

# Reconfigurable IEEE1451-FPGA based weblab infrastructure

Ricardo J. Costa<sup>12</sup>, Gustavo R. Alves<sup>1</sup>, Mário Zenha-Rela<sup>2</sup>  
ISEP/CIETI/LABORIS<sup>1</sup>, FCTUC/CISUC<sup>2</sup>  
rjc@isep.ipp.pt; gca@isep.ipp.pt; mzrela@dei.uc.pt

**Abstract** - Weblabs are spreading their influence in Science and Engineering (S&E) courses providing a way to remotely conduct real experiments. Typically, they are implemented by different architectures and infrastructures supported by Instruments and Modules (I&Ms) able to be remotely controlled and observed. Besides the inexistence of a standard solution for implementing weblabs, their reconfiguration is limited to a setup procedure that enables interconnecting a set of preselected I&Ms into an Experiment Under Test (EUT). Moreover, those I&Ms are not able to be replicated or shared by different weblab infrastructures, since they are usually based on hardware platforms. Thus, to overcome these limitations, this paper proposes a standard solution that uses I&Ms embedded into Field-Programmable Gate Array (FPGAs) devices. It is presented an architecture based on the IEEE1451.0 Std. supported by a FPGA-based weblab infrastructure able to be remotely reconfigured with I&Ms, described through standard Hardware Description Language (HDL) files, using a Reconfiguration Tool (RecTool).

**Index Terms** - Weblabs, Remote Labs, Reconfiguration, FPGA, IEEE1451.0 Std..

## I. INTRODUCTION

The permanent discoveries in S&E domains require ways of spreading them through the society. In this process, education takes an important role, becoming fundamental to create good teaching and learning methods so future scientists and engineers can encompass the evolution promoted by those discoveries. This is a key issue that can be fulfilled by enabling new ways of providing the required laboratorial work, which is fundamental so students can understand, validate and eventually questioning learned theories. Only with critical attitudes students may construct their knowledge in a specific domain, and the laboratorial work, focused on the interaction with real equipment with the inherent unpredictably of the obtained results, is a key component that must be provided in S&E courses [1][2][3][4]. However, the evolution of knowledge on S&E domains, the limited resources and time constraints of the courses' curricula, impair that all laboratorial work may be provided using traditional laboratories. To handle these issues, the advantages brought by new technological solutions, especially by the possibility of networking the I&Ms usually adopted in every laboratory, promoted the appearance of weblabs, which allow remote conducting real experiments through the Internet. Current trends show their widespread in education [5] as important resources used to complement and, in some cases, replace traditional laboratories [6][7], supported by several international

consortiums and projects, some of them currently active, as illustrated in table I.

TABLE I. WEBLABS' RELATED CONSORTIUMS AND PROJECTS.

<b>Lab2go:</b> Created by the Carinthia University of Applied Sciences Villach, Austria. It is an online web portal repository where developers can describe their own weblabs using a predefined Online Laboratory Metadata - Reference Model Specification. Further information available on <a href="http://www.lab2go.net/">http://www.lab2go.net/</a> .
<b>iLab Project:</b> Created by the Massachusetts Institute of Technology, United States of America. It provides a scalable and decentralized web service infrastructure named iLab Shared Architecture (ISA) that enables inter-connecting different weblabs. Further information available on <a href="https://wikis.mit.edu/confluence/display/ILAB2/about+iLabs">https://wikis.mit.edu/confluence/display/ILAB2/about+iLabs</a> .
<b>Labshare:</b> Consortium, since 2011 an Institute, composed by several Australian universities. It aims to provide a set of services for the integration and development of remote experiments in Australia. Further information available on <a href="http://www.labshare.edu.au/">http://www.labshare.edu.au/</a> .
<b>VISIR (Virtual Instrument Systems In Reality):</b> Open weblab platform created by the Blekinge Institute of Technology, Sweden, for running and managing remote experiments. Further information available on <a href="http://openlabs.bth.se/">http://openlabs.bth.se/</a> .
<b>GOLC (Global Online Laboratory Consortium):</b> Consortium that aims to define standard solutions for creation of sharable, online experimental environments. Further information available on <a href="http://online-lab.org/">http://online-lab.org/</a> .
<b>LiLa (Library of Labs):</b> Consortium headed by the Stuttgart University that provides an organizational framework for the exchange of experiments between institutions. Further information available on <a href="http://www.lila-project.org/">http://www.lila-project.org/</a> .

Despite the advantages of using weblabs [3], the Accreditation Board for Engineering and Technology (ABET) [8][9][10], indicates as fundamental for a successful learning in S&E courses the design skills acquired by students, which focus on the design, build, or assemble instruments and systems using specific methodologies, equipments, or materials. Although some weblabs already allow reconfiguring the infrastructure using a setup procedure that enables connecting preselected I&Ms into different locations of an EUT (e.g. [11]) (this partially fulfilling the design skills requirements) they are not able to be reconfigured with different I&Ms, as in traditional laboratories where users may select all type of available I&Ms. Furthermore, the I&Ms may be expensive and/or inaccessible, not persuading the adoption of weblabs in some courses, especially in countries with economical restrictions. In these countries, it becomes even more important an institutional collaboration, not only by providing remote access to experiments, eventually created by highly reputed institutions in a specific domain, but also by facilitating I&Ms' sharing according to the requirements of each experiment. This last aspect is not handled by current weblab infrastructures, because they adopt traditional I&Ms based on hardware platforms, impossible to be shared and integrated in other remotely located weblabs. Additionally, weblabs traditionally follow non-standard architectures requiring specialized developers and technicians to create and setup remote experiments,

which may difficult their adopting in some courses. Therefore, to handle all these issues, we have implemented a weblab architecture based on the IEEE1451.0 Std.[12] supported by a FPGA-based weblab infrastructure able to be remotely reconfigured with different and sharable I&Ms, described through standard HDLs files, using a RecTool.

