function with operational amplifiers and the exponential characteristic of bipolar transistors. One single logarithmic amplifier (e.g. the one in the bottom left corner) has the transfer function with V_T as the temperature voltage and I_S as the saturation current of the transistor.

$$V_a = -V_T \ln \frac{V_2}{I_S \cdot R_6} \quad , \quad (2)$$

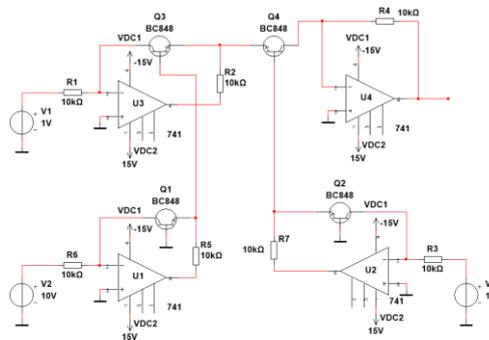


Fig. 6. Simulation circuit of the single-quadrant multiplier in MultisimTM [9]

The next logarithmic amplifier (top left in Fig. 6) adds this resulting voltage to its transfer function, and the one in bottom right subtracts its output voltage from this sum, which leads to the very useful fact, that I_S as transistor parameter and V_T as temperature influence can be canceled out if the following two requirements are fulfilled:

- The four transistors have to be of the same (ideally: selected) type.
- The four transistors are operating at the same ambient temperature.

Both requirements can be met if an integrated set of transistors, e.g. a THAT 300 from THATCorp is chosen, resulting in (1), which shows no influence of temperature or transistor characteristics [9].

3.3 Analog-to-digital converter experiments

The third board developed contains basic exercises with feedback ADCs. The analog-to-digital converter circuit function according to the procedure described in the principle shown in Fig. 7. The board provides three different logic modules:

- A ramp ADC
- A tracking ADC
- An ADC with Successive Approximation Register (SAR)

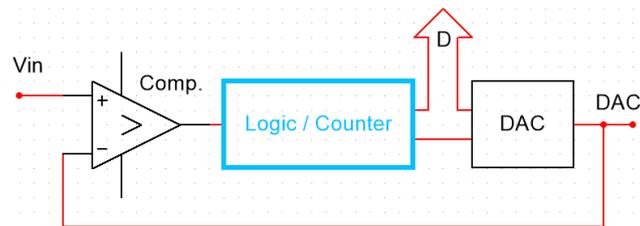


Fig. 7. Analog-to-digital converter principle

The client allows to set the logic module, as well as the amplitude, offset and waveform type of the input signal. The exercises include determination of the logic modules' clock frequency, the resolution of the ADC, minimum and maximum conversion time of the converters and the upper limit of the input signal change (in V/ms) where the converter still can work reliably.

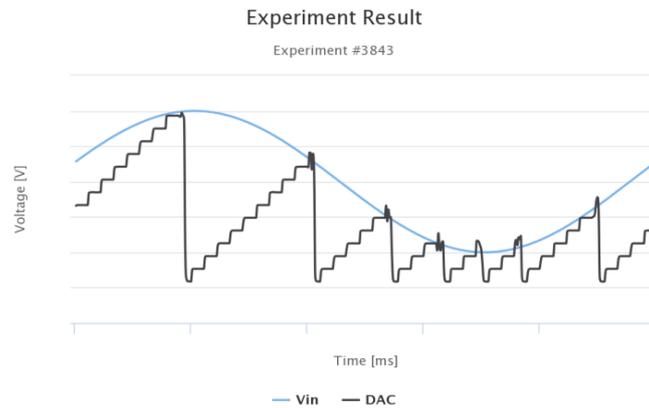


Fig. 8. Example of an experiment result (Ramp ADC)

4 Software Architecture

The Remote Lab makes use of the Experiment Dispatcher - a software architecture that provides ubiquitous deployment of Online Laboratories [10]. The Lab Infrastructure as a Service (LIaaS) paradigm facilitates the deployment of the Remote Lab by providing a set of Web service calls and recording the experiments. As seen in Fig. 9 the Dispatcher supports multiple RLMS systems [11], namely the MIT iLab Shared Architecture [12], the University of Queensland new implementation of the iLab Shared Architecture [13] and the WebLab Deusto from University of Deusto [14]. A

web client can either directly connect to a Virtual Lab Server instance provided by the Dispatcher, or connect via one of the aforementioned RLMS systems.

