Intelligent Measureserver for Controlling Remote Real Experiments with Embedded Simulations and Advanced Diagnostics

Ing. Michal Gerža, Ph.D.

Doctoral Thesis Summary

Tomas Bata University in Zlín

Tomas Bata University in Zlín
Faculty of Applied Informatics

Doctoral Thesis Summary

Inteligentní Measureserver pro řízení vzdálených experimentů s vestavěnou simulací a pokročilou diagnostikou

Intelligent Measureserver for Controlling Remote Real Experiments with Embedded Simulations and Advanced Diagnostics

Author:	Michal Gerža, MSc.
Degree programme:	Engineering Informatics P3902
Degree course:	Engineering Informatics 3902V023
Supervisor:	Prof. Dr. František Schauer, DSc.
External examiners:	Prof. Dr. Javier Garcia-Zubia Prof. Dr. Mikuláš Huba Assoc. Prof. Dr. Peter Frankovský

Zlín, November 2017

Published by **Tomas Bata University in Zlín** in the Edition **Doctoral Thesis Summary**.
The publication was issued in the year 2017.

Key words in Czech: *REMLABNET, RLMS, ISES, Measureserver, vzdálená laboratoř, zálohování dat, pokročilá diagnostika, vestavěná* simulace.

Key words: *REMLABNET, RLMS, ISES, Measureserver, remote laboratory, data archiving, advanced diagnostics, embedded simulation.*

Full text of the doctoral thesis is available in the Library of TBU in Zlín.

ISBN 978-80-7454-707-2

ABSTRAKT

Lidé vyvíjejí různé technologie využívající fyzikálních zákonů, které popisují přírodní jevy. Tento vývoj je kontinuální a je v současné době rychlejší a intenzivnější. Dosažený technologický pokrok ovlivňuje všechny oblasti každodenního života zahrnující i oblast vzdělávání. Současná společnost požaduje složitější nástroje poskytující dostupnější a efektivnější vzdělávací metody. Informační a komunikační technologie nabízejí mnoho nástrojů vyhovující požadavkům pro podporu výuky a studia, obzvláště v oblasti přírodních věd a techniky. Studium těchto oborů je založeno na reálné experimentální práci v laboratořích se specifickým vybavením a zařízeními pro lepší pochopení daných jevů. Tato forma studia, kdy uživatel musí pravidelně navštěvovat laboratoře, aby vykonal nezbytná měření, již není akceptována z mnoha důvodů. Většina laboratoří nemůže poskytnout široký rozsah reálných experimentů, kde jsou k dispozici drahá zařízení pro objasnění a analýzu požadovaných jevů. Vývoj směrem k těmto požadavkům vedl k návrhu a realizaci konceptu vzdálených laboratoří s cílem poskytnout fyzikální experimenty přes Internet. Připojeným uživatelům jsou poskytnuty rozdílné nástroje, jako je například odpovídající teorie ke studovanému jevu. V našem případě je koncept nazván Inteligentní školní experimentální systém (ISES) běžící jako otevřená systémová platforma.

Tato doktorská práce se zaměřuje na návrh a implementaci softwarových komponent související s jednotkou Measureserver ISES, v podstatě s konečným stavovým automatem, jenž je hlavní jednotkou této platformy. Aktivity v této práci jsou zaměřeny na pět úkolů, které jsou definovány jako cíle této disertace. Tyto cíle byly řešeny jako nezávislé projektové práce, integrující progresivní koncepty, přístupy a technologie, které přinášejí nové funkce k dosažení lepších výsledků ve výuce, spolehlivosti a údržbě vzdálených laboratoří ISES.

ABSTRACT

People are developing various technologies, using laws of physics behind real-world phenomena. This process is continuous and is presently getting faster and more intensive. Achieved technological advances affect all areas of every day’s life, including the field of education. Contemporary society requires more complex tools providing more accessible and effective educational methods. Information and communication technologies offer many tools to meet the requirements on which rest teaching and learning, especially in natural sciences and engineering fields. Studying these fields is based on real experimental work in laboratories with specific equipment and devices for a better understanding the given phenomenon. This form of study, when the user must regularly attend laboratories to perform necessary measurements, is not always acceptable for many reasons. Most of laboratories cannot provide a wide range of the real experiments, where expensive devices, needed for the elucidating and analysis of desired phenomena, are involved. Development in this direction of these requirements led to the design and implementation of the concept of the remote laboratories to deliver physical experiments via the Internet. The connected users are provided with the different tools as, for example, corresponding theory of the studied phenomenon. In our case, the concept is called Intelligent School Experimental System (ISES) operated as an open system platform.

The doctoral thesis focuses on the design and implementation of software components related to the ISES Measureserver, finite-state machine in principle that is a core unit of this platform. The activities within the thesis are aimed at five tasks, defined as the goals of the thesis. These goals were solved as the independent project works, integrating progressive concepts, approaches and technologies, which bring new features contributing to better teaching outcomes, reliability and maintenance of the ISES remote laboratories.

CONTENTS OF THE THESIS

1. INTRODUCTION 7

1.1 Remote laboratories 7

1.2 Remote laboratories globally 7

2. ISES HANDS-ON LABORATORIES, STATE OF THE ART 9

3. ISES REMOTE LABORATORIES, STATE OF THE ART 10

3.1 ISES remote laboratory history and basic working scheme 10

3.1.1 ISES remote laboratory units 11

3.2 ISES Measureserver unit and its communication 13

3.3 Working scheme of ISES Measureserver unit 13

3.4 Easy Remote Laboratory expert system tool 14

3.5 RLMS REMLABNET 14

4. OBJECTIVES OF THE THESIS 15

5. SELECTED PROCESSING METHODS 16

5.1 Finite-state machine classification and concept 16

5.2 Language parsers 16

5.3 Context-free grammars 17

5.4 ISES Measureserver unit and its psc control script 17

5.5 Grammar syntax of psc script 17

5.6 Libraries and functionalities of psc script 18

5.7 Analysis of Measureserver functioning and data traffic 18

6. RESULTS 19

6.1 Data archiving and logging module 19

6.2 First level diagnostics module 24

6.2.1 Notifying thread component......24

6.2.2 Operating thread component......25

6.3 Second level diagnostics module......28

6.3.1 Detecting thread component......29

6.4 Embedded real-world phenomena simulation module......32

6.4.1 Modified Euler integrator......33

6.4.2 Runge-Kutta integrator......33

6.4.3 Integrator parameters component......33

6.4.4 Data results visualization......33

7. CONTRIBUTION TO SCIENCE AND PRACTICE......36

CONCLUSION......37

BIBLIOGRAPHY......38

LIST OF FIGURES......45

LIST OF SYMBOLS, ACRONYMS AND ABBREVIATIONS USED......47

LIST OF PUBLICATIONS BY THE AUTHOR......48

AUTHOR’S PROFESSIONAL CURRICULUM VITAE......50

1. INTRODUCTION

The Information and Communication Technology today has enabled fast and rich ways of exchanging information among people from different domains with a variety of applications. One such application is the remote laboratory (RL), which are web-based interactive systems giving clients the ability to control and observe the specific characteristics and real processes of remote equipment through the Internet anytime and anywhere.

1.1 Remote laboratories

The RLs can provide access to resources, which are otherwise inaccessible to users, denoted as clients. Typically, all that is required is a web browser and an Internet connection to enable rich educational possibilities. A number of collaborative projects have integrated physical experiments under a common infrastructure. Aims of these efforts typically include simple access by clients and the support of collaboration among connected clients. Research has mainly focused on the integration of the RLs into higher education, predominantly undergraduate courses, and predefined sets of common experiments.

The RLs traditionally consist of two parts, the server and the client. The client's side is used by a client engaging the RL and learning from the use of it. The server side provides the experiment rig (physical hardware), as well as the environment responsible for designing, constructing and maintaining the RL that also provides accompanying teaching and materials. Remote Laboratory Management System (RLMS) serves for the function of components and interfaces of the whole system forming the RL [1].

1.2 Remote laboratories globally

Plenty of RLMS systems covering a broad range of the RLs have been designed and deployed worldwide. Historically, the first two RLs have occurred in Slovakia, nearly simultaneously:

1. In the field of automatic control in the Department of Automatic Control, Faculty of Electrical Engineering of the Slovak University of Technology in Bratislava, that has been developed by the research team of Prof. Mikuláš Huba in 2007 [2][3],
2. In the field of "Electrochemical cell" in 2008, in the workplace of the Department of Physics, Faculty of Education in the University of Trnava

in Trnava, has been developed by team of Prof. František Schauer and Assoc. Prof. Miroslava Ožvoldová [4][5].

Some of the largest and most widely used RLMSs are platforms constituted as the iLab [6][7], SAHARA [8], VISIR [9] and WebLab-Deusto [10].

The iLab has disposable three-layer architecture called the iLab Shared Architecture. Users connect with a service broker server, which in turn makes a connection with the actual laboratory server. The system architecture heavily rests on web services [11]. The iLab has also been used to implement extensions such as iLab-MIT-Africa [12] in African countries and some universities in Australia. Experiments in the iLab have been categorized into three different delivery methods: batched, interactive and sensor.

The SAHARA originally followed a client-server architecture, where all experiments were hosted at the UTS laboratories, and accessed upon request by remote users. In this design, the lists of experiments are stored by the central server, which is also responsible for other operational aspects, which include running the RL, scheduling, and operating the rig.

The VISIR implementations also follow client server architecture, where the complete experiment lists are stored in centralized databases along with user details, and connection to the same server is used for booking and operating the experiment. Both iLab and VISIR use the LabVIEW as the main platform and language to compile controlling programs to run RLs.

The next RL is WebLab-Deusto that has been developed at University of Duesto, Bilbao in Spain [13] and [14]. This system platform uses the client-server mechanism utilizing mostly time reservation with priority queuing based scheduling, although the nature of scheduling may change if connecting to other systems. Within this system, there is a wide variety of experiments ranging from basics of physics to Field Programmable Gate Array however the main focus is on electronics and electrical experiments.

Finally listed RLs in brief, the RemoteElectlab [15] has been developed at University of Porto, Portugal targeting mechanical, physics electronics and meteorological experiments. The eMersion and SMARTLAB have been used at Technical University in Lausanne, Switzerland that uses the Graasp social media platform [16]. This system is also intended for higher education users in the fields of the control theory, physics and others.

2. ISES HANDS-ON LABORATORIES, STATE OF THE ART

The Intelligent School Experimental System (ISES) is established as the advanced experimental tool for school laboratories and building of hands-on laboratory with the real-time operation, data acquisition, data processing, experimental hardware controlling, visualizing and analyzing. This system has been designed and developed by Assoc. Prof. Dr. František Lustig in 1992, and its development still continues [17]. There are registered 500 installations of this experimental tool at Czech and Slovak schools.

The ISES hands-on laboratory is an open system platform consisting of the ISES physical hardware (HW) and ISES WIN software components. It provides the processing of measured data, for example the integration, differentiation, approximation and fitting. The data transfer to plotting environments (preset various charts and data tables) is straightforward. The complete set of the physical modules like meters, probes, sensors, devices and converter, including the ISES WIN software displayed on the screen, is illustrated in Figure 1.

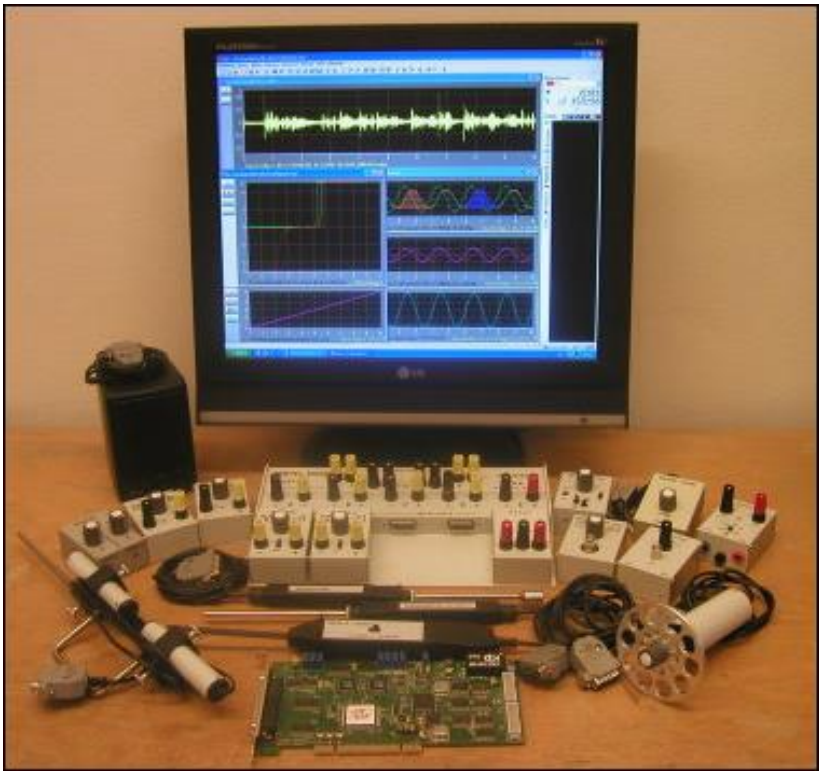


Figure 1 ISES physical hardware and software components; the PCI ADDA converter, the set of meters, sensors and device

3. ISES REMOTE LABORATORIES, STATE OF THE ART

The ISES remote laboratories have come into existence as the next step of the ISES hands-on laboratories. The ISES physical hardware is unique as it possesses, as the only school experimental system in world, both inputs and outputs, which is a prerequisite for building RLs controlled by clients. Each RL offers a software solution for controlling the physical HW, and the client does not need any special program to be installed, and uses the standard web services (communication protocols and ports, web browsers, etc.) via the Internet. The RLs have been designed and implemented, using the ISES WEB Control Kit in 2002 by the development team at Charles University in Prague led by Assoc. Prof. Dr. František Lustig. Since 2014 RLs have been significantly improved both in Prague and later at Tomas Bata University in Zlín [18][19][20]. The improvement, related to the ISES Measureserver (MS) that is the basic unit of every RL, working as a finite-state machine (FSM), is subject of this doctoral thesis. Because of the implemented substantial improvements of the ISES MS, we will call this new type as ISES Intelligent Measureserver unit.

3.1 ISES remote laboratory history and basic working scheme

The starting version of the ISES RLs has been developed at Charles University in Prague led by Assoc. Prof. Dr. František Lustig in 2002. The software components have been programmed by MSc. Jiří Dvořák. It has been since constantly improved both at Charles University in Prague and Tomas Bata University in Zlín implementing features for the new user environment, Easy Remote Laboratory (ER-L), to simplify a process of the designing, building and maintaining RLs by a laymen. The ER-L also provides the RLs data archiving, MS diagnostics and embedding the simulations.

The ISES RL concept consists of five integral units as the HW 1) physical components (ADDA signal converter, ISES control board, physical modules categorized as meters, sensors, probes and specific devices), and units as the software 2) Measureserver, 3) Imageserver, 4) Webserver and 5) Webclient. More technical details and applications are available in [21][22][23].

All the built ISES RLs were recently integrated into Remote Laboratory Management System (RLMS) called REMLABNET.

3.1.1 ISES remote laboratory units

The ISES RL concept is based on the autonomous hardware and software exploited for the monitoring, controlling and processing to deliver real experiments residing in laboratories to connected clients for their educational purposes. The following subsection deals with a description of these units.