Section II of this paper explores the importance of standardization and reconfiguration, presenting the advantages of using an architecture based on the IEEE1451.0 Std. and FPGA technology to create weblab infrastructures. Section III describes the proposed architecture and section IV details the implemented RecTool, presenting its interface, utilization sequence and implementation issues. Before conclusions, section V briefly demonstrates the use of the RecTool to reconfigure the weblab with two digital I/O modules and one module to control a bipolar step-motor.

## II. STANDARDIZATION & RECONFIGURATION USING THE IEEE1451.0 STD. AND FPGAS

Weblab infrastructure's standardization can be partially fulfilled if all adopted I&Ms follow standard access solutions. Most of them already integrate standard solutions, maintained by the IVI foundation [13], to facilitate users' interaction through computer mediation. An example is the adoption of the Standard Commands for Programmable Instrumentation (SCPI) that provides a set of textual command formats to control I&Ms, which is currently complemented by other solutions that focus on high level software architectures. The most known is the Interchangeable Virtual Instruments (IVI) drivers that export a standard set of commands for a limited class of instruments, and the Virtual Instrument Software Architecture (VISA) used for configuring, programming, and troubleshooting instrumentation systems using particular interfaces like GPIB, VXI, PXI, Serial, Ethernet, and/or USB interfaces. Although SCPI, IVI and VISA solutions can be interesting for controlling some I&Ms integrated in every weblab infrastructure, they are essentially focused on the software architecture. Moreover, they are not versatile enough to control all type of I&Ms that may be required by a particular weblab infrastructure. For example, both the SCPI and the IVI only provides commands and classes for specific instruments, while the VISA, which is usually used with the IVI and SCPI, requires the use of particular interfaces. Additionally, none of them provides an architecture to network-interface the I&Ms, neither a way to provide information about each I&M, which can be useful if remote users wish to decide for their adoption in a particular experiment.

The relevance of these, and other issues, namely the standardization required for weblab infrastructures, is being considered by the research community, through a recent created consortium named Global Online Laboratory Consortium (GOLC) [14]. Its members are focused on creating a standard solution for accessing and sharing weblab resources. They are currently active on creating interface definitions, by specifying a set of profiles to interact with the laboratory apparatus (that include I&Ms) across the internet. The interface, currently on a draft proposal, provides APIs both for batch and interactive experiments, so users may control several issues of a weblab architecture namely the access,

experiment status, booking and queuing management, and others. Despite not referring the reconfiguration concept, the high level of abstraction proposed in the current version of the proposed APIs, may, in the future, integrate a profile for reconfiguring weblab infrastructures with different I&Ms. However, GOLC members are still focused on software architectures, underestimating the advantages of reconfiguring weblabs using I&Ms described through HDLs (named embedded I&Ms), which may require further standardization, namely in the I&Ms' design and integration into weblab infrastructures.

Therefore, in spite of the advantages brought by current weblabs for education, and the technical solutions proposed by GOLC, there are still open aspects that should be improved and specially handled, namely:

- i) enable weblabs' reconfiguration with different I&Ms, and not only setting up connections;
- ii) facilitate sharing I&Ms to increase collaboration among institutions and decrease the associated costs;
- iii) provide a standard approach to create a versatile solution to design, remotely access and integrate those I&Ms into infrastructures, according to experiments needs.

Thus, the big issue is to create a standard weblab infrastructure able to be reconfigured. The IEEE1451.0 Std. [12] is a possible solution that, used in conjunction with FPGA technology, already viewed as an interesting solution for embedding I&Ms [15][16][17], may satisfy all the enumerated aspects.

The IEEE1451.0 Std. provides a common HTTP API to control transducers, which may form the I&Ms required in a weblab infrastructure, and defines a network-architecture based on two modules named Transducer Interface Module (TIM) and Network Capable Application Processor (NCAP). Using different interface types (e.g. RS-232), this last module bridges the access to one or more TIMs that interfaces the EUT using transducer modules accessed through one or more Transducer Channels (TCs). To organize information about features and states of each transducer, and eventually of the entire architecture [18], the standard gathers all that information in an organized way using the so-called Transducer Electronic Data Sheets (TEDSs).

The advantage of using FPGA technology focus on its reconfigurable nature and on its ability of embedding modules described through standard HDLs like Verilog or VHDL. This way, the description of I&Ms using these languages enables them to be embedded into FPGAs of different manufacturers, which justifies the denomination of embedded I&Ms. Moreover, if those I&Ms were described according to the IEEE1451.0 Std., they will be easily created and controlled using specific APIs, contributing for an universal solution similar to a worldwide workbench approach where users may select I&Ms and reconfigure the weblab like in the traditional laboratories, with the extended capability of using any number of I&Ms only limited by the capacity and interfaces available in a selected FPGA-based board.

To prove the presented ideas that focus on creating a reconfigurable weblab infrastructure supported by FPGA technology and following the IEEE1451.0 Std. foundations, next sections present an implemented

architecture and the RecTool that enables users to reconfigure a weblab infrastructure with embedded I&Ms.

### III. IMPLEMENTED ARCHITECTURE

The implemented architecture enables reconfiguring weblabs without changing the hardware platform where their infrastructure is implemented. Instead of using traditional instrumentation to interact with the EUT, the architecture adopts embedded I&Ms, described through HDL files able to be synthesized into FPGAs, that form the core of the weblab infrastructure through the so-called FPGA-based boards. Besides an FPGA, these boards also integrate other digital and analog I/O peripherals to interface the EUT. Therefore, taking the advantage of the reconfigurable nature of FPGAs and their ability of embedding different I&M that may run in parallel, allows, with the inherent power limitations of the I/O peripherals, reconfiguring weblab infrastructures using the same hardware platform. Moreover, the objective of creating a standard weblab solution able of being accessed using standard APIs, is guaranteed by the IEEE1451.0 Std..