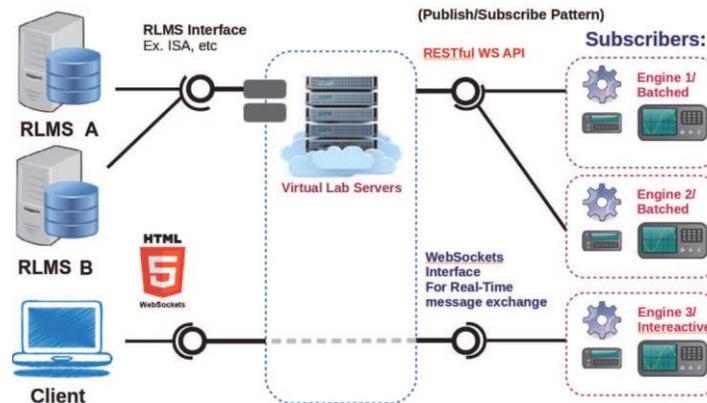


Fig. 9. Lab Infrastructure as a Service (LIaaS) [11]

The interface used for this Remote Lab is the iLab Batched Middleware [12]. Remote Labs can be categorized into two types – batched and interactive labs. Batched labs offer their services to multiple users simultaneously. This is achieved by setting all parameters of an experiment prior to its execution. Once parameters are set the user submits them to the Dispatcher. The experiment with its set parameters is put on a batch and the experiments are executed one after another according to the FIFO principle (first in, first out). Interactive experiments, on the other hand, allow for a change of parameters in real-time. However, the main disadvantages are that it usually can handle only one user at a time and it requires a scheduling system.

An Experiment Engine (or Subscriber Engine) which connects to the Dispatcher over its RESTful Web service API controls the lab hardware. The engine is completely language-agnostic. Implementations in LabVIEW, Matlab, Python, PHP and Java have been developed.

4.1 The web client

The Web client is programmed in HTML and JavaScript to work with all devices, from smartphones to standard PCs, and it can be directly started from the Moodle learning platform.

The client provides a generic structure for electronic experiments. On startup, a lab configuration in JSON format is loaded from the Dispatcher. It contains up-to-date information about the connected myDAQ boards, their circuits (called setups) and their inputs, which have to be set by the user. Thus, it allows adding new experiment boards and setups with little to no additional programming effort.

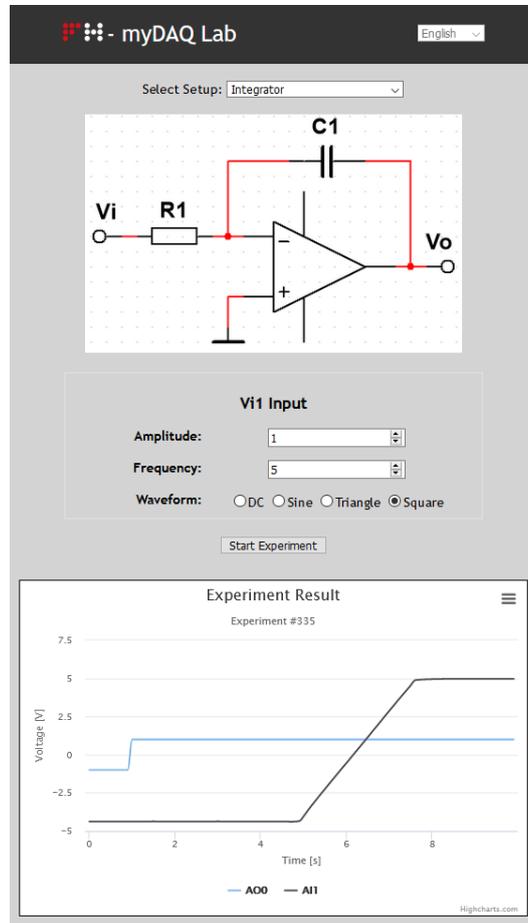


Fig. 10. Web client with experiment result

With the provided lab configuration, the Web client generates an interface as seen in Fig. 10. From top to bottom, the interface consists of:

- Title
- Setup select menu
- Circuit schematic
- Input parameter(s)
- Start Experiment button; and finally
- Experiment results

An experiment setup can have one or multiple inputs, such as a DC voltage supply, a function generator or a relay represented as a toggle switch. Each input has to be defined with its components and ranges. The following snippet shows the function generator input "vi1" (used in operational amplifier experiments) with an amplitude

range from -5V to +5V, a frequency range from 0Hz to 100Hz and the waveform options "DC", "Sine", "Triangle" and "Square".

```
"v1": {
  "type": "FGEN",
  "label": "V1 Input",
  "attr": {
    "ampl": {"min": -5, "max": 5, "def": 0},
    "freq": {"min": 0, "max": 100, "def": 0},
    "waveform": ["DC", "Sine", "Triangle", "Square"]
  }
}
```