Physical hardware and software

The physical HW serves the standard ISES system. It is a modular platform based on three basic components. As the first component, it is a set of the physical modules like meters, sensors and devices, which are used for physics, biology, chemistry and electro-engineering. The physical modules are wired to the ISES control board involved as the second component. It transfers signals to the PCI ADDA convertor (interface card), as to the last component. As the example, the physical components, which represent the ISES RL "Sound laboratory", are shown in Figure 2. There are installed and running the sound generator, voltage source, relay board, control board with computer, and apparatus with speaker and two microphones.

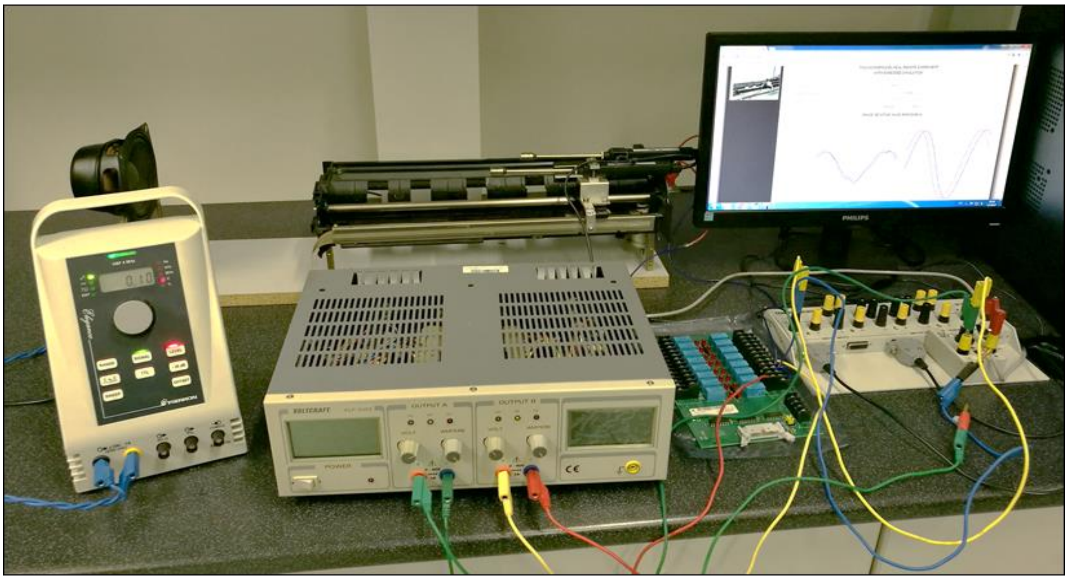


Figure 2 Example of the ISES RL "Sound laboratory" including the sound generator, voltage source, relay board, control board, computer, and apparatus with speaker and two microphones

Informatics hardware and software

The informatics HW serves a personal computer/notebook and network infrastructure to access the RL to connected clients. The software component part constitutes the specific units to ensure the ISES RL online.

Measureserver unit

The MS is a significant software component of the ISES RL concept. It is the processing and communication server located between the physical HW (rig) and remote clients. The MS core is designed as a deterministic finite-state machine [24] to setup and perform all the logical and maintaining instructions for solving the prescribed activities. Its functioning is controlled by the concise program script (psc) file loaded before the unit operation.

With respect to the physical HW, the MS in reality communicates with the PCI ADDA interface card. This is the entirely digital process based on the direct reading of data (real values) from particular pins of the physical sensors and modules, writing data to respective pins, which are translated by the ADDA signal converter. These data pins are both inputs and outputs located on the control board allowing the access to both physical modules (meters, sensors and devices) and providing signals for controlling of experiments.

Instructions (specific commands), coming from a remote client, are processed by the listening MS (receiving service requests from connected clients through preset port). The communication is realized by standard protocols via the Internet. Some commands go via the MS translator to the REMLABNET where clients can exploit additional services for the experimentation.

All the commands (given by psc file) are processed in MS a deterministic way by two different parsers. The first is called the LR(1) parser that processes commands from the configuration file for the purpose of the graphical user interface settings. This parser is based on static state transition tables (parsing tables), which codify a given language grammar. These parsing tables are parameterized together with a lookahead terminal that establishes the maximum tokens, the parser can use to decide, which rule it should use.

The second is the Recursive descent parser that processes commands coming from the psc file to create defined data structures and logic schemes for the RL. It uses a general form of top-down parsing where backtracking may be involved. The parsing algorithm is based on the walking through a tree.

Imageserver unit

The Imageserver unit transports image information to connected clients. This unit periodically stores snaps (stream) acquired by the webcam, which are then directly displayed by a special object to the client’s screen.

Webserver unit

The Webserver unit provides the web (Nginx) services when client enters a web page of the ISES RL via the REMLABNET platform. The Nginx is an open source reverse proxy server for TCP, HTTP and others.

Webclient unit

The Webclient unit represents web pages accessible via the Internet allowing clients to work with the RLs. It was designed for two separate websites, for classic PCs and mobile devices (mobile phones and tablets).

3.2 ISES Measureserver unit and its communication

The MS is important component of every ISES RL. This unit takes care for the communication between the rig (physical HW represented by the ISES control board and its used meters, sensors and devices) and connected clients. The communication is bidirectional, from the rig's side to the client to obtain measured data and back from the client to the rig to control it [25][26].

3.3 Working scheme of ISES Measureserver unit

The ISES MS, as the core software unit, is consisted of several internal and external modules, which are essential for the RL control and monitoring, data gathering and processing to provide results to connected clients. The schematic overview illustrating typical components is shown in Figure 3.

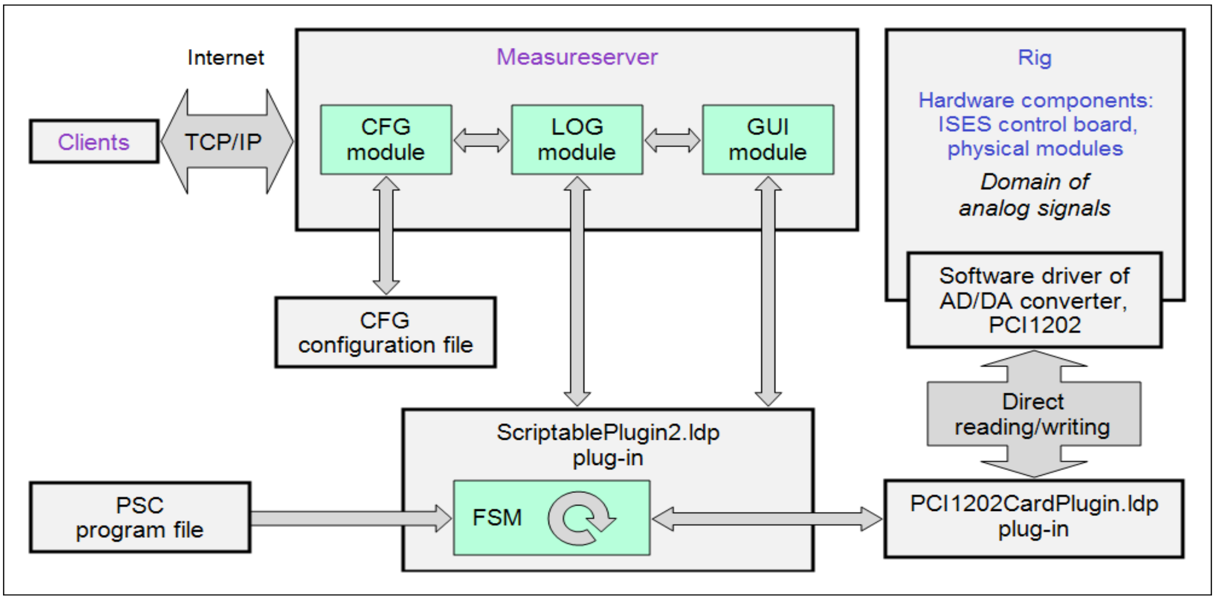

Internet
Clients
TCP/IP
Measureserver
CFG module
LOG module
GUI module
CFG configuration file
Rig
Hardware components: ISES control board, physical modules
Domain of analog signals
Software driver of AD/DA converter, PCI1202
Direct reading/writing
ScriptablePlugin2.ldp plug-in
FSM
PSC program file
PCI1202CardPlugin.ldp plug-in

Figure 3 Scheme of the internal and external modules constituting the MS unit that mediates the connection between the rig and remote clients

Configuration module

The first internal module is called CFG (configuration) intended for the initial setup of the MS behavior and the graphical user interface. During the unit startup phase, the CFG sets all the parameters needed for the operation.

Data logging module

The second internal module is called LOG (data log) that is responsible for the logging of used pins (located on the ISES control board) to files.

Graphical user interface module

The third module is called GUI (graphical user interface) also determined for the RL designers and administrators to easily configure the MS unit.

External modules

The remaining modules are called ldp plugins, which are external by reason of their deployment according to types of the signal converters and rigs. They provide a fixed set of functions, for example, mathematical operations.

Control program

The psc control program is stored in a file with the extension psc that contains the complete logic according to which the RL operates. There is also the section loading single devices (pins), which are associated with specific widgets (small functional objects) placed on client's web pages.

3.4 Easy Remote Laboratory expert system tool

The ER-L is a new user environment exploited to simplify a process of the designing, building and maintaining the RLs by laymen. The expert system tool has been developed since 2012 at Tomas Bata University in Zlín by Michal Krbeček, Ph.D. The ER-L provides three designing categories according to their complexity; 1) Starting level, 2) Basic level, and 3) Advanced level [27].

3.5 RLMS REMLABNET

The REMLABNET is a platform that integrates and manages the RLs for starting university level and secondary schools. Its building was initiated both from the extensive use and expertise in ISES, the built RLs and the lack of similar systems for secondary schools and universities in Europe. Another added value of REMLABNET is better reliability of the RLs. It has been developed since 2014 at Tomas Bata University in Zlín as the team project.

4. OBJECTIVES OF THE THESIS

The goals of this doctoral thesis are focused on the software improvements and development related to the MS unit and its components given below:

1. **Measured data archiving to structured data files -** It is responsible for the gathering, filtering and saving measured data and experiment metadata in the specific data format to an xml file, including the files dispatchment to the data warehouse to archive and for the detailed analysis,
2. **Continuous clients' activities monitoring to text files -** It allows logging of all the activities performed by connected clients, when experimenting, and saving to a log file, with an option of the files dispatchment to the data warehouse for the purpose of didactical analysis,
3. **First level diagnostics of the integrated software modules -** It provides the remote laboratory administrators the notifying and warning messages to ensure the Measureserver unit functioning with the aim of avoiding or reducing occasional failures caused by various influences,
4. **Second level diagnostics of the physical hardware modules -** It gives the remote laboratory administrators the features to detect and monitor the physical modules, like meters, sensors, probes and devices, connected to the rig to prevent their failures or disconnections,
5. **Embedded simulation of real-world phenomena and its visualization with the control on the client’s web page -** It represents the simulation approach running concurrently and synchronized with the real experiment by the integrated solvers to provide approximate numerical solutions used for a motivation before the real measurement as the introductory step to better acquaint the measured phenomenon.

5. SELECTED PROCESSING METHODS

This chapter deals with the theoretical background needed for understanding particular principles, concepts and ideas used for the implementation of proposed goals in the practical part of this doctoral thesis. There are important fields, related to the controlling the behavior of experiments, which include the input grammar concept and the grammar parsing principle, leading to the data measurement process, data distribution and analysis [28][29][30].

5.1 Finite-state machine classification and concept

There are many ways of controlling the behavior of systems, and the use of state machines is one of the oldest and best known. State machines allow us to think about the “state” of a system at a particular point in time and characterize the behavior of the system based on that state. The use of this controlling technique is not limited to the development of specific systems.

This technique can be used to design software systems by identifying what states the system can be in, what inputs or events trigger state transitions, and how the system will behave in each state. In this model, there is seen the execution of the software as a sequence of transitions that move the system through its various states [31]. Concepts and techniques are described in the next subchapters to better understand the FSM models implemented inside of the MS unit as the LR(1) parser and Recursive descent parser.

5.2 Language parsers

A natural language parser is a program that works out the grammatical structure of sentences, for instance, which groups of words go together and which words are the subjects or objects of a verb.

Parsing, or more formally, syntactic analysis, is the process of analyzing a text, made of a sequence of tokens (for example, words), to determine its grammatical structure with respect to a given formal grammar [32].

LR(1) parser

The first concept is a canonical LR parser more frequently called the LR(1). It has been designed to use the bottom-up parsing approach. The theoretical background of the bottom-up LR(1) parser is based on the formal mathematics behind this FSM (parsing machine) operation that reads lexical symbols from a source sentence provided as its input, and it proceeds to recognize productions comprising a particular grammar. It means, a table-driven algorithm can be produced for any given grammar (such as a computer programming language), which parses and recognizes valid sentences [33].

Recursive descent parser

As the second functional concept used is the Recursive descent parser that is a kind of the top-down process built from a set of mutually recursive procedures (or a non-recursive equivalent) where each such procedure usually implements one of the productions of the grammar. Thus the created structure of the resulting program closely mirrors that of the grammar it recognize [34].

5.3 Context-free grammars

Context-free grammars are a recursive representation of the context-free languages, which are a larger class of regular languages. These grammars play a substantial role in the compiler technology longer time. They turned the implementation of parsers from an ad-hoc time-consuming implementation task to a routine job that can be performed very quickly.

5.4 ISES Measureserver unit and its psc control script

The RL functioning has been since the start of ISES RL based on the psc control program and MS. This approach requires creation of the programmable script (executable code running without a compilation) that addresses inputs, outputs and logic of the ISES RL. The script is very condensed and addressing ability rich because it allows programmers to code logical parts, data structures in arbitrary sections in one program. It also allows attaching of plugins (drivers) of specific devices indispensable for the RL functioning.

5.5 Grammar syntax of psc script

The script language grammar is parsed by the Recursive descent parser to build the experiment behavior, data communication and logic. The language developed for the purpose is superficially similar to the C language. Its leading difference compared to the C language is that it is not compiled and functions interpreter-like. This feature is an advantage for the RLs setup and its modification because of avoiding the use of a compiler and its more complex settings. This approach allows better, faster and more effective work.

The manual design and creation of the psc control program possessed of two versions, which mainly differ in the approach of programming.

Old psc script version

Users had to write all the script without any syntax errors and save it as a psc file. The creation of the psc script (version 1.0) was very demanding as users had to know all constituent parts of the system and all relationships.

New psc script version

Since the old script version obstructed the desired dissemination of the RLs to more designers and teachers, the script concept was improved towards the simplified script version. The resulting key improvement was the support for the faster, reliable FSM control and the accessibility of the fast experiments building. The new psc script (version 3.1 and higher) was implemented and encapsulated to the ScriptablePlugin2.ldp plugin.

5.6 Libraries and functionalities of psc script

The MS unit has been designed and developed in the C++ programming language that allows using standard and extended libraries simplifying creation of the application and ensuring its proper operation [35].

The second set of libraries and functions is intended for own functioning of the MS to start up and run a given RL. This specific set is encapsulated in the ScriptablePlugin2.ldp plugin that is attached to the MS during its startup. This plugin is the strongest and the most difficult by reason of its wide use.

5.7 Analysis of Measureserver functioning and data traffic

The MS unit is a Windows application designed and built as a server that is responsible for processing and transferring of measured data and metadata between the rig's physical modules and connected clients.

Before the MS starts working, the unit has to initialize several components ensuring its proper functioning. The components are loaded as the files from various directories and processed by the respective parsers.