As illustrated in figure 1 and figure 2, the implemented architecture is supported by a Labserver and by a infrastructure comprehending a NCAP and a TIM, in accordance with the IEEE1451.0 Std.. The Labserver was implemented through a common PC acting as a HTTP web server. It integrates several tools for creating a binary code file used for reconfiguring the TIM, which is implemented by an FPGA-based board that interface the EUT. A micro computer, also acting as a small HTTP web server, implements the NCAP, that is able of being accessed through IEEE1451-HTTP commands, and interfaces the TIM using two connection types: i) reconfiguration connection implemented through a JTAG bus, used to reconfigure the FPGA, and ii) control/monitoring connection implemented through a RS-232 bus, used to access each embedded I&M.

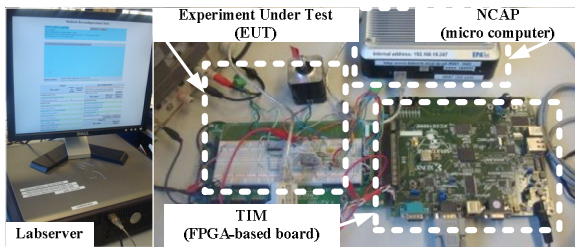


Figure 1. Picture of the weblab architecture.

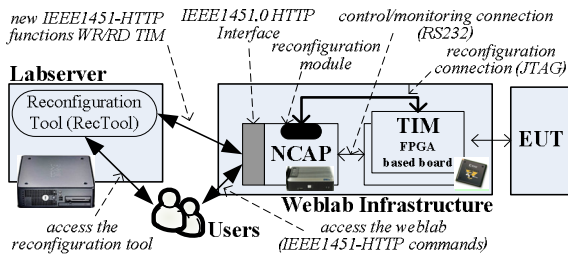


Figure 2. Block diagram of the weblab architecture.

Users directly interact with the Labserver accessing it through a RecTool. This tool enables users to describe a weblab project, synthesizing it to a binary code file used to reconfigure the FPGA-based board, namely by defining and/or selecting files that describe each I&Ms required to run a specific experiment. Those I&Ms are embedded into the FPGA with a predefined IEEE1451-module, described through HDL files [19], according to the IEEE1451.0 Std.. The NCAP implements the interface with the TIM, enabling users' control/monitoring of the EUT using IEEE1451-HTTP commands. The weblab reconfiguration is also possible by the use of new commands named WriteTIM and ReadTIM, which represent an extension to the IEEE1451-HTTP API already proposed in [18]. They are implemented with the remaining IEEE1451-HTTP commands in the NCAP providing, this way, the ability of writing and reading the TIM configuration. The RecTool uses these commands to monitor and change the weblab configuration using a module inside the NCAP to interface the TIM by the reconfiguration connection. The control connection allows users to remotely control/monitoring each embedded I&M using IEEE1451-HTTP commands.

### IV. RECONFIGURATION TOOL

The RecTool is accessed through a Reconfiguration Tool Interface (RecToolInt) using a simple web browser, without any specific plug-ins or software tools installed in the users' device. It enables reconfiguring the weblab infrastructure with different I&Ms, changing it according to the experiments' requirements. The weblab infrastructure is based on a FPGA device and uses an IEEE1451-module already described in HDL files available in the Labserver File System (LFS). For controlling the I&Ms using the IEEE1451-HTTP commands, users should follow a specific methodology for binding those I&Ms into the IEEE1451-module. If this process is successfully made, creating this way a weblab project described through HDL files, users should synthesize it to reconfigure the weblab infrastructure.

#### A. Interface

The weblab reconfiguration is made through the RecToolInt illustrated in figure 3. It is divided in three main sections, enabling users to control all the reconfiguration process:

- i) upload - Allows uploading configuration and project files to create the weblab project, and/or files, already synthesized by this same RecToolInt, to reconfigure the weblab infrastructure;
- ii) information - All feedback actions made by users are printed in this section, which also presents the current Labserver state, namely its current time and if it is busy synthesizing a weblab project;
- iii) panels - This section has three panels: i) Build panel, presents all weblab project files; ii) Reconfiguration panel, has all synthesized files used to reconfigure the TIM; and iii) Reports panel, provides reports generated during users' interaction with the RecToolInt.

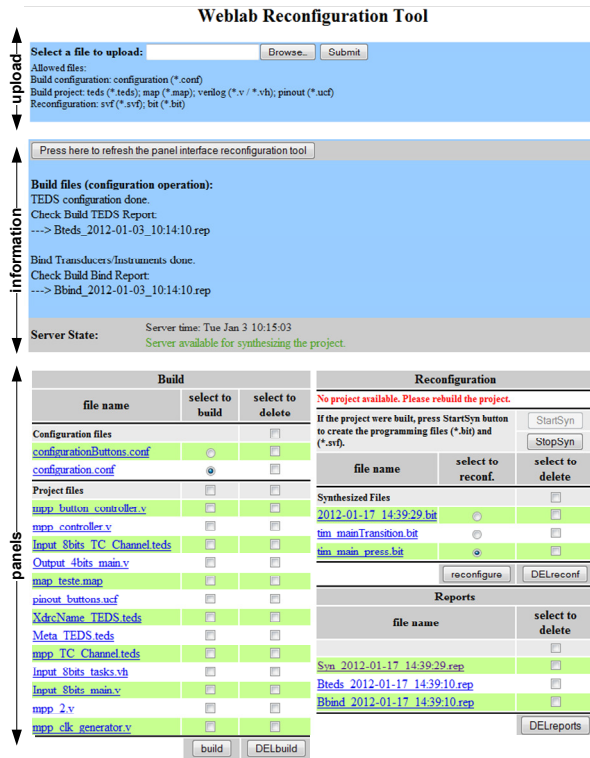


Figure 3. Weblab Reconfiguration Tool Interface (RecToolInt).