For each setup, the web client requires a label for the select menu, an image of the circuit schematic and a list of the used inputs. The following snippet shows the "integrator" setup:

```
"integrator": {
  "label": "Integrator",
  "image": "integrator.png",
  "input": ["v1"]
}
```

By pressing the "Start Experiment" button the Web client generates an Experiment Specification (JSON) which is saved at the Dispatcher. The client then waits for the Experiment Engine to process the batch. The status is updated by a continuous polling process. When the status is set to "completed" the Experiment Results are pulled from the Dispatcher and displayed either as text or as graphic, depending on the experiment.

4.2 The experiment engine

The Experiment Engine handles the incoming Experiment Specifications from the Web client, as well as the experiment execution and returns the Experiment Results to the Dispatcher. It is implemented as a finite-state machine in LabVIEW as shown in Fig. 11.

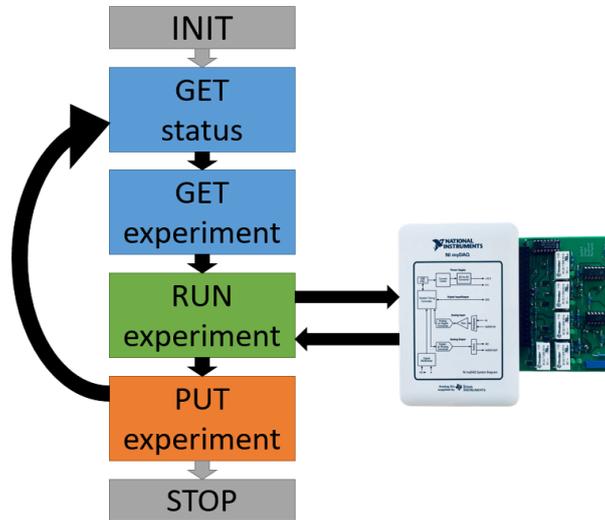


Fig. 11. LabVIEW Experiment Engine

List of Experiment Engine states:

- INIT: Loads configuration data, such as Dispatcher URL and authentication credentials, from a file.
- GET status: Checks if a new experiment is available.
- GET experiment: Retrieves the Experiment Specification from the Dispatcher.
- RUN experiment: Selects the myDAQ board and runs the VI, which is saved in the wrapper.
- PUT experiment: Returns the Experiment Result to the Dispatcher.
- STOP: In case of a critical error or manual abortion by the lab owner.

The states where the name starts with "GET" or "PUT" are named after the REST-API methods of the Dispatcher. E.g., "GET status" sends an HTTP GET request to the Dispatcher asking for the status of new experiments. The state "RUN experiment" checks first which board was requested in the Experiment Specification and then loads the respective wrapper, which parses the whole JSON message and runs the actual experiment. With this approach, multiple myDAQ boards can be handled with a single LabVIEW instance. A new experiment board is nothing more than an additional Sub-VI with the parameters as input and the measurement results as output.

Since the device name of a myDAQ depends on the order in which the myDAQs are connected to the system, a configuration file lists all active experiment boards and the serial number of the myDAQ it is connected to. The Experiment Engine reads out the serial number of each myDAQ and compares it to the configuration file.

5 Conclusion

The three experiments presented are expanding the number of remote exercises that can be used for electrical and electronic engineering education, and therefore are a perfect supplement for other systems like VISIR. Moreover, exercises based on this principle can be developed easily with the NI ELVIS platform, and later implemented on the much cheaper NI myDAQ. Since a single lab PC can handle multiple exercise boards, the whole setup is inexpensive. The software architecture based on Lab Infrastructure as a Service (LIaaS) paradigm further simplifies the development of additional boards. New experiments are implemented via the lab configuration settings that the clients request at startup. Thus, no adaptations to the web client need to be made.

First face-to-face feedback from students indicates that the experiment boards are well received by our students. We plan to collect reactions and feedback in the form of surveys in the near future. Additionally, we continue to develop new experiment boards.