When the initialization is successfully competed and the psc script is parsed by the respective parser, the ScriptablePlugin2.ldp plugin initializes and starts all the components for the purpose of the experiment operation.

The MS is able to receive or send data in any state in which the experiment resides. The direct communication between the MS and all the clients is realized by the WebSocket protocol that uses the TCP/IP connection [36].

6. RESULTS

In this chapter are presented the results of the envisaged goals of the doctoral thesis in the Chapter 4 using the methods summarized in Chapter 5. This chapter is constituted by the extensive programming work of the internal grant project IGA of the group of Ph.D. students at Tomas Bata University in Zlín and the partial fulfillment of the project SCOPES of the Swiss National Science Foundation and Swiss Agency for Developments and Cooperation.

For the purpose of introducing into results, there is presented the general scheme of the MS unit shown in Figure 4 with new added modules, which are denoted by the red frame. These modules are called DAM, MUD, PMD and EPS described at some length in the next subchapters.

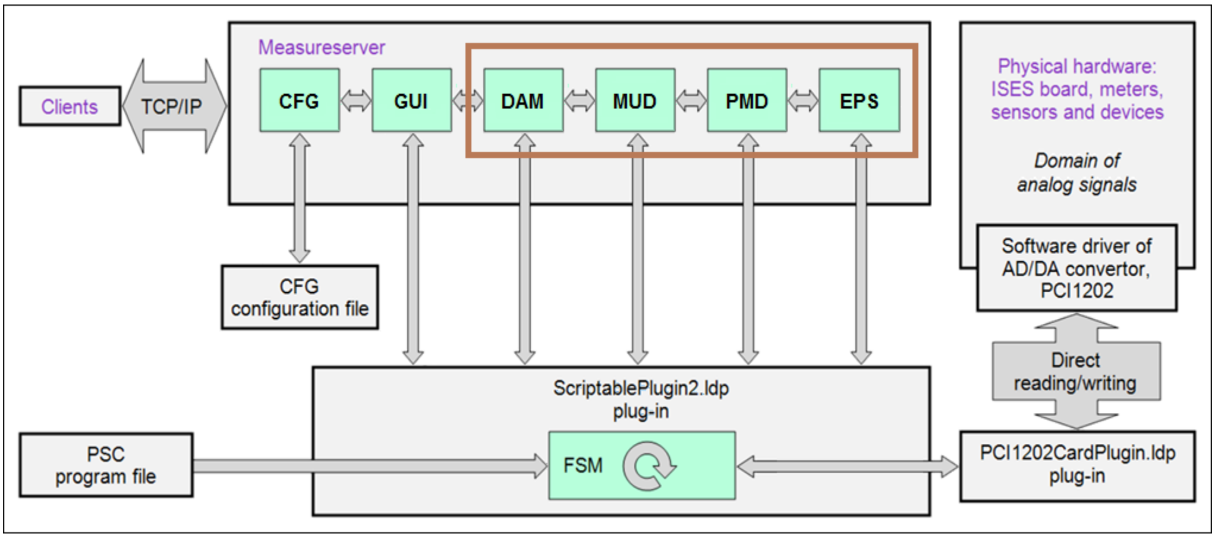

Clients
TCP/IP
Measureserver
CFG
GUI
DAM
MUD
PMD
EPS
Physical hardware: ISES board, meters, sensors and devices
Domain of analog signals
Software driver of AD/DA convertor, PCI1202
CFG configuration file
Direct reading/writing
ScriptablePlugin2.ldp plug-in
PSC program file
FSM
PCI1202CardPlugin.ldp plug-in

Figure 4 Scheme of the MS concept with the new modules located in the red frame; 1) DAM is data archiving management, 2) MUD is Measureserver unit diagnostics, 3) PMD is physical modules diagnostics, and 4) EPS is embedded real-world phenomena simulation

6.1 Data archiving and logging module

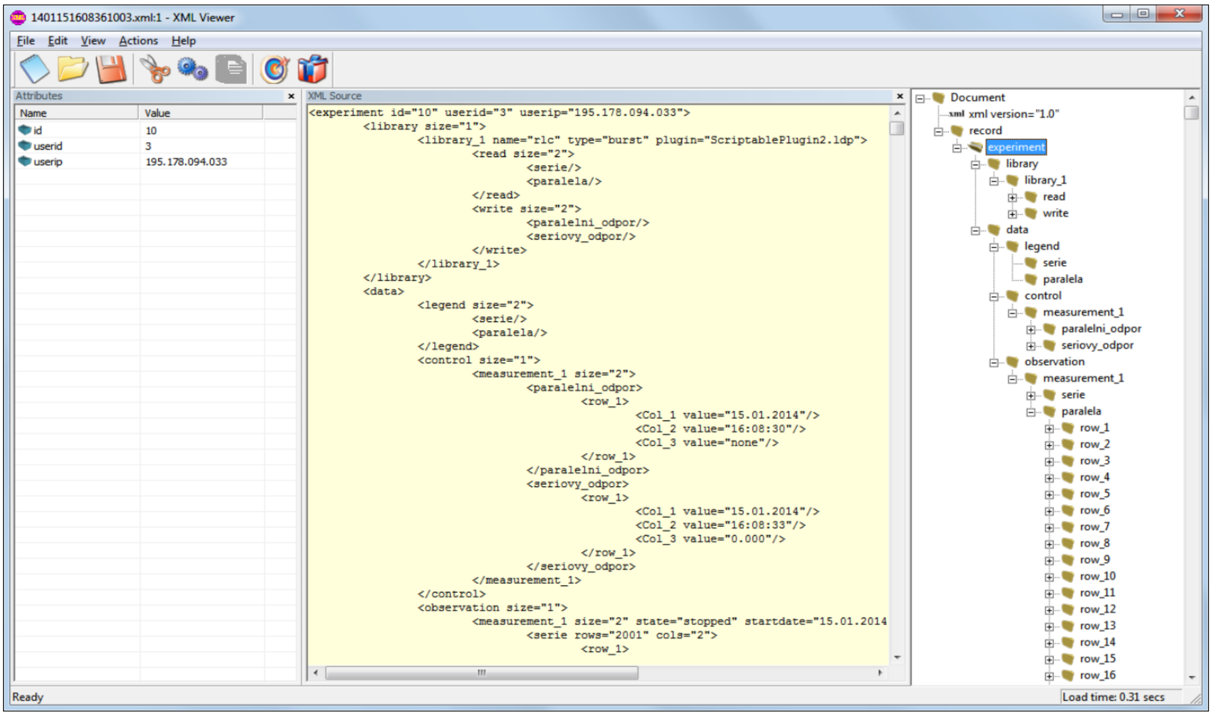
The first task of this doctoral thesis was to design and implement the module called DAM (data archiving management) responsible for gathering, filtering and saving measured data and experiment metadata to an xml file. The module also allows an option to log all performed activities by the connected clients during their experimentations to a log file. The reason to implement this module was to have complete information about the RLs operation and clients whose interventions (their behavior) affect the experimental process.

The generated xml and log files are being sent to the data warehouse where they are analyzed in detail and distributed to other subsystems. The xml files are provided to clients who require measured data coming from typical cases without the real experimenting to reduce time to obtain the measured data.

The DAM exploits for the purpose of measured data archiving the XML (eXtensible Markup Language) data format. The XML is a markup language that defines a set of the rules for encoding documents in a format that is both human-readable and machine-readable [37].

For the implementation, the Microsoft XML Core Services (MSXML) was exploited to speed up its development. The MSXML provides a set of the services to build high-performance XML based applications [38].

As the example, the 1401151608361003.xml file archiving measured data is presented in Figure 5 that includes the data structure and the respective selected branch attributes. The filename is consisted of the date and time stamp of the performed measurement, client's identifier and sequence identifier.



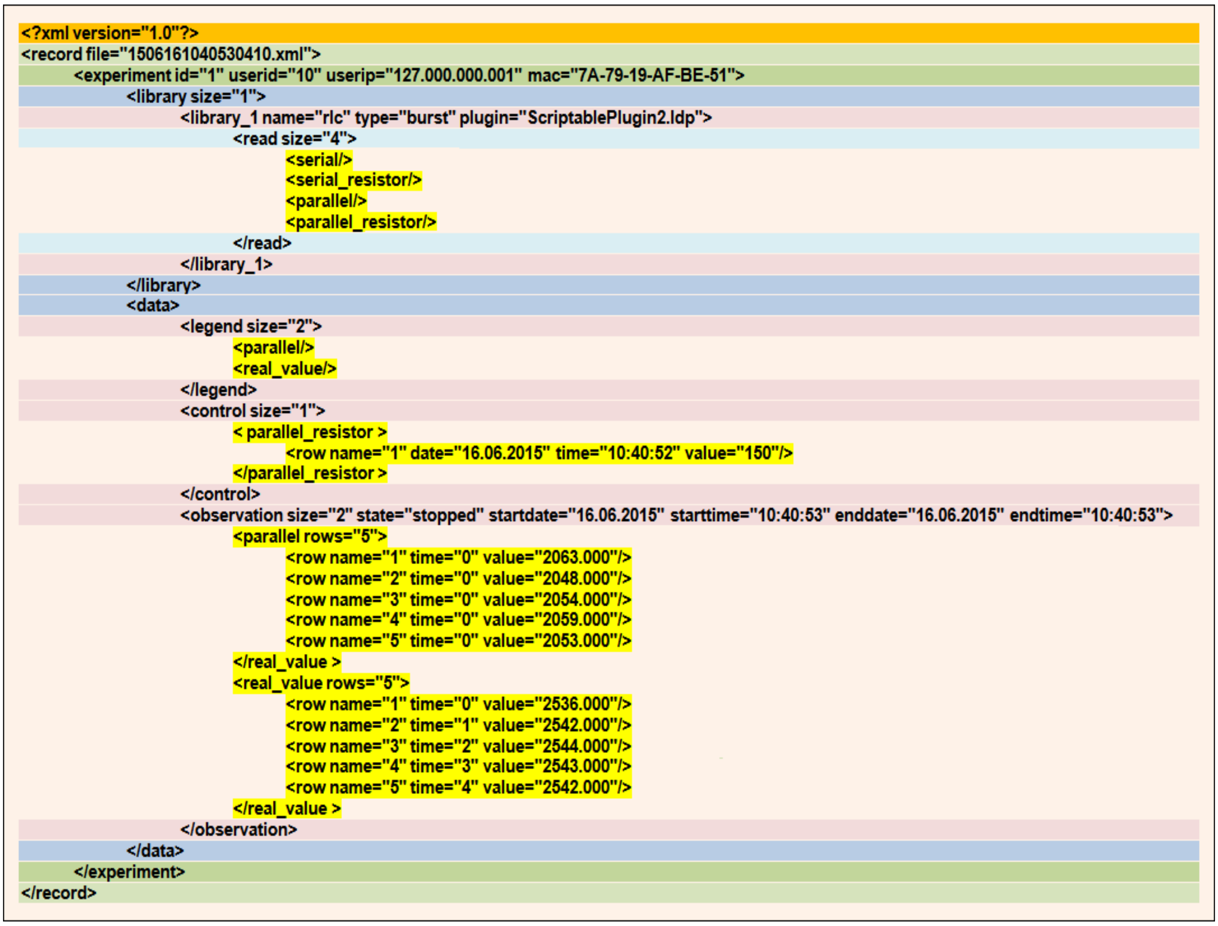
Name	Value
id	10
userid	3
userip	195.178.094.033

```
<experiment id="10" userid="3" userip="195.178.094.033">
        <library size="1">
                <library_1 name="rlc" type="burst" plugin="ScriptablePlugin2.ldp">
                        <read size="2">
                                <serie/>
                                <paralela/>
                        </read>
                        <write size="2">
                                <paralelni_odpor/>
                                <seriovy_odpor/>
                        </write>
                </library_1>
        </library>
        <data>
                <legend size="2">
                        <serie/>
                        <paralela/>
                </legend>
                <control size="1">
                        <measurement_1 size="2">
                                <paralelni_odpor>
                                        <row_1>
                                                <Col_1 value="15.01.2014"/>
                                                <Col_2 value="16:08:30"/>
                                                <Col_3 value="none"/>
                                        </row_1>
                                </paralelni_odpor>
                                <seriovy_odpor>
                                        <row_1>
                                                <Col_1 value="15.01.2014"/>
                                                <Col_2 value="16:08:33"/>
                                                <Col_3 value="0.000"/>
                                        </row_1>
                                </seriovy_odpor>
                        </measurement_1>
                </control>
                <observation size="1">
                        <measurement_1 size="2" state="stopped" startdate="15.01.2014
                                <serie rows="2001" cols="2">
                                        <row_1>
```

```
Document
    xml version="1.0"
    record
        experiment
            library
                library_1
                    read
                    write
            data
                legend
                    serie
                    paralela
                control
                    measurement_1
                        paralelni_odpor
                        seriovy_odpor
                observation
                    measurement_1
                        serie
                        paralela
                            row_1
                            row_2
                            row_3
                            row_4
                            row_5
                            row_6
                            row_7
                            row_8
                            row_9
                            row_10
                            row_11
                            row_12
                            row_13
                            row_14
                            row_15
                            row_16
```

Figure 5 Example of the XML file that is generated by the DAM's functions; the file covers the measured data, metadata and control values affecting the measurement. The viewer presents the three panels, the Attributes panel showing attributes of the "experiment" branch, the XML source panel listing the data structure and the branches tree panel

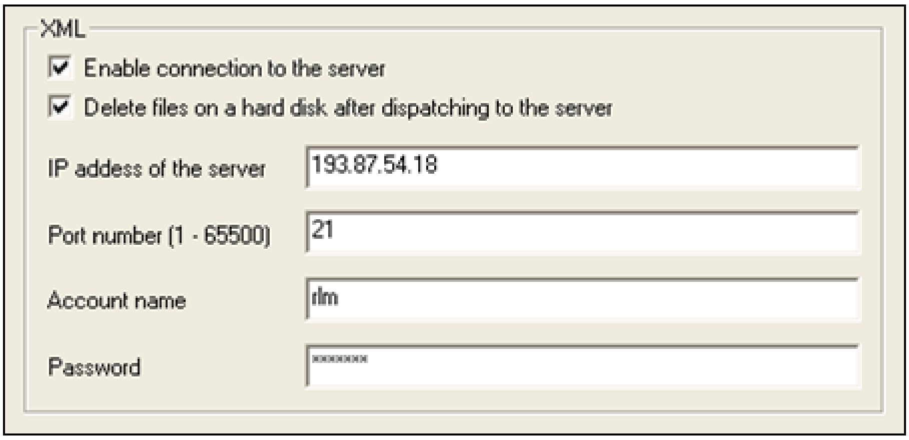
Further example of the archive file is introduced in Figure 6 that exposes a content of the 1506161040530410.xml file presenting the particular branches as the subsection elements. The file content is highlighted by the colored lines to distinguish respective subsections in the data structure.



```
<?xml version="1.0"?>
<record file="1506161040530410.xml">
        <experiment id="1" userid="10" userip="127.000.000.001" mac="7A-79-19-AF-BE-51">
                <library size="1">
                        <library_1 name="rlc" type="burst" plugin="ScriptablePlugin2.ldp">
                                <read size="4">
                                        <serial/>
                                        <serial_resistor/>
                                        <parallel/>
                                        <parallel_resistor/>
                                </read>
                        </library_1>
                </library>
                <data>
                        <legend size="2">
                                <parallel/>
                                <real_value/>
                        </legend>
                        <control size="1">
                                < parallel_resistor >
                                        <row name="1" date="16.06.2015" time="10:40:52" value="150"/>
                                </parallel_resistor >
                        </control>
                        <observation size="2" state="stopped" startdate="16.06.2015" starttime="10:40:53" enddate="16.06.2015" endtime="10:40:53">
                                <parallel rows="5">
                                        <row name="1" time="0" value="2063.000"/>
                                        <row name="2" time="0" value="2048.000"/>
                                        <row name="3" time="0" value="2054.000"/>
                                        <row name="4" time="0" value="2059.000"/>
                                        <row name="5" time="0" value="2053.000"/>
                                </real_value >
                                <real_value rows="5">
                                        <row name="1" time="0" value="2536.000"/>
                                        <row name="2" time="1" value="2542.000"/>
                                        <row name="3" time="2" value="2544.000"/>
                                        <row name="4" time="3" value="2543.000"/>
                                        <row name="5" time="4" value="2542.000"/>
                                </real_value >
                        </observation>
                </data>
        </experiment>
</record>
```

Figure 6 Example of the XML structure that presents the 1506161040530410.xml file to point out the subsection elements. The file comes from the RL "Transients in RLC circuits". The content is highlighted by the colored lines to distinguish particular subsections; the identical colored line pairs indicate subsections at the same level. The yellow elements, "control" and "observation", expose the control value and two measured buffers

The DAM module also has the user interface that provides activating of the direct connection to the data warehouse and the file deletion after dispatching. It allows setting the IP address, port number, account name and password to establish the communication with the data warehouse. The user interface is presented in Figure 7 as the XML section residing in the Diagnosis dialog box belonging the MS's GUI used for the general setup.