The LFS provides a space shared by different users for generic files and for applications; and another space reserved for each user, so they can keep their own files. Those files may be divided into two groups: i) files used for building a project (available in the build panel), and ii) files generated by the RecTool (available in the reconfiguration and reports panels) that have in their names the date of their creation, so users may understand the cause and the time when they were created (e.g.: Bbind\_date.rep).

For building the weblab project, the build panel has two file groups:

- i) configuration files (\*.conf) - Text files containing all rules used for redefining the weblab project, namely for check consistency, generate and interface project files into the IEEE1451-module. It also specifies all configurations required for binding I&Ms to that same infrastructure.
- ii) project files (\*.v/vh/map/teds/ucf) - Comprehend five file types required to build the weblab project: i) HDL files with the design of each I&M (\*.v); ii) HDL files with the interface for binding those I&Ms with the IEEE1451-module (\*.vh); iii) binary files describing the TEDSs used by the IEEE1451-module to access the I&Ms (\*.teds); iv) a binary file to map those same TEDSs into the IEEE1451-module (\*.map); and v) one file to describe the pinout used by the FPGA-based board (\*.ucf).

The Reconfiguration panel provides two file types generated by the Labserver for reconfiguring the TIM. It may have bitstream files (\*.bit), that contains the binary code used to reconfigure the TIM, and/or Simple Vector

Format (SVF) files (\*.svf), that have that same binary code and a set of instructions to control the JTAG interface adopted by the reconfiguration connection.

During users' interaction, the RecTool may generate five report types (\*.rep) that will be available in the Reports panel, namely:

- i) Bbind\_date.rep - Bind of the I&Ms with the IEEE1451-module;
- ii) Bteds\_date.rep - TEDS consistency check, generation of HDL files with TEDS contents and their interface with the IEEE1451-module;
- iii) Syn\_date.rep - Weblab project synthesis;
- iv) Reconf\_date.rep - Reconfiguration process;
- v) Svf\_date.rep - Creation of a \*.svf file.

Despite the organized sections presented by the RecToolInt, users should follow a specific utilization sequence for reconfiguring the weblab infrastructure, as presented in the next subsection.

### B. Utilization sequence

The RecToolInt was developed so users may keep tracking of all operations made during the reconfiguration process. The common sequence involves 3 main operations: i) build the weblab project binding all I&Ms to the IEEE1451-module; ii) synthesize the project to create an \*.svf file; and iii) sending that file to the NCAP to reconfigure the TIM. During this sequence, users may interact with the RecToolInt from different access points, as illustrated in the diagram of figure 4.

The access point depends if users want to create a new weblab project or using an old one already synthesized and available in LFS. Five access points are considered:

- i) Access 1 - After uploading project/configuration files, users may follow the entire sequence or go to access points 4 or 5 to reconfigure the TIM;
- ii) Access 2 - Users have already all project files in their LFS space to create a new weblab project. No uploading is required, but users should follow the remaining sequence;
- iii) Access 3 - The weblab project is already built and available to be synthesized;
- iv) Access 4 - The binary code files (\*.bit) used to reconfigure the FPGA are already available in the user's LFS space. Users should select one \*.bit to be converted to an \*.svf file used to reconfigure the TIM;
- v) Access 5 - An \*.svf file is available to reconfigure the TIM. This access point is the light one because the Labserver only has to send and monitor the reconfiguration process. No file conversion is required, only a report will be generated.

During the upload, only relevant file types are allowed (\*.conf/ucf/vh/v/teds/map). Once selected and uploaded, they will be placed in the user's LFS space appearing in the correspondent RecToolInt panel (e.g. if users select a configuration file, it will be automatically placed in the build configuration panel). Next, users must select the files to build the weblab project. In this operation, the RecTool activates a validation mechanism to guarantee that only allowed files were selected with the correct cardinality (e.g. only one \*.map and \*.ucf files can be



selected). At the end, users should initiate the build operation, that is concluded almost instantaneous because the processing power required to the Labserver machine is small, and therefore, the time consumed is short, which does not happen in the synthesis operation. If there are no reported errors after the build operation, users should start the synthesis. This may be much time consuming, from a few minutes to hours, depending on the complexity of the weblab project, that is related with the embedded I&Ms selected to bind with the IEEE1451-module, the Labserver power processing, and the reported errors, i.e., if during the synthesis operation an early error were detected, it stops the synthesis in a short period of time, otherwise it may takes hours before stopping and retrieving that error. Although users may stop the synthesis by pressing the *StopSyn* button, it was decided to run the synthesis operation in background, allowing users to keep interacting with the RecToolInt, but with some restrictions: i) they cannot start another synthesis since the involved consuming power processing can stuck the Labserver machine or increase much more the time required to finish the operation, and ii) the build operation becomes enactive, because users cannot change the weblab project files while a synthesis is running. However, users can still access the RecToolInt using the access 4 and 5, i.e. they can reconfigure the weblab infrastructure with solutions already available in their LFS space. To alert users that a synthesis operation has finished, the RecTool will automatically send an e-mail suggesting users to consult the RecToolInt, namely the *Syn* report to evaluate if the synthesis was succeeded. If

there are no reported errors, an \*.bit becomes available so users may select it to start the reconfiguration operation.