6 References

- [1] Digest of Education Statistics, “Number and percentage of students enrolled in degree-granting postsecondary institutions, by distance education participation, location of student, level of enrollment, and control and level of institution: Fall 2017 and fall 2018,” National Center for Education Statistics, 2019. [Online]. Available: https://nces.ed.gov/programs/digest/d19/tables/dt19_311.15.asp. [Accessed: 10-Jul-2020]. <https://doi.org/10.3886/icpsr02447>
- [2] I. Gustavsson et al., “A Flexible Electronics Laboratory with Local and Remote Workbenches in a Grid,” *International Journal of Online Engineering (iJOE)*, vol. 4, no. 2, pp. 12-16, 2008.
- [3] M. C. Viegas et al., “The VISIR+ Project — Preliminary results of the training actions,” in *Online Engineering & Internet of Things. Lecture Notes in Networks and Systems*, Michael E. Auer and D. G. Zutin, Eds. New York: Springer, Cham, 2018, pp. 375-391. https://doi.org/10.1007/978-3-319-64352-6_36
- [4] L. C. M. Schlichting, G. S. Ferreira, D. D. de Bona, F. de Faveri, and J. A. Anderson, “Remote Laboratory: Application and usability,” *Conference on Applied Technologies for Electronic Teaching (TAEE)*, Sevilla, Spain, 2016. <https://doi.org/10.1109/taee.2016.7528355>
- [5] T. Klinger, C. Kreiter, “Experiences with the Use of Pocket Labs in Engineering Education,” in *Teaching and Learning in a Digital World. ICL 2017. Advances in Intelligent Systems and Computing*, M. E. Auer, D. Guralnick, and I. Simonics, Eds. Springer, Cham, 2018, pp. 665-670. https://doi.org/10.1007/978-3-319-73204-6_72
- [6] N. Lima, C. Viegas, G. R. Alves, F. J. Garcia-Peñalvo, “VISIR’s Usage as an Educational Resource: a Review of the Empirical Research,” *Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM)*, Salamanca, Spain, November 2016. <https://doi.org/10.1145/3012430.3012623>
- [7] T. Klinger, D. Garbi Zutin, C. Madritsch, “Parallel Use of Remote Labs and Pocket Labs in Engineering Education,” *International Conference on Remote Engineering and Virtual*

- Instrumentation (REV), New York City, USA, March 2017. https://doi.org/10.1007/978-3-319-64352-6_42
- [8] S. Malekar and R. Arts, "Opamp Experimenting Kit for myDAQ," *Elektor Magazine*, pp. 105–109, May & June 2016.
- [9] U. Tietze and C. Schenk, *Electronic Circuits*. 2nd Ed: Springer, Berlin, 2015.
- [10] D. Garbi Zutin and C. Kreiter, "A Software Architecture to Support an Ubiquitous Delivery of Online Laboratories," *International Conference on Interactive Mobile Communication, Technologies and Learning (IMCL)*, San Diego, USA, October 2016. <https://doi.org/10.1109/imctl.2016.7753779>
- [11] D. G. Zutin, M. Auer, P. Orduña and C. Kreiter, "Online lab infrastructure as a service: A new paradigm to simplify the development and deployment of online labs," *2016 13th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, Madrid, 2016. <https://doi.org/10.1109/rev.2016.7444467>
- [12] V. J. Harward et al., "The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories," in *Proceedings of the IEEE*, vol. 96, no. 6, pp. 931-950, June 2008. <https://doi.org/10.1109/jproc.2008.921607>
- [13] S. Colbran and M. Schulz, "An update to the software architecture of the iLab Service Broker," *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, Bangkok, 2015, pp. 90-93, <https://doi.org/10.1109/rev.2015.7087269>.
- [14] P. Orduña, "Transitive and Scalable Federation Model for Remote Laboratories," *Doctoral Thesis*, University of Deusto, Bilbao, Spain, Apr. 2013. [Online]. Available: <http://morelab.deusto.es/media/publications/theses/pablo-orduna.pdf> [Accessed: 10-Jul-2020]

7 Authors

Christian Kreiter has a BSc degree in Systems Engineering and an MSc degree in Systems Design from Carinthia University of Applied Sciences (CUAS). He presently works at CUAS as junior researcher and lecturer. His research interest is about Online Engineering. Email: c.kreiter@fh-kaernten.at

Thomas Klinger studied Electrical Engineering at the University of Technology Vienna and worked there as Assistant Lecturer for three years while working on his PhD thesis. After that, he worked for seven years at Philips in Klagenfurt, Austria, holding positions as a development engineer, and, in the last years, development manager. Since 1998, he is a professor at CUAS; in the first years teaching System Integration. He has been Dean of the Engineering & IT department for about 10 years and was elected twice for the position of CUAS' Vice-Rector. Currently, he owns no higher position at CUAS and teaches Electrical and Electronic Engineering and Image Processing. Email: t.klinger@fh-kaernten.at

Article submitted 2020-07-13. Resubmitted 2020-08-13. Final acceptance 2020-08-14. Final version published as submitted by the authors.