XML

☑ Enable connection to the server

☑ Delete files on a hard disk after dispatching to the server

IP addess of the server: 193.87.54.18

Port number (1 - 65500): 21

Account name: rlm

Password: *******

Figure 7 Example setup of the user interface that presents the XML parameters used for the direct connection to the data warehouse to dispatch generated xml files

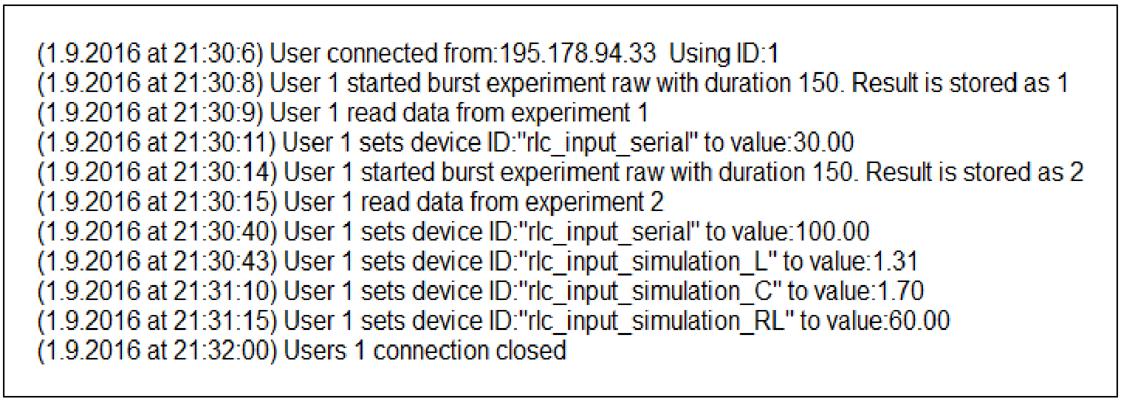
The DAM also provides a component for the clients' activities logging. The component is responsible for continuous writing events reporting activities of the connected clients. The MS assigns a unique identifier for every client to distinguish his activities. These events are being saved to the Users.log file for all the time the client is connected to the system.

The clients' activities logging process is designed to run in several sections of the MS to log respective events. It is deployed in the section where the clients connect/disconnect or change their status. Further logging process is involved in the section for clients' interventions (changing slider positions, button states and other widgets to control the experiment or modify options). These events are especially interesting for the RL designers in the didactical aspects.

When a new client successfully connects from his web browser to the MS, the logging component makes a record (date, time, IP address and client's unique identifier) about this event to the same log file.

Furthermore, the module is able to record events incoming from the MS functioning like its startup, shutdown, loaded device plugins and data pins allocated on the control board of respective experiment.

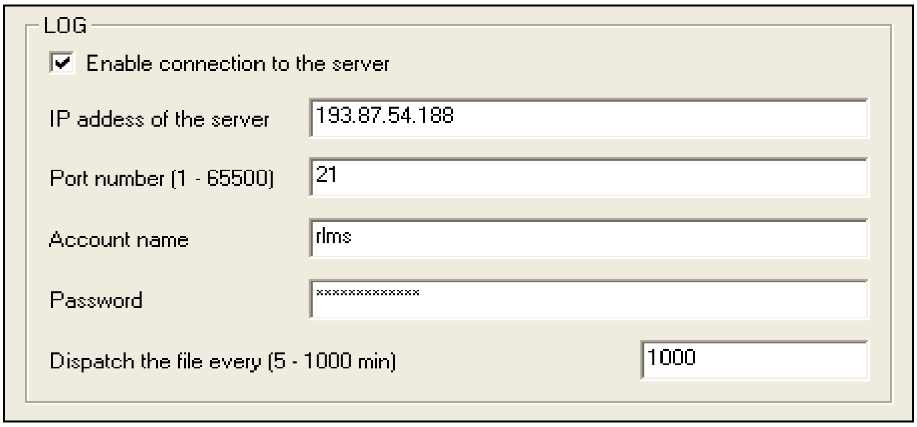
The example record stored in the Users.log file is presented in Figure 8 that exposes the date and time stamp, client's unique identifier assigned when he is connected to distinguish his activities. There are also the records "sets device" indicating the changed parameters for the real and simulated process.



```
(1.9.2016 at 21:30:6) User connected from:195.178.94.33  Using ID:1
(1.9.2016 at 21:30:8) User 1 started burst experiment raw with duration 150. Result is stored as 1
(1.9.2016 at 21:30:9) User 1 read data from experiment 1
(1.9.2016 at 21:30:11) User 1 sets device ID:"rlc_input_serial" to value:30.00
(1.9.2016 at 21:30:14) User 1 started burst experiment raw with duration 150. Result is stored as 2
(1.9.2016 at 21:30:15) User 1 read data from experiment 2
(1.9.2016 at 21:30:40) User 1 sets device ID:"rlc_input_serial" to value:100.00
(1.9.2016 at 21:30:43) User 1 sets device ID:"rlc_input_simulation_L" to value:1.31
(1.9.2016 at 21:31:10) User 1 sets device ID:"rlc_input_simulation_C" to value:1.70
(1.9.2016 at 21:31:15) User 1 sets device ID:"rlc_input_simulation_RL" to value:60.00
(1.9.2016 at 21:32:00) Users 1 connection closed
```

Figure 8 Example record of the clients' activities located in the Users.log file to archive to the data warehouse

The clients' activities logging component has the respective user interface to provide activating of the connection to the data warehouse. Furthermore, it allows setting the IP address, port number, account name and password placed for the communication with the data warehouse. The last included parameter is defined for the dispatching periodicity as introduced in Figure 9.



☑ Enable connection to the server

IP addess of the server: 193.87.54.188

Port number (1 - 65500): 21

Account name: rlms

Password: *************

Dispatch the file every (5 - 1000 min): 1000

Figure 9 Example setup of the user interface that presents the LOG parameters used for configuring of the clients' activities logging component

Summary of the subchapter "6.1 Data archiving and logging module": This subchapter fulfils the goal No. 1 and 2. The remote laboratories produce excessive, even big data, produced by measured ones, which are generated by measuring process and clients' activities data that should be stored for the later recovery and processing. This constitutes the improvement in the intelligent version of the Measureserver unit. The novelty and the scientific approach rest in the use of XML data output for the online analysis.

6.2 First level diagnostics module

The second task of the doctoral thesis was to design and implement the module called MUD (Measureserver unit diagnostics) used as the first level diagnostics. The module provides to RL administrators information to ensure the MS functioning with the aim of avoiding or reducing occasional failures caused by various influences. The MUD also has a self-recovery mechanism to recover the MS functioning after failure events. The typical failure is caused by attached device plugins, which can suddenly stop communicating with their ADDA interface cards or physical modules. The diagnostics is optional that can be disabled to avoid the compatibility conflict with older experiments.

The MUD concept is based on the checking operational sections (blocks) inside the MS core including limited corrections when the evaluating conditions detect failure states in the checked sections. This module is consisting of two internal components working independently in different threads.

6.2.1 Notifying thread component

The first built component called RlmsNotifyingThread is responsible for the notification process that periodically monitors the MS functioning. If the MS status is detected okay (it works properly), the MUD dispatches the notification message "Measureserver is running" to the diagnostic server (DS). This is a superior server that processes all the messages incoming from the MS and sends back commands to request for the RL status or to recover the operation. In case of the mentioned notification message, the DS identifies this message and exposes as the three lights semaphore placed on every web page of the RLs in the REMLABNET. The semaphore is especially useful for connected clients who can see three colors on it; the red one indicates the RL is unavailable, the orange indicates it is reserved and the green signalizes it is ready to use.

The MUD generates all the messages in the same text format before their dispatching to uniquely distinguish them in the DS. Each message contains identifying details, which are placed in square parenthesis. For example, the message "Measureserver is running" is generated with the appended items and dispatched to the DS in the following text form:

[10.05.2017] [15:20:30] [7A-79-19-25-10-1C] [192.168.1.100] [LAB] [RLC]
[0] [114] [Measureserver is running].

6.2.2 Operating thread component

The second component build in the MUD is called RlmsOperatingThread used for various diagnosing operations in the MS core. This component is implemented as a function running periodically to process all messages.

This specific function is designed for three targets. The first implemented target is responsible for the dispatching of the generated diagnostic messages to the DS for the analysis and the distribution to other subsystems.

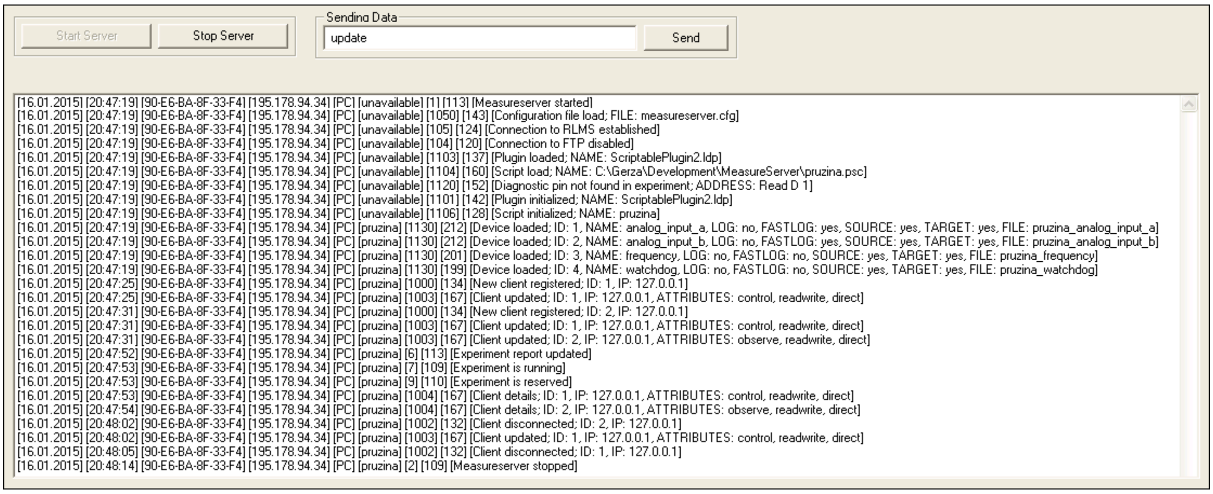
The second target is focused on receiving the messages from the DS as the commands to accomplish the prescribed actions in successive steps.

The third target deals with the accomplishment of the commands to deliver status information to the RL administrators; these commands are:

1. *accept* - It returns the message “RLMS connection accepted” to inform about the MS activity whether it runs correctly.
2. *update* - It returns the message “Experiment report updated” with the one of the following messages, “Experiment is idle”, “Experiment is running” or “Experiment is reserved” to inform about the experiment status.
3. *restart* - It performs that the MS restarts itself and then it returns back the message “Measureserver restarted” to inform about the restart.
4. *reboot* - It performs that the computer reboots itself and then it returns back the message “Computer rebooted” to inform about the reboot.
5. *stop* - It performs that the MS stops running to finish the experiment to make the rig’s modules maintenance or recovery.
6. *detect* - It returns the messages informing about the status and setup of connected physical modules. This command relates to the second level diagnostics that is discussed in the next subchapter.

Each command can be accompanied by an IP address as the parameter that determines which RL is concerned to obtain status information from, or to perform required actions on it. If there is not entered the IP address, then all the RLs (connected to the DS) accomplish the incoming commands.

As the example, the “update” command dispatched from the DS to MS unit causes that the MS sends back the list of complex information describing the respective experiment as shown in Figure 10 There is primarily delivered the data and time stamp of the request, the configuration of connected physical modules including attached plugins and registered clients.