The reconfiguration operation is not so time consuming as in the previous synthesis operation. It involves sending an \*.svf file to the weblab infrastructure through the NCAP. Users may start the reconfiguration by selecting an \*.bit or \*.svf file using accesses 4 or 5. In access 4, the selection of a \*.bit file requires the use of the RecTool to create the \*.svf file that will be automatically transferred to the NCAP, becoming available in the reconfiguration panel for future use. In access 5, the \*.svf file is already available to be transferred, and no file conversion is required. Both accesses require some processing in the RecTool but, since they are concluded in a few seconds, they don't run in background. The generated reports give possible errors occurred during the reconfiguration (*Reconf\_date.rep*) and during the creation of the \*.svf file (*svf\_date.rep*). When errors occur, information will be printed in the information panel and in the generated reports. In situations of an unsuccessful reconfiguration not detailed in reports, users may considerate that the NCAP is offline or the network, that interfaces the NCAP with the Labserver, is down. If no errors are reported, that means the TIM was correctly reconfigured, and users should evaluate it using IEEE1451-HTTP commands.

Next section details all RecTool implementation issues, namely the involved tools, files of the LFS and all the management made after every operation.

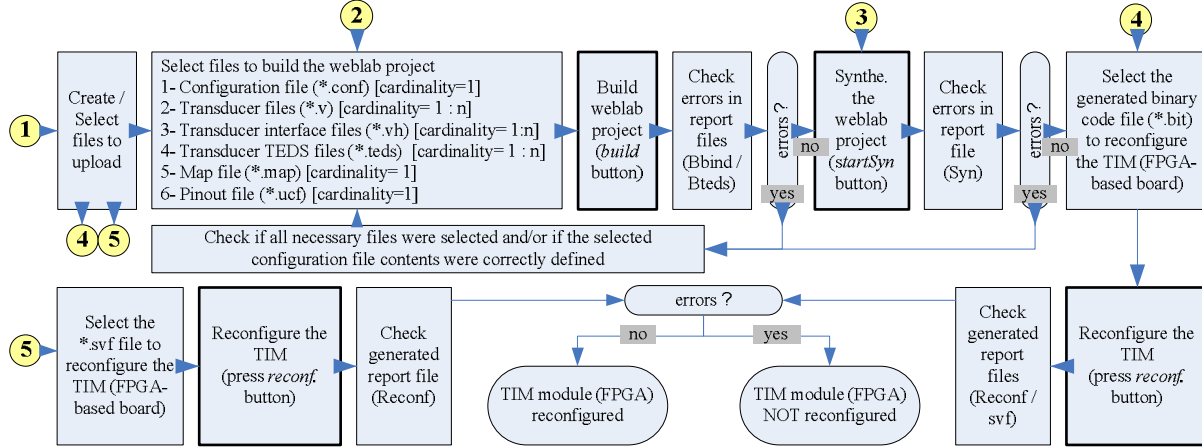


Figure 4. Utilization sequence of the RecToolInt.

### C. Implementation issues

The RecTool uses several software applications to reconfigure the TIM. All actions made by the Labserver are supported by the LFS that gathers files and applications required to manage and reconfigure the TIM using a specific software application in the NCAP to send the binary code, available in a \*.svf file, to the FPGA.

The Labserver was implemented in a computer with an Ubuntu Linux distribution [20] running the Apache HTTP server [21] and the PHP Hypertext Preprocessor [22] to provide remote access to the RecToolInt. It uses several software applications controlled and monitored by a Labserver Controller (LC) developed using the PHP

server-side scripting language. As illustrated in figure 5, all actions made in the RecToolInt are managed by the LC that, supported by the LFS, uses a set of software applications to build and synthesize the weblab project used to reconfigure the TIM.

Despite the current version of the RecToolInt does not manage users' accesses, the LFS organization guarantees that future developments may easily handle this issue. For this purpose, besides the installation of several proprietary software applications in the Labserver, the LFS was divided in two folder groups: i) the TIM folder, that provides all weblab project HDL files and configuration programs to redefine the project according to the selected I&M; and ii) the users' folder, that contains all files that

belong to a specific user, namely the build, configuration and report files, and project files created in the *ise\_project* folder during a synthesis operation.

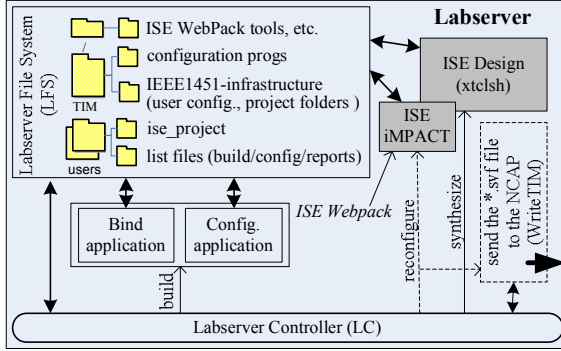


Figure 5. Labserver internal modules.

The I&Ms files selected by each user in the build panel, and the HDL files describing the IEEE1451-module, that are already available within project folders inside the *TIM/IEEE1451-infrastructure* folder, are automatically redefined by two software applications available in the TIM folder, named *Bind* and *Config*. These applications, developed specifically for the RecTool, change some of the IEEE1451-module HDL files based on the configuration and project files selected in the build panel of the RecToolInt. This way, it is possible to bind different I&M to the same IEEE1451-module, since all changes are automatic and transparent to the user.

Internally, after pressing the *build* button, the Labserver manages the LFS files so the *Bind* and *Config* applications may reformulate the weblab project according to the rules defined in the selected configuration file. Those rules, exemplified in figure 6a), are defined through a set of predefined tags that should be in accordance with the files selected in the project panel, otherwise the *Bind* and *Config* applications retrieve errors. Besides specifying the number of adopted TCs, the configuration file defines: i) HDL files to bind the I&Ms to the IEEE1451-module, namely the \*.v, describing the I&Ms, and \*.vh files, describing internal HDL tasks for interfacing the IEEE1451-module to those I&Ms; ii) signals to specify the bus connections of that interface; iii) TEDSs and their associations to a set of memories, which are automatically created and integrated into the weblab project; and iv) a map table to associate each TIM/TC to one or more TEDS previously generated in the build operation. Not specified by the IEEE1451.0 Std., this map table is specially defined for the implemented IEEE1451-module. As illustrated in figure 6b), it comprehends a structure similar to a TEDS, whose contents map each TIM/TC to one or more TEDSs. When the build operation finishes, the Bbind and Bteds reports are created with all information generated by the *Bind* and *Config* applications.

