Extending the IEEE1451.0 Std. to serve distributed weblab architectures

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

***Abstract* - The appliance of the IEEE1451.0 Std. into the remote experimentation domain may be an interesting solution not only to develop reconfigurable weblab infrastructures, but also to improve the way infrastructures, and their experiments, may be shared. Therefore, this paper proposes a distributed weblab architecture supported on a IEEE1451 concept named Transducer Electronic Data Sheet (TEDS). It is suggested the use of a new TEDS, named LabTEDS, to provide information about weblab infrastructures namely, web location, technical resources and type of experiments described according a metadata model specification defined by the Lab2go project. The access to the architecture is made through the IEEE1451.0 HTTP API extended with new functions. At the end of the paper a thin implementation of the architecture is presented, supported on a cross-mapping established between the HTTP functions and the low-level commands, which are used to control the weblabs.**

Keywords - Weblabs, Weblab architectures, IEEE1451.0 Std., Remote Experimentation.

I. INTRODUCTION

Changes in education were promoted by the appearance of new technological resources. One of those resources is the Internet and associated services that allow remotely accessing educational contents like real experiments, which are fundamental for fulfilling the learning objectives in engineering education [1][2]. In the last 2 decades, several institutions have being creating weblab infrastructures providing remote access to experiments that were previously only accessible using traditional laboratories, e.g. [3]. However, those infrastructures are generally developed following different architectures, which difficult their development and sharing because they use different Application Programming Interfaces (APIs) to access their features and, in many situations, they are not able to be easily found in the World Wide Web. Many experts are aware of these limitations, which led to the creation of the Global Online Laboratory Consortium (GOLC) [4] that, among other objectives, focus on researching ways to improve collaboration among institutions, and to facilitate users' (students & teachers) access to every weblab by using standards. Therefore, this paper suggests the use of the IEEE1451.0 Std. to serve distributed weblab architectures. Supported on its characteristics, that focus on implementing and network-interfacing transducers (sensors and actuators), it is proposed some extensions to facilitate the dissemination and the access to weblab infrastructures and experiments they may handle. So readers may understand the proposed IEEE1451.0 Std. extensions, section II provides a brief overview of the standard, focusing on Transducer Electronic Data Sheets (TEDSs) that specifie technical characteristics of transducers, and on functions provided by the IEEE1451.0 HTTP API, that allow remotely accessing those same transducers. Section III describes the proposed architecture, and section IV presents a new TEDS, named LabTEDS, to be used for describing weblab infrastructures. The operational sequence for accessing weblab infrastructures is specified in section V, with a detailed description of the registering, discovering and accessing processes. Before concluding, section VI presents a thin implementation of the IEEE1451.0-architecture, that focus on interfacing the IEEE1451.0 HTTP functions to a set of low-level commands for simplifying developments and accesses to weblab infrastructures.

II. IEEE1451.0 STD. OVERVIEW

Defined in 2007, the IEEE1451.0 Std. [5] aims to network-interface transducers and defines a set of operating modes, based on specifications provided by TEDSs. It is the basis for forthcoming and previous members of the IEEE1451.x family, so they can operate together according to operating modes controlled through low-level commands that can be applied using a set of APIs. The standard defines an architecture based on two modules: the Transducer Interface Module (TIM) that controls Transducer Channels (TCs), and the Network Capable Application Processor (NCAP) that provides network access to those TCs. Each module is connected through an interface defined by another standard of the IEEE1451.x family, some already specified according to the IEEE1451.0 Std. (e.g. the IEEEp1451.6 Std. for the CANopen interface) and others intended to be modified in the future (e.g. IEEE1451.2 Std. which defines point-to-point interface).

To control the behaviour of a transducer, the IEEE1451.0 Std. provides low-level commands to read/write TEDSs, control transducers, etc., and allows manufacturers to define their own commands. To monitor the operations of a transducer, the IEEE1451.0 Std. implements a 32 bit status register for each TC and another for the TIM. To access (remotely or not) a specific transducer 3 APIs are specified:

- Module Application API: implemented both in the TIM and in the NCAP, provides functions to transfer data and issue commands between these two modules;
- Transducer Services API: implemented in the NCAP, provides the interface between applications running in the NCAP and low-level commands defined in the TIM;
- HTTP API: implemented in the NCAP, uses the HTTP protocol and defines a communication message format for accessing the Transducer Services API.