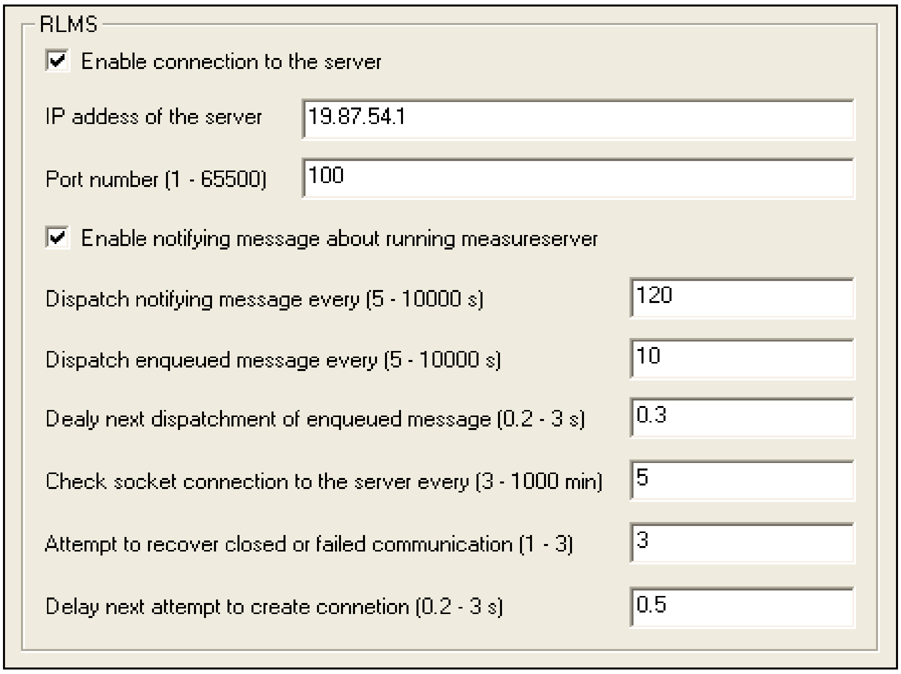


update

```
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [1] [113] [Measureserver started]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [1050] [143] [Configuration file load; FILE: measureserver.cfg]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [105] [124] [Connection to RLMS established]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [104] [120] [Connection to FTP disabled]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [1103] [137] [Plugin loaded; NAME: ScriptablePlugin2.ldp]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [1104] [160] [Script load; NAME: C:\Gerza\Development\MeasureServer\pruzina.psc]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [1120] [152] [Diagnostic pin not found in experiment; ADDRESS: Read D 1]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [1101] [142] [Plugin initialized; NAME: ScriptablePlugin2.ldp]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [unavailable] [1106] [128] [Script initialized; NAME: pruzina]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1130] [212] [Device loaded; ID: 1, NAME: analog_input_a, LOG: no, FASTLOG: yes, SOURCE: yes, TARGET: yes, FILE: pruzina_analog_input_a]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1130] [212] [Device loaded; ID: 2, NAME: analog_input_b, LOG: no, FASTLOG: yes, SOURCE: yes, TARGET: yes, FILE: pruzina_analog_input_b]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1130] [201] [Device loaded; ID: 3, NAME: frequency, LOG: no, FASTLOG: no, SOURCE: yes, TARGET: yes, FILE: pruzina_frequency]
[16.01.2015] [20:47:19] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1130] [199] [Device loaded; ID: 4, NAME: watchdog, LOG: no, FASTLOG: no, SOURCE: yes, TARGET: yes, FILE: pruzina_watchdog]
[16.01.2015] [20:47:25] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1000] [134] [New client registered; ID: 1, IP: 127.0.0.1]
[16.01.2015] [20:47:25] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1003] [167] [Client updated; ID: 1, IP: 127.0.0.1, ATTRIBUTES: control, readwrite, direct]
[16.01.2015] [20:47:31] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1000] [134] [New client registered; ID: 2, IP: 127.0.0.1]
[16.01.2015] [20:47:31] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1003] [167] [Client updated; ID: 1, IP: 127.0.0.1, ATTRIBUTES: control, readwrite, direct]
[16.01.2015] [20:47:31] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1003] [167] [Client updated; ID: 2, IP: 127.0.0.1, ATTRIBUTES: observe, readwrite, direct]
[16.01.2015] [20:47:52] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [6] [113] [Experiment report updated]
[16.01.2015] [20:47:53] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [7] [109] [Experiment is running]
[16.01.2015] [20:47:53] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [9] [110] [Experiment is reserved]
[16.01.2015] [20:47:53] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1004] [167] [Client details; ID: 1, IP: 127.0.0.1, ATTRIBUTES: control, readwrite, direct]
[16.01.2015] [20:47:54] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1004] [167] [Client details; ID: 2, IP: 127.0.0.1, ATTRIBUTES: observe, readwrite, direct]
[16.01.2015] [20:48:02] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1002] [132] [Client disconnected; ID: 2, IP: 127.0.0.1]
[16.01.2015] [20:48:02] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1003] [167] [Client updated; ID: 1, IP: 127.0.0.1, ATTRIBUTES: control, readwrite, direct]
[16.01.2015] [20:48:05] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [1002] [132] [Client disconnected; ID: 1, IP: 127.0.0.1]
[16.01.2015] [20:48:14] [90-E6-BA-8F-33-F4] [195.178.94.34] [PC] [pruzina] [2] [109] [Measureserver stopped]
```

Figure 10 Example of the "update" command dispatched from the DS to MS unit to obtain back the complex information describing the experiment

The MUD is allowed to configure by the respective user interface. It provides the RL administrators the specific parameters to set the connection to the DS and to also set the management of notification and faulty messages during the operation as presented on the example settings shown in Figure 11.



☑ Enable connection to the server

IP addess of the server: 19.87.54.1

Port number (1 - 65500): 100

☑ Enable notifying message about running measureserver

Dispatch notifying message every (5 - 10000 s): 120

Dispatch enqueued message every (5 - 10000 s): 10

Dealy next dispatchment of enqueued message (0.2 - 3 s): 0.3

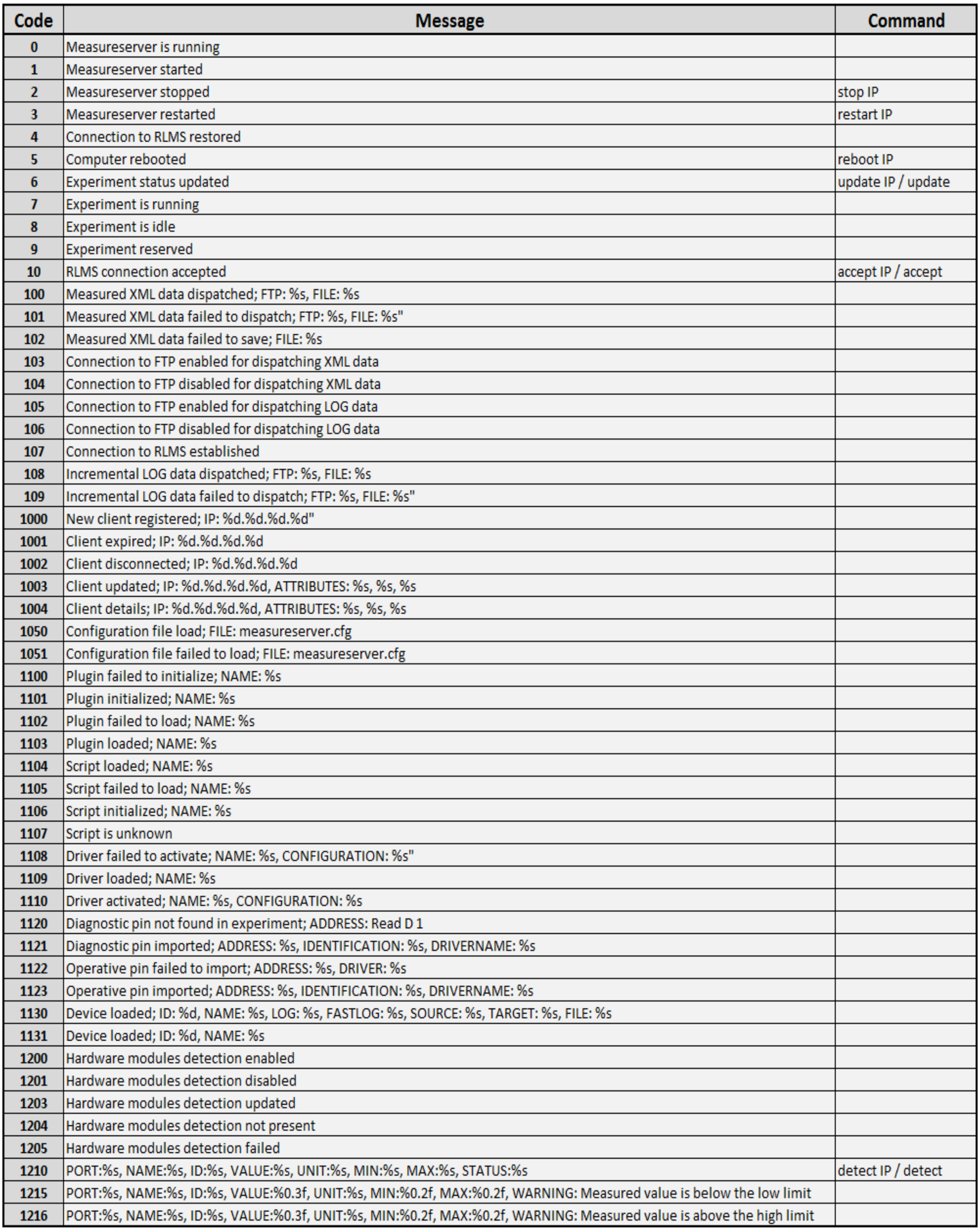
Check socket connection to the server every (3 - 1000 min): 5

Attempt to recover closed or failed communication (1 - 3): 3

Delay next attempt to create connetion (0.2 - 3 s): 0.5

Figure 11 Example settings of the MUD module available in the MS unit to configure the direct connection to the DS and the messaging process

The notification and faulty messages, which are generated by occurred events, and the involved commands for the management are listed in Figure 12.



Code	Message	Command
0	Measureserver is running	
1	Measureserver started	
2	Measureserver stopped	stop IP
3	Measureserver restarted	restart IP
4	Connection to RLMS restored	
5	Computer rebooted	reboot IP
6	Experiment status updated	update IP / update
7	Experiment is running	
8	Experiment is idle	
9	Experiment reserved	
10	RLMS connection accepted	accept IP / accept
100	Measured XML data dispatched; FTP: %s, FILE: %s	
101	Measured XML data failed to dispatch; FTP: %s, FILE: %s"	
102	Measured XML data failed to save; FILE: %s	
103	Connection to FTP enabled for dispatching XML data	
104	Connection to FTP disabled for dispatching XML data	
105	Connection to FTP enabled for dispatching LOG data	
106	Connection to FTP disabled for dispatching LOG data	
107	Connection to RLMS established	
108	Incremental LOG data dispatched; FTP: %s, FILE: %s	
109	Incremental LOG data failed to dispatch; FTP: %s, FILE: %s"	
1000	New client registered; IP: %d.%d.%d.%d"	
1001	Client expired; IP: %d.%d.%d.%d	
1002	Client disconnected; IP: %d.%d.%d.%d	
1003	Client updated; IP: %d.%d.%d.%d, ATTRIBUTES: %s, %s, %s	
1004	Client details; IP: %d.%d.%d.%d, ATTRIBUTES: %s, %s, %s	
1050	Configuration file load; FILE: measureserver.cfg	
1051	Configuration file failed to load; FILE: measureserver.cfg	
1100	Plugin failed to initialize; NAME: %s	
1101	Plugin initialized; NAME: %s	
1102	Plugin failed to load; NAME: %s	
1103	Plugin loaded; NAME: %s	
1104	Script loaded; NAME: %s	
1105	Script failed to load; NAME: %s	
1106	Script initialized; NAME: %s	
1107	Script is unknown	
1108	Driver failed to activate; NAME: %s, CONFIGURATION: %s"	
1109	Driver loaded; NAME: %s	
1110	Driver activated; NAME: %s, CONFIGURATION: %s	
1120	Diagnostic pin not found in experiment; ADDRESS: Read D 1	
1121	Diagnostic pin imported; ADDRESS: %s, IDENTIFICATION: %s, DRIVERNAME: %s	
1122	Operative pin failed to import; ADDRESS: %s, DRIVER: %s	
1123	Operative pin imported; ADDRESS: %s, IDENTIFICATION: %s, DRIVERNAME: %s	
1130	Device loaded; ID: %d, NAME: %s, LOG: %s, FASTLOG: %s, SOURCE: %s, TARGET: %s, FILE: %s	
1131	Device loaded; ID: %d, NAME: %s	
1200	Hardware modules detection enabled	
1201	Hardware modules detection disabled	
1203	Hardware modules detection updated	
1204	Hardware modules detection not present	
1205	Hardware modules detection failed	
1210	PORT:%s, NAME:%s, ID:%s, VALUE:%s, UNIT:%s, MIN:%s, MAX:%s, STATUS:%s	detect IP / detect
1215	PORT:%s, NAME:%s, ID:%s, VALUE:%0.3f, UNIT:%s, MIN:%0.2f, MAX:%0.2f, WARNING: Measured value is below the low limit	
1216	PORT:%s, NAME:%s, ID:%s, VALUE:%0.3f, UNIT:%s, MIN:%0.2f, MAX:%0.2f, WARNING: Measured value is above the high limit	

Figure 12 Notification and faulty messages generated by the MS unit when events occur and the control commands coming from the DS to use for the management of the unit in case of problems or updates

Summary of the subchapter "6.2 First level diagnostics module": This subchapter deals with the goal No. 3. The diagnostics of remote laboratories is not to be found, to our knowledge, in any published laboratory, irrespective of the fact of its importance. The first level diagnostics serves to inform the client about general function of the respective remote laboratory. It is based on checking the consistency of the function of every plugins and drivers, signaled by the "traffic lights". This constitutes the improvement in the intelligent version of the Measureserver unit. The novelty and the scientific approach rest in the use of real-time checking of the critical points of the units in question.

6.3 Second level diagnostics module

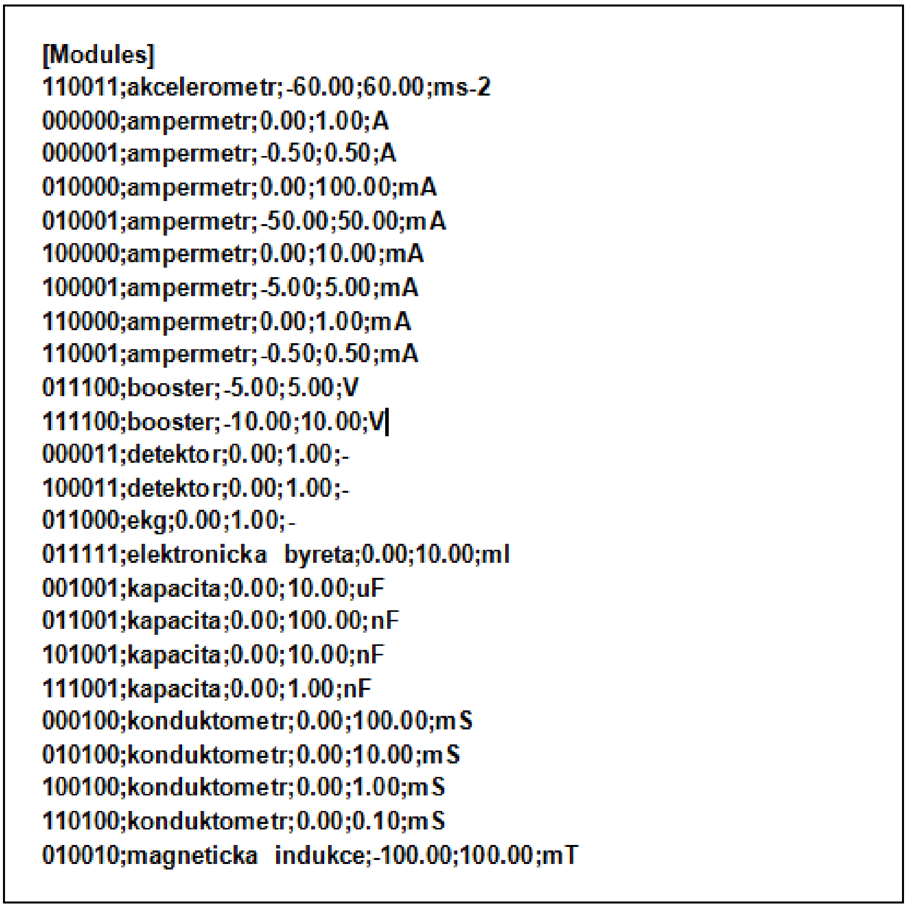
The third task of this doctoral thesis was to design and realize the module called PMD (physical modules diagnostics) used as the second level diagnostics. The module provides the RL administrators the features to detect and monitor the ISES physical modules (meters, sensors, probes and devices) connected to the rig to prevent their failures or disconnections during the measurement.

The physical modules are used to obtain the quantities of electric current and voltage, temperature, speed, sound, etc. They are connected to the ISES control board used as a central board of the RL rig. As the primary aim, the PMD was implemented to detect and determine a kind of the added (attached) module and its configuration, like the number of data channels and sensing range.