While the build operation uses standard HDLs, which makes it independent of the adopted technology for the TIM, the synthesis operation, initiated after building the weblab project, requires the use of technological-dependent applications. Therefore, since the TIM was implemented in a FPGA-based board with a Xilinx FPGA Spartan3E-1600 device, it was selected the ISE Webpack design software from Xilinx [23].

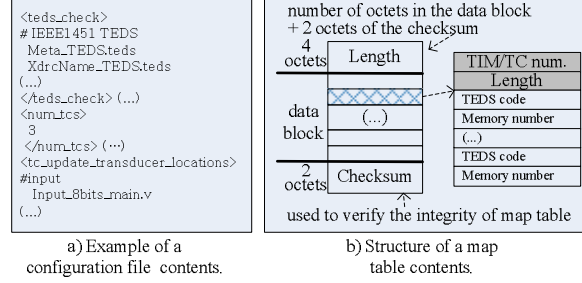


Figure 6. Configuration file example and structure of a map table.

Based on the weblab project created during the build operation, in the synthesis operation the LC creates an ISE project inside the *user/ise\_project* folder. Since the synthesis is usually much time and computational resource consuming, the RecTool only runs a single synthesis operation. In this operation, the LC starts evaluating the Labserver availability checking if a synthesis is already running. If no synthesis is running, the LC starts the synthesis operation by setting an internal variable (shared by all users) indicating the Labserver became busy, and creating a Tool Command Language (TCL) script file [24]. This script contains all the instructions to control the execution of the ISE tool during the synthesis operation namely, the name of the project that will be created in the *user/ise\_project* folder, the adopted FPGA device, synthesis directives, indication of all files used in the project and, writes specific instructions to send an automatic email to the user when the synthesis operation finishes. Due to the large time required to finish the synthesis, the TCL script is executed in background using the *xtclsh* tool that belongs to the ISE Webpack.

The last operation is the reconfiguration, which involves sending an \*.svf file to the weblab infrastructure using the HTTP WriteTIM command. During this operation, the management made in the Labserver depends on the selected file made in the reconfiguration panel. Since the NCAP can only handle \*.svf files to reconfigure the TIM, if users select an \*.bit file it will be converted to an \*.svf file. Internally, the LC starts evaluating the selected file type, and if it is a \*.bit file it starts running a tool named *iMPACT* to transform the \*.bit into an \*.svf file. As with the *xtclsh* tool used for the synthesis operation, the *iMPACT* also belongs to the ISE Webpack, since the target is an FPGA from Xilinx. Hence, if other FPGA from a different manufacturer was selected, other applications can be adapted to the RecTool, since both the *xtclsh* and the *iMPACT* are accessed using Linux commands defined in bash files.

To upload the \*.svf file from the Labserver to the NCAP, the LC uses the *libcurl* API that is a multiprotocol file transfer library [25] that enabled the use of the HTTP WriteTIM command. Once uploaded to the NCAP, a tool named UrJTAG [26] reads the \*.svf file and, using the JTAG connection to the FPGA-based board, reconfigures the FPGA. The success of this operation can be monitored using the information panel and the output of the UrJTAG software, which is printed in the *Conf\_date.rep* report file.

## V. DEMO EXAMPLE

To validate the RecTool, the weblab infrastructure was reconfigured with 3 modules: two digital I/O modules and one module to control a bipolar step-motor. The digital I/O modules group the external connections in 8 inputs and 4 outputs, respectively, controlled using two independent TCs. The step-motor controller uses a single TC to control a set of 6 digital output lines that manage the velocity, direction, number and step sequences of the motor. Despite this paper is not focused on detailing the way these modules were constructed, each of them use a set of project files, namely TEDSs, HDL files to describe the modules and to interface them with the IEEE1451-module, a \*.map file to associate TEDSs with TCs and a \*.ucf file to describe the external connections of the FPGA-based board. Using the RecToolInt all these files and one configuration file were uploaded to the LFS and selected before pressing the *build* button. Once pressed, the build operation generated two reports (Bteds and Bbind) without errors, as exemplified in figure 7.

```
> TEDS files contents are correctly defined.
> Memory Verilog TEDS files correctly defined.
> MAP_Table verilog file correctly defined.
(...)
> Status memory correctly created.
> State memory correctly created.
-----
a) Output extracted from the report file Bteds_date.rep
**General interface files:
> <declarations> correctly created.
> <directions> correctly created.
> <initial> correctly created.
(...)
> <bps_uart> correctly created.
-----
b) Output extracted from the report file Bbind_date.rep
```

Figure 7. Output extracted from the generated reports Bteds and Bbind after a build operation.

Since no errors were generated, the synthesis operation started after pressing the *startSyn* button. During synthesis, we have faced some problems since the Labserver machine entered in a deadlock state, which didn't happen when we synthesized the same project (with the same attached I&Ms) in a personal machine, despite it took about 4 hours to finish. Considering the characteristics of both machines, illustrated in table II, which shows that the personal machine has higher memory and power processing than the Labserver machine, we concluded the problem is caused by: i) the associated complexity of the implemented IEEE1451-module, namely with some implemented low-level commands that uses large FPGA resources and, ii) the limited memory and power processing of the adopted Labserver machine.

TABLE II. MACHINES USED FOR SYNTHESIS.

<b>Personal machine: Intel Core 2 D CPU P84002.26GHz (2 GB RAM)</b> Synthesis concluded in about 4 hours with the complete IEEE1451-module.
<b>Labserver machine: Intel Pentium D CPU 3.40GHz (1 GB RAM)</b> Synthesis entered in a deadlock state when using the complete IEEE1451-module. Only after removing most of the low-level commands from the IEEE1451-module the synthesis was concluded in about 8 minutes.

Therefore, to validate the current RecTool, we have decided to exclude most of the low-level commands implemented by the IEEE1451-module, just using the required to start and stop the step-motor. As expected, the synthesis operation was correctly concluded in a few minutes.

The synthesis report, that includes all information generated by the *xtclsh* tool, did not indicate any error (only a few warnings), meaning that the \*.bit file was successfully generated, as indicated in figure 8a). After selecting this file and pressing the *config* button, it was converted to an \*.svf file, as illustrated by the extracted output lines of the svf\_date.rep presented in figure 8b). During this same operation, the \*.svf file was sent to the NCAP for reconfiguring the TIM, i.e. the FPGA-based board. As in previous operations, its success was evaluated by the generated report, exemplified in figure 8c), which corresponds to the output generated by the UrJTAG installed in the NCAP. At this stage the weblab infrastructure was reconfigured, however, to validate it, the IEEE1451-HTTP Trigger command was applied to the TC number 3 (that corresponds to the step-motor controller), correctly starting the step-motor as proved by the reply presented in figure 8d).

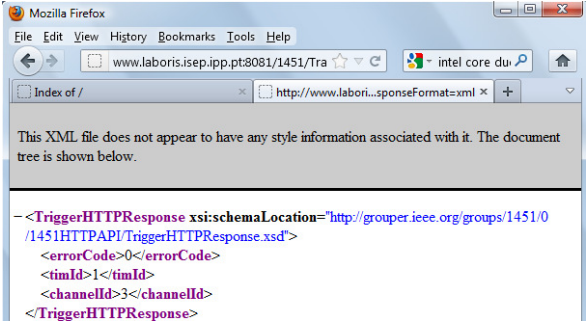
```
(...) Started : "Synthesize - XST".
Running xst... (...)
Process "Generate Post-Place & Route Static Timing" completed successfully
(...) Process "Generate Programming File" completed successfully
-----
a) Extract of the report file Syn_date.html
-----
Release 13.3 - iMPACT O.76xd (lin)
Copyright (c) 1995-2011 Xilinx, Inc. All rights reserved. (...)
'1': Programmed successfully.
Elapsed time = 1 sec.
-----
b) Extract of the report file svf_date.rep
-----
Connected to libftdi driver.
IR length: 30 Chain length: 4 (...)
Warning: USB-Blaster frequency is fixed to 12000000 Hz
-----
c) Extract of the report file Reconf_date.rep
-----
Command and response in XML format (in accordance with the
IEEE1451-HTTP API):
http://www.laboris.isep.ipp.pt:8081/1451/TransducerManager/Trigger?
timId=1&channelId=3&timeout_secs=123&timeout_nsecs=123&
triggertime_secs=123&triggertime_nsecs=123&SamplingMode=4
&responseFormat=xml
-----

-----
d) Example of using a IEEE1451-HTTP Trigger command.
```

Figure 8. Outputs extracted from report files (syn, svf and reconf) and an example of using the Trigger IEEE1451-HTTP command.

## VI. CONCLUSIONS

Although some weblabs already provide a reconfiguration capability, they traditionally follow non-standard solutions and restrict the reconfiguration to a preselected number of I&Ms able to connect into different locations of an EUT. Additionally, those I&Ms are supported by independent hardware platforms, eventually expensive, becoming improper to be shared by different weblab infrastructures.

This paper presented a solution that may handle these limitations by the adoption of embedded I&Ms able to be reconfigured into FPGA-based boards that belong to the weblab infrastructure. For an institutional collaboration during the development and integration of those I&Ms into weblab infrastructures, we adopted the IEEE1451.0 Std. whose foundations are general enough to integrate different technologies but, at the same time, to specify an architecture that enables to develop, network-interface and access each I&M using a set of standard HTTP commands. To reconfigure the infrastructure we developed a RecTool, describing its interface, utilization sequence and some implementation issues that can be useful for future developments, namely for solutions based on different FPGAs' manufacturers. The RecTool was evaluated by reconfiguring the weblab infrastructure with 3 modules (two digital I/Os and one step-motor controller). The results of this evaluation showed that FPGA technology, especially the tools required for synthesizing the implemented weblab project, requires high-power processing of the Labserver machine, and may be much time consuming, which can be a restriction in situations requiring fast weblab reconfigurations. As indicated in the results obtained from the demo example, a solution is to adopt a powerful machine for the Labserver, or simplify the IEEE1451-module redefining it by selecting a limited number of low-level commands, only the required for each specific weblab infrastructure. Additionally, in spite of our solution suggests the use of a unique hardware platform based on FPGA-based boards, they currently have a restricted number of I/O interfaces with limited power, which can become inappropriate for some experiments.

Nevertheless all these technical problems, current trends show that technology is in a permanent evolution, leading us to believe they will be solved in the near future namely with: i) powerful computer machines, for running faster and more efficiently the synthesis operation (eventually with updated tools) and, ii) a large offer of FPGA-based boards integrating more and powerful I/O interfaces, at lower prices. Consequently, the presented solution can be important for a wider adoption of weblabs in education, because it provides a way for reconfiguring the infrastructures using shareable I&Ms, which may lead to significant cost reductions and improvements in the students' design skills. Moreover, since it follows a standard approach (uses the IEEE1451.0 Std. and I&Ms described through standard HDLs) it can be a stimulus for involving all educational actors in the creation and maintenance of weblab infrastructures. Supported by these perspectives, in the future we intend to run a deep evaluation of this architecture to understand its relevance in a real educational scenario.