The detection principle is based on the periodical testing of input and output ports built in the ISES control board where the physical modules are connected as exchangeable devices. The ADDA conversion is realized by the PCI 1202 interface card via the driver between the MS and the ISES control board. When any physical module is connected, re-connected or removed from the port, then the appropriate technique to update the status is activated.

The PMD is able to provide all the technical parameters of particular physical modules connected to the CAN-15 ports in cooperation with the Module32.map database file as listed in Figure 13 presenting a piece of this file that is parsed by the MS to make module references for the detection process.

The RL administrators can comfortably monitor the updated list of the physical modules displayed in the concerned GUI containing additional details. If any physical module, defined in the psc file, fails during its operation by the reason of unexpected disconnection or even destruction caused by strokes or



```
[Modules]
110011;akcelerometr;-60.00;60.00;ms-2
000000;ampermetr;0.00;1.00;A
000001;ampermetr;-0.50;0.50;A
010000;ampermetr;0.00;100.00;mA
010001;ampermetr;-50.00;50.00;mA
100000;ampermetr;0.00;10.00;mA
100001;ampermetr;-5.00;5.00;mA
110000;ampermetr;0.00;1.00;mA
110001;ampermetr;-0.50;0.50;mA
011100;booster;-5.00;5.00;V
111100;booster;-10.00;10.00;V
000011;detektor;0.00;1.00;-
100011;detektor;0.00;1.00;-
011000;ekg;0.00;1.00;-
011111;elektronicka byreta;0.00;10.00;ml
001001;kapacita;0.00;10.00;uF
011001;kapacita;0.00;100.00;nF
101001;kapacita;0.00;10.00;nF
111001;kapacita;0.00;1.00;nF
000100;konduktometr;0.00;100.00;mS
010100;konduktometr;0.00;10.00;mS
100100;konduktometr;0.00;1.00;mS
110100;konduktometr;0.00;0.10;mS
010010;magneticka indukce;-100.00;100.00;mT
```

Figure 13 Exemplary piece of the physical modules list stored in the Modules.txt file used for the detection

vibrations, then the PMD detects this anomaly and dispatches it to the modules list in the GUI and to the DS interface to analyze.

The PMD is consisted of several components deployed on different places of the MS structure and attached plugins. As the main, there is implemented the HwmdsDetectingThread component responsible for the detection process.

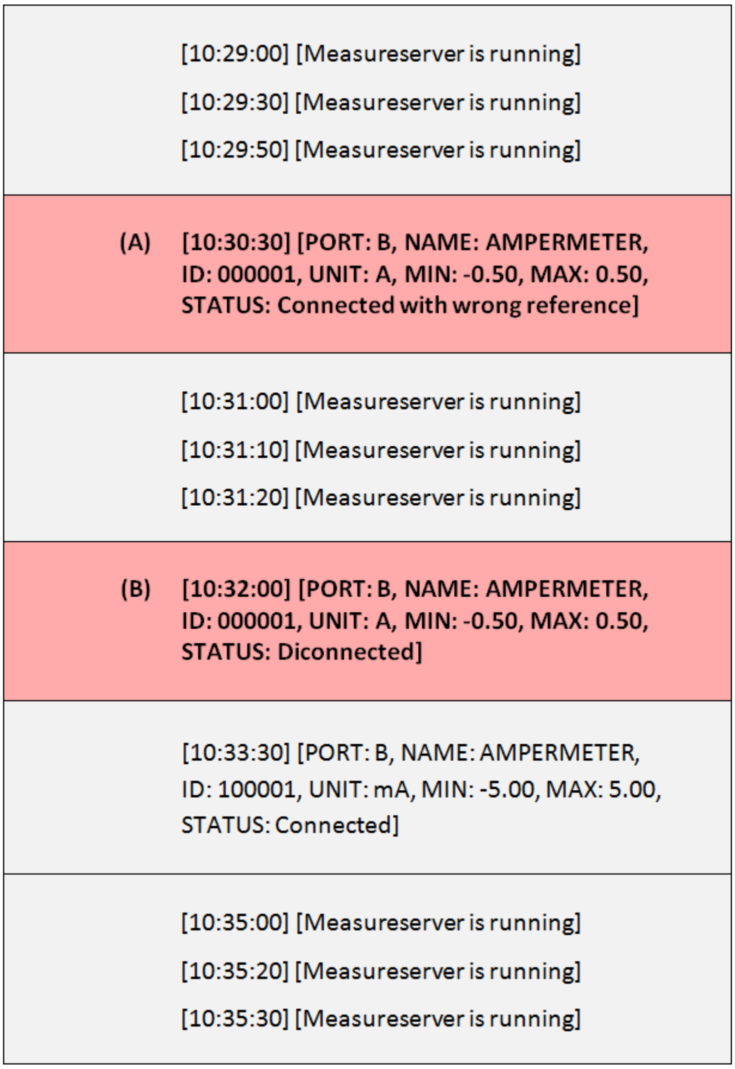
6.3.1 Detecting thread component

This component is handled as the function running in a separate tread, and includes the functions directly connected to the PCI1202CardPlugin.ldp to work with the control board and its used physical modules.

The detection process generates one of the status information for every port as the notification message reporting: 1) Undefined, 2) Just connected, 3) Just disconnected, 4) Connected with the wrong reference, 5) Disconnected with the wrong reference, or else 6) Detected the empty port.

Except the periodical detection process, there is also possible to dispatch the command *detect* (optionally accompanied by an IP address) from the RL administrator to obtain the current status list of the CAN-15 ports.

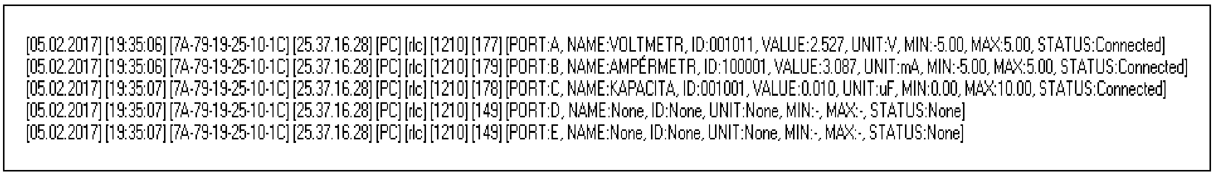
For example, the MUD dispatcher generates and delivers to the DS user interface the periodical notifications related to the MS running and the status messages about the connected physical module as listed in Figure 14.



```
[10:29:00] [Measureserver is running]
[10:29:30] [Measureserver is running]
[10:29:50] [Measureserver is running]

(A)   [10:30:30] [PORT: B, NAME: AMPERMETER,
      ID: 000001, UNIT: A, MIN: -0.50, MAX: 0.50,
      STATUS: Connected with wrong reference]

[10:31:00] [Measureserver is running]
[10:31:10] [Measureserver is running]
[10:31:20] [Measureserver is running]

(B)   [10:32:00] [PORT: B, NAME: AMPERMETER,
      ID: 000001, UNIT: A, MIN: -0.50, MAX: 0.50,
      STATUS: Diconnected]

[10:33:30] [PORT: B, NAME: AMPERMETER,
ID: 100001, UNIT: mA, MIN: -5.00, MAX: 5.00,
STATUS: Connected]

[10:35:00] [Measureserver is running]
[10:35:20] [Measureserver is running]
[10:35:30] [Measureserver is running]
```

Figure 14 Example of the MUD records; correctly running (gray), two failed (red); A) setup changed caused by the range of amperemeter, and B) one amperemeter was disconnected

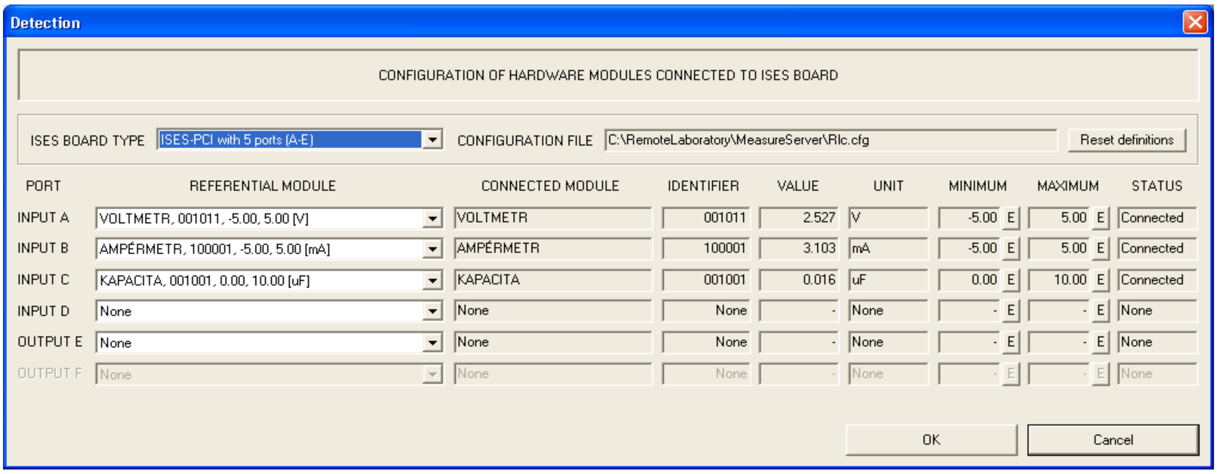
As the next example, if the RL administrator demands the current status of connected physical modules, then he must enter and dispatch the command “detect 10.20.54.66” to the MS unit. After a short time, the MS’s PMD returns the complete list to the DS user interface as shown in Figure 15.



```
[05.02.2017] [19:35:06] [7A-79-19-25-10-1C] [25.37.16.28] [PC] [rlc] [1210] [177] [PORT:A, NAME:VOLTMETR, ID:001011, VALUE:2.527, UNIT:V, MIN:-5.00, MAX:5.00, STATUS:Connected]
[05.02.2017] [19:35:06] [7A-79-19-25-10-1C] [25.37.16.28] [PC] [rlc] [1210] [179] [PORT:B, NAME:AMPÉRMETR, ID:100001, VALUE:3.087, UNIT:mA, MIN:-5.00, MAX:5.00, STATUS:Connected]
[05.02.2017] [19:35:07] [7A-79-19-25-10-1C] [25.37.16.28] [PC] [rlc] [1210] [178] [PORT:C, NAME:KAPACITA, ID:001001, VALUE:0.010, UNIT:uF, MIN:0.00, MAX:10.00, STATUS:Connected]
[05.02.2017] [19:35:07] [7A-79-19-25-10-1C] [25.37.16.28] [PC] [rlc] [1210] [149] [PORT:D, NAME:None, ID:None, UNIT:None, MIN:-, MAX:-, STATUS:None]
[05.02.2017] [19:35:07] [7A-79-19-25-10-1C] [25.37.16.28] [PC] [rlc] [1210] [149] [PORT:E, NAME:None, ID:None, UNIT:None, MIN:-, MAX:-, STATUS:None]
```

Figure 15 Example of the detailed list presenting the current status of connected physical modules to the ports on the ISES PCI control board

The MS provides the RL administrator the PMD user interface that allows monitoring the complete status of the CAN-15 ports. There is an option to preset the referential physical modules and to change the measurement ranges. The configuration and changes are stored to the specific diagnostic file. The user interface presenting the exemplary settings is shown in Figure 16.



CONFIGURATION OF HARDWARE MODULES CONNECTED TO ISES BOARD

ISES BOARD TYPE: ISES-PCI with 5 ports (A-E)

CONFIGURATION FILE: C:\RemoteLaboratory\MeasureServer\Rlc.cfg

PORT	REFERENTIAL MODULE	CONNECTED MODULE	IDENTIFIER	VALUE	UNIT	MINIMUM	MAXIMUM	STATUS
INPUT A	VOLTMETR, 001011, -5.00, 5.00 [V]	VOLTMETR	001011	2.527	V	-5.00 E	5.00 E	Connected
INPUT B	AMPÉRMETR, 100001, -5.00, 5.00 [mA]	AMPÉRMETR	100001	3.103	mA	-5.00 E	5.00 E	Connected
INPUT C	KAPACITA, 001001, 0.00, 10.00 [uF]	KAPACITA	001001	0.016	uF	0.00 E	10.00 E	Connected
INPUT D	None	None	None	-	None	- E	- E	None
OUTPUT E	None	None	None	-	None	- E	- E	None
OUTPUT F	None	None	None	-	None	- E	- E	None

Figure 16 Example settings of the PMD user interface to present the involved referential physical modules and detected information with current status

Summary of the subchapter “6.3 Second level diagnostics module”: This subchapter deals with the goal No. 4. The diagnostics of remote laboratories on this level is not to be found, to our knowledge, in any published laboratory, irrespective of the fact of its importance. The second level diagnostics serves to check the conformity of the physical hardware compared with the standard setup. The checking results, when providing negative results, are forwarded to the remote laboratory provider to recover experiment. This constitutes the improvement in the intelligent version of the Measureserver unit. The novelty and the scientific approach rest in the use of guarding all parameters of physical modules as the function, its range, polarity and signal overload.

6.4 Embedded real-world phenomena simulation module

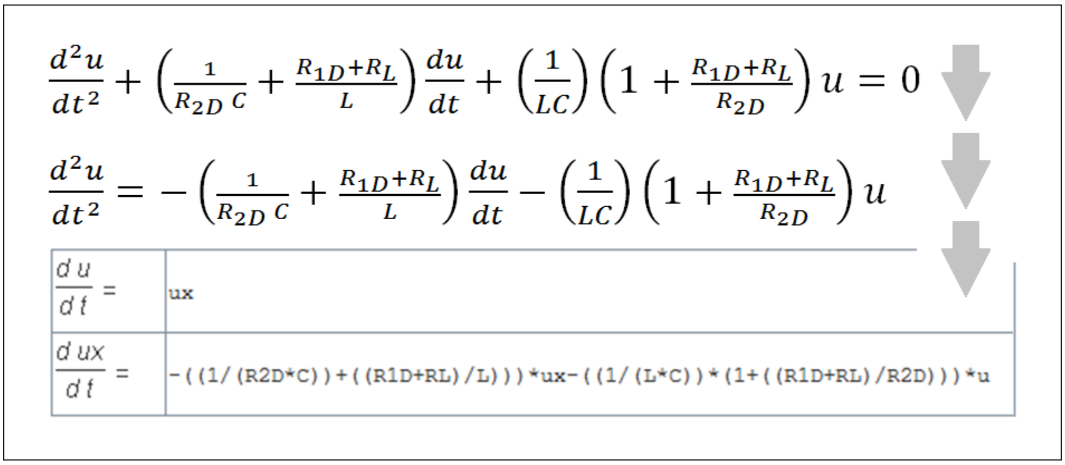
Simulations play an increasingly important role in the way of teaching or doing science. This is especially true in education, where computers are being used more often as a way to make lectures more attractive to users, and to help them achieve deeper understanding of the subject being taught.

This subchapter deals with the design and implementation of the module called EPS (embedded real-world phenomena simulation), as the last task of the doctoral thesis, that can run concurrently and synchronized with the real experiment, and visualized on the client's web page.

The EPS extends the MS's library of mathematical functions. It provides the RL designers advanced numerical function as the solvers to deal with ordinary differential equations (ODEs). There is created the EPS interface to parse and process the first and second order ODE in the analytical form [39] to calculate its solution through the specific iterative functions in the psc script.

The EPS interface was designed to apply the Lua programming language [40] to simply enter (specify) the ODE analytical form coded and saved to the plain text file, called Simulation.lua as the input setting for the solver.