## REFERENCES

- [1] Norrie S. Edward, 'The role of laboratory work in engineering education: student and staff perceptions', *International Journal of Electrical Engineering Education*, vol. 39, no. 1, pp. 11–19, Jan. 2002.
- [2] L. D. Feisel and A. J. Rosa, 'The Role of the Laboratory in Undergraduate Engineering Education', *Journal of Engineering Education*, vol. 94, no. 1, pp. 121–130, Jan. 2005.
- [3] J. Ma and J.V. Nickerson, 'Hands-on, simulated, and remote laboratories: A comparative literature review', *ACM Computing Surveys*, vol. 38, no. 3, p. 7, 2006.
- [4] O. A Soysal, 'Computer Integrated Experimentation in Electrical Engineering Education Over Distance', *ASEE 2000 Annual Conference, Saint Louis, MO*, Jun. 2000.
- [5] Javier García Zubía and Gustavo R. Alves, *Using Remote Labs in Education - Two Little Ducks in Remote Experimentation*, 1 vols. University of Deusto Publications, 2011.
- [6] Gomes, L. and Bogosyan, S., 'Current Trends in Remote Laboratories', *IEEE Transactions on Industrial Electronics*, vol. 56, no. 12, pp. 4744–4756, Dec. 2009.
- [7] N. Sousa et al., 'An Integrated Reusable Remote Laboratory to Complement Electronics Teaching', *IEEE Transactions on Learning Technologies, Special Issue on Remote Labs*, 2009.
- [8] ABET - Accreditation Board for Engineering and Technology, [Online]. Available: <http://www.abet.org/>. [Accessed: 24-Jan-2012].
- [9] Lyle D. Feisel and Albert J. Rosa, 'A Colloquy on Learning Objectives For Engineering Education Laboratories', *American Society for Engineering Education Annual Conference*, 2002.
- [10] Lyle D. Feisel and George D. Peterson, 'Learning objectives for engineering education laboratories', *32nd Annual Frontiers in Education (FIE'02) in Boston, MA*, Nov. 2002.
- [11] I. Gustavsson, 'A Flexible Electronics Laboratory with Local and Remote Workbenches in a Grid', *International Journal of Online Engineering (iJOE)*, vol. 2, no. 3, 2008.
- [12] IEEE 1451.0 Std., 'Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats', *The Institute of Electrical and Electronics Engineers, Inc.*, p. 335, 2007.
- [13] IVI Foundation - Interchangeable Virtual Instrument Foundation, [Online]. Available: <http://www.ivifoundation.org>. [Accessed: 26-Jan-2012].
- [14] GOLC - Global Online Laboratory Consortium, [Online]. Available: <http://online-lab.org/>. [Accessed: 26-Jan-2012].



- [15] C. Q. Grana and E. M. Perez, 'Reconfigurable Platform to implement Electronic Instrumentation', in *IEEE Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007*, 2007, pp. 1–6.
- [16] M. J. Moure, M. D. Valdes, E. Mandado, and A. Salverria, 'Educational application of virtual instruments based on reconfigurable logic', in *IEEE International Conference on Microelectronic Systems Education, 1999. MSE '99*, 1999, pp. 24–25.
- [17] C. Quintans, M. J. Moure, M. D. . Pena, and E. Mandado, 'A virtual instrumentation laboratory based on a reconfigurable coprocessor', *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 2, pp. 635–645, Apr. 2006.
- [18] Ricardo J. Costa, Gustavo R. Alves and Mário Zenha-Rela, 'Extending the IEEE1451.0 Std. to serve distributed weblab architectures', in *1st Experiment@ International Conference (exp.at'11)*, Calouste Gulbenkian Foundation, Lisboa-Portugal, <http://ave.dee.isep.ipp.pt/~rjc/>, 2011.
- [19] Ricardo Costa, Gustavo Alves and Mário Zenha-Rela, 'Work-in-progress on a thin IEEE1451.0-architecture to implement reconfigurable weblab infrastructures', *International Journal of Online Engineering (iJOE)*, vol. 7, no. 3, 2011.
- [20] Ubuntu operating system, [Online]. Available: <http://www.ubuntu.com/>. [Accessed: 24-Jan-2012].
- [21] Apache HTTP Server Project, [Online]. Available: <http://httpd.apache.org/>. [Accessed: 24-Jan-2012].
- [22] PHP: Hypertext Preprocessor, [Online]. Available: <http://www.php.net/>. [Accessed: 24-Jan-2012].
- [23] ISE WebPACK Design Software, [Online]. Available: <http://www.xilinx.com/tools/webpack.htm>. [Accessed: 24-Jan-2012].
- [24] Tcl Developer site, [Online]. Available: <http://www.tcl.tk/>. [Accessed: 24-Jan-2012].
- [25] cURL and libcurl, [Online]. Available: <http://curl.haxx.se/>. [Accessed: 24-Jan-2012].
- [26] UrJTAG - Universal JTAG library, server and tools, [Online]. Available: <http://urjtag.org/>. [Accessed: 24-Jan-2012].

#### AUTHORS

**Ricardo Costa** is an Assistant Professor at the Polytechnic of Porto, School of Engineering (IPP/ISEP), and he is a PhD. Student at the Faculty of Sciences and Technology of the University of Coimbra (FCTUC) (e-mail: [rjc@isep.ipp.pt](mailto:rjc@isep.ipp.pt) / [rjc@dei.uc.pt](mailto:rjc@dei.uc.pt) ).

**Gustavo Alves** is a full Professor at the Polytechnic Institute of Porto, School of Engineering (IPP/ISEP), (e-mail: [gca@isep.ipp.pt](mailto:gca@isep.ipp.pt) ).

**Mário-Zenha Relá** is an Auxiliary Professor at the Faculty of Sciences and Technology of the University of Coimbra (FCTUC) (e-mail: [mzrela@dei.uc.pt](mailto:mzrela@dei.uc.pt) ).