The initial idea of ODEs entering and passing the solver was taken from the Easy Java/JavaScript Simulations (EjsS) environment [41]. As the example, based on this idea, the analytical form of second order ODE can be adjusted and entered on separate lines to process by the solver as shown in Figure 17.



$$\frac{d^2u}{dt^2} + \left(\frac{1}{R_{2D}\,C} + \frac{R_{1D}+R_L}{L}\right)\frac{du}{dt} + \left(\frac{1}{LC}\right)\left(1 + \frac{R_{1D}+R_L}{R_{2D}}\right)u = 0$$

$$\frac{d^2u}{dt^2} = -\left(\frac{1}{R_{2D}\,C} + \frac{R_{1D}+R_L}{L}\right)\frac{du}{dt} - \left(\frac{1}{LC}\right)\left(1 + \frac{R_{1D}+R_L}{R_{2D}}\right)u$$

$\frac{d\,u}{d\,t}$ =	ux
$\frac{d\,ux}{d\,t}$ =	-((1/(R2D*C))+((R1D+RL)/L)))*ux-((1/(L*C))*(1+((R1D+RL)/R2D)))*u

Figure 17 Adjustment and entry of the second order ODE to a readable form processed by the EjsS as the basic idea used for entering such the mathematical expressions to the EPS interface

The EPS handles the Simulation.lua file in four objects to read declared coefficients and to initialize and run the defined functions dx1 (first derivative) and dx2 (second derivative). The objects represent the numerical integration functions (integrators) used to implement the simulation process by manual coding in the psc script; these integrator objects are:

1. ScriptMEODE1Integrator,
2. ScriptMEODE2Integrator,
3. ScriptRK4ODE1Integrator,
4. ScriptRK4ODE2Integrator.

The listed objects are divided into two groups based on the applied numerical methods called Modified Euler [42] and Fourth Order Runge-Kutta [43]. They are used to solve the first and second order ODEs. The mentioned order is indicated by the number 1 or 2 contained in the object's name.

6.4.1 Modified Euler integrator

It is the ScriptMEODExIntegrator component that represents two integrators using the Modified Euler method. This is not so precise method (less suitable for the simulation) but it was implemented because of the didactical purposes.

6.4.2 Runge-Kutta integrator

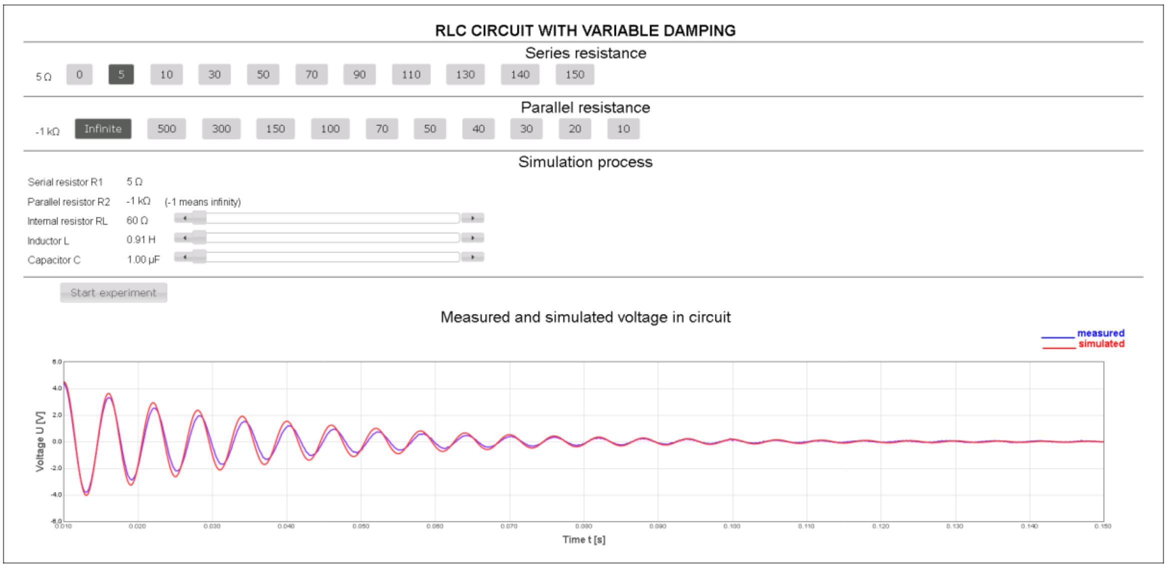
It is the ScriptRK4ODExIntegrator component that represents two integrators using the Fourth Order Runge-Kutta method (also called Runge-Kutta 4). This method produces a better solution in fewer steps. The numerical method is used deliberately because it is suitable for the simulation process.

6.4.3 Integrator parameters component

The EPS also handles the Simulation.lua file for the purpose of modifying declared coefficients in its header section by using the ScriptIntegratorParameter object that is called from the psc script to set a new value.

6.4.4 Data results visualization

The results coming from the simulation are visualized on the client's web page by specific widgets like charts and tables. There are also available widgets to set and change coefficients involved in the simulation process by using buttons and sliders. The example presenting the visualization of the embedded simulation on the client's side is shown in Figure 18.



RLC CIRCUIT WITH VARIABLE DAMPING

Series resistance

Parallel resistance

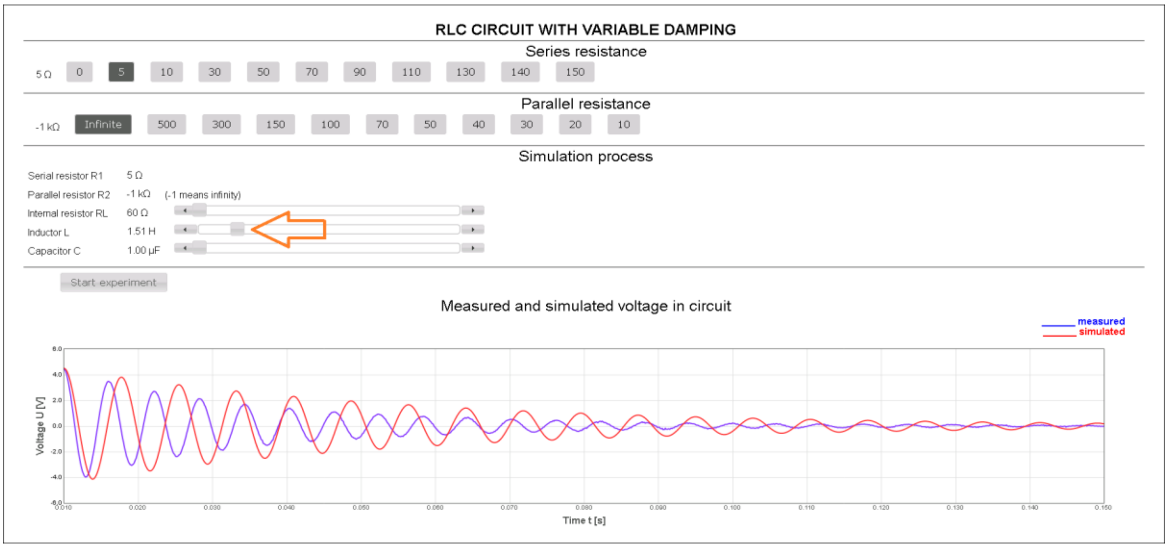
Simulation process

Serial resistor R1 5 Ω
Parallel resistor R2 -1 kΩ (-1 means infinity)
Internal resistor RL 60 Ω
Inductor L 0.91 H
Capacitor C 1.00 μF

Measured and simulated voltage in circuit

Figure 18 Example of the visualized embedded simulation provided by the EPS module that is integrated into the MS unit; it presents the RL "The electric and electromagnetic phenomena in the RLC circuit with the variable damping" where two colored curves are displayed; 1) blue: measured voltage, and 2) red: simulated voltage; both the approaches have the same values of the input parameters R_1, R_2, R_L, L and C; curves are similar

The next example represents the introduced RL, but it slightly differs in the setup of the simulated process as displayed in Figure 19.



RLC CIRCUIT WITH VARIABLE DAMPING

Series resistance

Parallel resistance

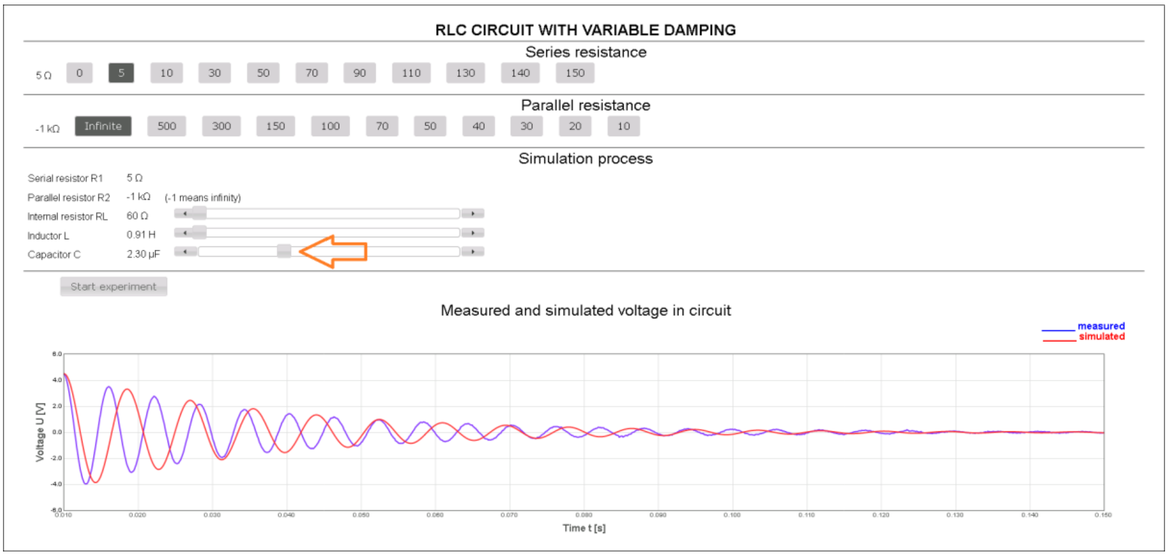
Simulation process

Serial resistor R1 5 Ω
Parallel resistor R2 -1 kΩ (-1 means infinity)
Internal resistor RL 60 Ω
Inductor L 1.51 H
Capacitor C 1.00 μF

Measured and simulated voltage in circuit

Figure 19 Example of the visualized embedded simulation; the L coefficient differs as the orange arrow points out; the curves have unmatched behavior

The last example concerning the mentioned RL demonstrates another coefficient adjustment of the simulated process as shown in Figure 20.



Serial resistor R1 5 Ω
Parallel resistor R2 -1 kΩ (-1 means infinity)
Internal resistor RL 60 Ω
Inductor L 0.91 H
Capacitor C 2.30 μF

Figure 20 Example of the visualized embedded simulation; the C coefficient differs as the orange arrow points out; the curves have unmatched behavior

The simulation functions can also be deployed in the slow RLs inside the sections characteristic for this kind of the experiment concept. Furthermore, the simulated data are coupled with the measured data and archived to xml files, and finally dispatched to the data warehouse for detailed analyses.

Summary of the subchapter "6.4 Embedded simulation module of real-world phenomena": This subchapter deals with the goal No. 5. The multi-parameter simulation synchronized with the real measured phenomenon is a tool of utmost importance, not to be found in published remote laboratories. For the purpose, the solver, including ordinary differential equations of basic mathematical function has to be built. Clients may compare the measured results with theoretical equations and adjust their parameters to learn their impact on final results, and to find their correct set. The simulation may be used as a motivation tool independently before real measurement as an introductory step for the acquaintance with measured phenomenon. This improvement is a major contribution to the intelligent version of the Measureserver unit resting in the fact of the necessity of sharing priorities of two processes. The novelty and the scientific approach rest in the use of the new embedding architecture of two concurrently running programs in one Measureserver unit. The synchronization is achieved by a specific function to trigger the computation.

7. CONTRIBUTION TO SCIENCE AND PRACTICE

The doctoral thesis tries to improve the ISES remote laboratory usability in education and dissemination of scientific knowledge, all that contributes to reliability, robustness, scalability, measured and control data storage and its processing, diagnostics and tools. For this purpose, we formulate the following contributions for the remote laboratories community and researchers.

The first contribution for the community is the implementation of new software modules, built in the Measureserver unit, deployed as the data archiving, clients' activities logging, and two level diagnostics to monitor and recover physical hardware components. These modules provide required functionalities for the REMLABNET platform.

The second contribution to the researchers in the branch is the possibility to have a feedback for remote laboratory operating and eventual improvement. This functionality is provided by the detailed analysis of the clients' activities, based on the data, logged during their experimentations.

The third contribution is to the branch of advanced tools. The embedded real-world phenomena simulation, integrated and synchronized with running remote laboratories, in the Measureserver unit. Simulation, as a simplified version of real experiment maybe used in various combinations with a real experiment, as the standalone, asynchronous and synchronous application, depending on the level of clients. In the simplification, the clients get insight into the basis of the observed phenomenon, in the most progressive case; the client may use the parameters of physics laws to fit the measurements.

The fourth contribution, aiming at the future, are modules important even more when the REMLABNET platform should host a large number of the remote laboratories focused on natural sciences and engineering fields.

CONCLUSION

The presented work is focused on the design and development of the software components related to the Measureserver unit. In the frame of work on the doctoral thesis a new Measureserver unit called Intelligent Measureserver was developed to improve stability, reliability and maintenance of the ISES remote laboratories. The implementation also helped the ISES remote laboratories to turn into high level to compete with other similar laboratories worldwide.

The following conclusions concerning the new software components in the Intelligent Measureserver unit are formulated:

1. Fulfillment of the doctoral thesis goals as documented in the individual chapters' summaries of the chapters,
2. Creation of the data archiving management module, responsible for gathering, filtering and saving measured data and metadata to an xml file, together with the option to log clients' activities to a log file,
3. Creation of the Measureserver first level diagnostics, providing the administrators warning and notifying information about the remote laboratories functioning with the recovery features,
4. Creation of the Measureserver second level diagnostics, providing the administrators to detect, monitor and maintain the ISES meters, sensors, probes or devices connected to the remote laboratory with the regards to their failures during the measurement and adjustment process,
5. Creation of the embedded real-world phenomena simulation, operating concurrently with the real remote laboratory with using the solvers to generate approximate solution of the corresponding differential equations, including continuous results visualized on the client's web page.

BIBLIOGRAPHY

[1] MAITI, Ananda, Andrew D. MAXWELL and Alexander A. KIST. Features, Trends and Characteristics of Remote Access Laboratory Management Systems. In: International Journal of Online Engineering (iJOE) [online]. 2014, 10(2), p. 30-37 [cit. 2017-01-07]. DOI: 10.3991/ijoe.v10i2.3221. ISSN 1861-2121. Available from: http://online-journals.org/index.php/i-joe/article/view/3221

[2] ŽÁKOVÁ, Katarína. Two ways of inverted pendulum remote control. In: The 6th WSEAS International Conference on Education and Educational Technology, Venice, Italy, November 21-23, 2007, El. Comp. Eng., p. 139-144 [cit. 2017-01-07]. Available from: http://www.wseas.us/e-library/conferences/2007venice/papers/570-625.pdf

[3] BISTÁK, Pavol. Matlab and Java Based Virtual and Remote laboratories for Control Engineering. In: The 17th Mediterranean Conference on Control and Automation, Thessaloniki, Jun 24-26, 2009, Vols. 1-3, p.1439-1444 [cit. 2017-01-07]. Available from: http://ieeexplore.ieee.org/document/5164749

[4] VÁLKOVÁ, Lenka and František SCHAUER. Remote interactive real experiment in electrochemistry as exemplified on the experiment Electrochemical cell. In: The 6th Int. Conference on Emerging e-learning Technologies and Applications, Slovakia, September 11-13, 2008 [cit. 2017-01-07]. Available from: http://www.iceta.sk/archive/2008

[5] VÁLKOVÁ, Lenka, František SCHAUER and Miroslava OŽVOLDOVÁ. Electrochemical cell characterization - is it start of remote experiments in chemistry education? In: The International Conference REV 2009, Bridgeport (USA), June 22-25, 2009, International Association of Online Engineering, 2009, ISBN 978-3-89958-480-6, p. 326-331 [cit. 2017-01-07]. Available from: http://www.utb.cz/file/14174_1_1/download

[6] GARCÍA-ZUBÍA, Javier, Pablo ORDUNA, Ignacio ANGULO, Unai HERNANDEZ, Olga DZIABENKO, Diego LOPEZ-IPINA and Luis RODRIGUEZ-GIL. Application and user perceptions of using the WebLab-Deusto-PLD in technical education. In: First Global Online Laboratory Consortium Remote Laboratories Workshop [online]. IEEE, 2011, p. 1-6 [cit. 2017-04-07]. DOI:

10.1109/GOLC.2011.6086780. ISBN 978-1-4577-1943-1. Available from: http://ieeexplore.ieee.org/document/6086780

[7] HARDISON, James L., Kimberly DELONG, Philip H. BAILEY and V. Judson HARWARD. Deploying interactive remote labs using the iLab Shared Architecture. In: 38th Annual Frontiers in Education Conference [online]. IEEE, 2008, S2A-1-S2A-6 [cit. 2017-04-07]. DOI: 10.1109/FIE.2008.4720536. ISBN 978-1-4244-1969-2. Available from: http://ieeexplore.ieee.org/document/4720536

[8] LOWE, D., S. MURRAY, E. LINDSAY and D. LIU. Evolving Remote Laboratory Architectures to Leverage Emerging Internet Technologies. In: IEEE Transactions on Learning Technologies [online]. 2009, 2(4), p. 289-294 [cit. 2017-04-07]. DOI: 10.1109/TLT.2009.33. ISSN 1939-1382. Available from: http://ieeexplore.ieee.org/document/5210092

[9] TAWFIK, M., E. SANCRISTOBAL, S. MARTIN, et al. Virtual Instrument Systems in Reality (VISIR) for Remote Wiring and Measurement of Electronic Circuits on Breadboard. In: IEEE Transactions on Learning Technologies [online]. 2013, 6(1), p. 60-72 [cit. 2017-04-07]. DOI: 10.1109/TLT.2012.20. ISSN 1939-1382. Available from: http://ieeexplore.ieee.org/document/6305453

[10] GOMES, Luís and Javier GARCÍA-ZUBÍA. Advances on remote laboratories and e-learning experiences. Bilbao, Spain: University of Deusto, 2007, 310 p. ISBN 978-84-9830-662-0. Available from: https://www.weblab.deusto.es/Advances_on_remote_labs.pdf

[11] GREEN, S. L. and N. M. ANID. Training K-12 teachers in STEM education: A multi-disciplinary approach. In: IEEE Integrated STEM Education Conference (ISEC) [online]. IEEE, 2013, p. 1-4 [cit. 2017-04-07]. DOI: 10.1109/ISECon.2013.6525206. ISBN 978-1-4673-5624-4. Available from: http://ieeexplore.ieee.org/document/6525206

[12] BUTIME, J, R. BESIGA, A. BWONYO, V. NAKANWAGI, T. TOGBOA and A. KATUMBA. Design of online Digital Electronics laboratories based on the NI ELVIS II platform. In: 9th International Conference on Remote Engineering and Virtual Instrumentation [online]. IEEE, 2012, p. 1-3 [cit. 2017-04-07]. DOI: 10.1109/REV.2012.6293098. ISBN 978-1467 3-2542-4. Available from: http://ieeexplore.ieee.org/document/6293098

[13] GARCÍA-ZUBÍA, Javier and Gustavo R. ALVES. Using Remote Labs in Education. Bilbao, Spain: University of Deusto, 2011, 465 p. ISBN 978-84-9830-398-8. Available from: http://www.deusto-publicaciones.es/deusto/pdfs/otraspub/otraspub01.pdf

[14] ORDUNA, Pablo. Transitive and scalable federation model for remote laboratories. Bilbao, Spain, 2013, 242 p. Doctoral thesis. University of Deusto. Supervisor Javier García-Zubía. Available from: http://www.weblab.deusto.es/pub/dissertation_pablo.pdf

[15] COSTA, David, Gustavo ALVES, Paulo FERREIRA and Juarez SILVA. Remote Labs Accessible through 3D environments A Case Study with Open Wonderland. In: 8th International Conference on Remote Engineering and Virtual Instrumentation [online], Romania: IAOE, 2011, p. 191-197 [cit. 2017-04-07]. Available from: www.tinyurl.com/zbyc898

[16] GILLET, Denis, Anh VU NGUYEN NGOC and Yassin REKIK. Collaborative Web-Based Experimentation in Flexible Engineering Education. In: IEEE Transactions on Education [online]. Vol. 48, No. 4. University of Leicester, Leicester, United Kingdom: IEEE, 2005, p. 696-704 [cit. 2017-04-07]. ISSN 0018-9359. Available from: http://www.cs.le.ac.uk/people/avnn1/papers/GilletNR-IEEETransEdu05.pdf

[17] FRANTIŠEK, Lustig. : Intelligent School Experimental System - ISES. E-Laboratory Project [online]. Prague, Czech Republic: Faculty of Mathematics and Physics, Charles University in Prague, 2017 [cit. 2017-04-07]. Available from: http://www.ises.info/index.php/en/systemises

[18] SCHAUER František, Michal KRBEČEK, Pavel BEŇO, Michal GERŽA, Lukáš PÁLKA and Petra ŠPILÁKOVÁ. REMLABNET - Open Remote Laboratory Management System for e-Experiments. REV, Porto, Portugal, 2014. ISBN 978-1-4799-2024-2. Available from: http://ieeexplore.ieee.org/document/6784273

[19] SCHAUER František, Michal KRBEČEK, Pavel BEŇO, Michal GERŽA, Lukáš PÁLKA and Petra ŠPILÁKOVÁ. REMLABNET II - Open Remote Laboratory Management System for University and Secondary Schools Research Based Teaching. In: 12th International Conference on Remote Engineering and Virtual Instrumentation [online].

Bangkok, Thailand: IEEE, 2015, p. 109-112 [cit. 2016-09-26]. DOI: 10.1109/REV.2015.7087273. ISBN: 978-1-4799-7839-7. Available from: http://ieeexplore.ieee.org/document/7087273

[20] SCHAUER, František, Michal KRBEČEK, Pavel BEŇO, Michal GERŽA, Lukáš PÁLKA, Petra ŠPILÁKOVÁ and Lukáš TKÁČ. REMLABNET III - Federated Remote Laboratory Management System for University and Secondary Schools. In: 13th International Conference on Remote Engineering and Virtual Instrumentation [online]. Madrid, Spain: IEEE, 2016, p. 232-235 [cit. 2016-09-26]. DOI: 10.1109/REV.2016.7444471. ISBN 978-1-4673-8245-8. Available from: http://ieeexplore.ieee.org/document/7444471

[21] KRBEČEK, Michal, František SCHAUER and Karel VLČEK. Communication Requirements of Laboratory Management System. In: Latest Trends on Systems - Volume II: Proceedings of the 18th International Conference on Systems [online]. Vol. 2. Santorini Island, Greece, 2014, p. 686-691 [cit. 2017-04-07]. ISBN 978-1-61804-244-6. Available from: http://www.europment.org/library/2014/santorini/bypaper/SYSTEMS/SYSTEMS2-56.pdf

[22] HAMID, R. and S. MOHAMMED. Remote access laboratory system for material technology laboratory work. In: Proceedings of the 7th WSEAS International Conference on Engineering Education [online]. Corfu Island, Greece: WSEAS, 2010, p. 311-316 [cit. 2017-04-07]. ISBN 978-960-474-202-8. Available from: http://dl.acm.org/citation.cfm?id=1864241

[23] KOUKIANAKIS, L. G. and J. G. GLENTZES. A virtual lab and e-learning system for renewable energy sources. In: WSEAS Transactions on Computers [online]. Vol. 5, No. 2. Greece: WSEAS, 2006, p. 337-341 [cit. 2017-04-07]. ISSN 1109-2750. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.579.9961&rep=rep1&type=pdf

[24] KELLER, Robert. Finite-State Machines. Finite Automata and Formal Languages [online]. Claremont, California, USA: Harvey Mudd College, 2009, p. 471-545 [cit. 2017-04-07]. Available from: http://www.cse.chalmers.se/~coquand/AUTOMATA/book.pdf

[25] BRADEN, R. Requirements for Internet Hosts - Communication Layers. Internet Standard [online]. Marina del Rey, California, USA: Internet Engineering Task Force, 1989 [cit. 2017-04-07]. Available from: https://tools.ietf.org/html/rfc1122

[26] BRADEN, R. Requirements for Internet Hosts - Application and Support. Internet Standard [online]. Marina del Rey, California, USA: Internet Engineering Task Force, 1989 [cit. 2017-04-07]. Available from: https://tools.ietf.org/html/rfc1123

[27] KRBEČEK, Michal, František SCHAUER and František LUSTIG. Easy Remote ISES - Development Environment Remote Experiments. In: Innovations 2013: World Innovations in Engineering Education and Research [online]. USA: Potomac, 2013, p. 81-100 [cit. 2017-04-07]. ISBN 978-0-9818868-4-8. ISSN 1553-9911. Available from: http://www.ises.info/old-site/clanky_pdf/Easy_Remote_ISES_2013.pdf

[28] GERŽA, Michal, František SCHAUER and Karel VLČEK. Communication Principles Between Client and Physical Hardware of ISES Remote Laboratory. AMCSE, Varna, Bulgaria, 2014. ISBN 978-1-61804-246-0. Available from: http://www.inase.org/library/2014/varna/bypaper/AMCSE/AMCSE-03.pdf

[29] GERŽA, Michal, František SCHAUER and Karel VLČEK. Advanced Communication Diagnostics in ISES Remote Experiment. International Journal of Communications [online]. North Atlantic University Union, 2015, p. 43-52 [cit. 2015-10-12]. ISSN 1998-4480. Available from: http://www.naun.org/main/NAUN/communications/2015/a142001-337.pdf

[30] GERŽA, Michal, František SCHAUER and Ivan ZELINKA. Artificial Intelligence in ISES Measureserver for Remote Experiment Control. NOSTRADAMUS, Ostrava, Česká Republika, 2014. ISBN 978-3-319-07401-6. ISSN 2194-5357. Available from: https://link.springer.com/chapter/10.1007/978-3-319-07401-6_41

[31] R. WRIGHT, David. Finite State Machines. CSC 216 Course Website [online]. USA: NC State University, 2005 [cit. 2017-04-07]. Available from: http://www4.ncsu.edu/~drwrigh3/docs/courses/csc216/fsm-notes.pdf

[32] DICK, Grune and Ceriel J.H. JACOBS. Parsing and its applications, a practical guide. Chichester, England: Ellis Horwood, 1990. ISBN 0-136514 31-6. Available from: https://dickgrune.com/Books/PTAPG_1st_Edition

[33] TRIBBLE, David. LR(k) Parsing Theory. Practical LR(k) Parser Construction [online]. Plano, Texas, USA, 2004 [cit. 2017-04-07]. Available from: http://david.tribble.com/text/lrk_parsing.html

[34] BUCK, Jamis. Writing a Simple Recursive Descent Parser. The Buckblog [online]. Smithfield, Utah, USA, 2015 [cit. 2017-04-07]. Available from: http://weblog.jamisbuck.org/2015/7/30/writing-a-simple-recursive-descent-parser.html

[35] GHOST, Gautam, Gauray ROY and Vikash SINGH. C++ Introduction. C++ Tutorials [online]. w3schools.in [cit. 2017-04-07]. Available from: http://www.w3schools.in/cplusplus-tutorial/intro

[36] KOVÁŘ, Martin. Diagnostics and Elucidation of the Environment Web ISES Control Kit for Remote Experiments Control. Zlín, Czech Republic, 2012, 78 s. Bachelor Thesis. Tomas Bata University in Zlín. Available from: http://digilib.k.utb.cz/handle/10563/22906

[37] RAY, Erik. Creating Self-Describing Data - Learning XML [online]. Sebastopol, USA: O'Reilly & Associates, 2001, 368 p. [cit. 2017-04-07]. ISBN 0-59600-046-4. Available from: https://doc.lagout.org/programmation/Java/OReilly%20Learning%20XML.pdf

[38] CAMPIONE, Ben. An MFC Wrapper for MSXML. Dr. Dobb's - The world of software development [online]. San Francisco, California, USA: UBM, 2001 [cit. 2017-04-07]. Available from: http://www.drdobbs.com/an-mfc-wrapper-for-msxml/184416288

[39] NAGY, Gabriel. Ordinary Differential Equations [online]. East Lansing, MI, USA: Mathematics Department, Michigan State University, 2017 [cit. 2017-07-16]. Available from: http://users.math.msu.edu/users/gnagy/teaching/ode.pdf

[40] IERUSALIMSCHY, Roberto, Waldemar CELES and Luiz FIGUEIREDO. Lua Documentation: Reference manual. The Programming Language

[online]. Rio de Janeiro, Brazil: LabLua, 2017 [cit. 2017-06-25]. Available from: https://www.lu[48]a.org/docs.html

[41] ESQUEMBRE, Francisco. EjsWiki Documentation. About Easy Java/Javascript Simulations [online]. Murcia, Spain: University of Murcia, 2017 [cit. 2017-06-25]. Available from: http://www.um.es/fem/EjsWiki/Main/Documentation

[42] KRISHNAN, Mangala, Kushal SEN and Bhaskar RAMAMURTHI. Numerical Solution of Ordinary Differential Equations: Modified Euler Method. National Programme on Technology Enhanced Learning [online]. Chennai, India: NPTEL, 2017 [cit. 2017-06-25]. Available from: http://nptel.ac.in/courses/111107063/3

[43] KRISHNAN, Mangala, Kushal SEN and Bhaskar RAMAMURTHI. Numerical Solution of Ordinary Differential Equations: Fourth Order Runge Kutta Methods. National Programme on Technology Enhanced Learning [online]. Chennai, India: NPTEL, 2017 [cit. 2017-06-25]. Available from: http://nptel.ac.in/courses/111107063/5

LIST OF FIGURES

Figure 1 ISES physical hardware and software components; the PCI ADDA converter, the set of meters, sensors and device 9

Figure 2 Example of the ISES RL “Sound laboratory” including the sound generator, voltage source, relay board, control board, computer, and apparatus with speaker and two microphones 11

Figure 3 Scheme of the internal and external modules constituting the MS unit that mediates the connection between the rig and remote clients 13

Figure 4 Scheme of the MS concept with the new modules located in the red frame; 1) DAM is data archiving management, 2) MUD is Measureserver unit diagnostics, 3) PMD is physical modules diagnostics, and 4) EPS is embedded real-world phenomena simulation 19

Figure 5 Example of the XML file that is generated by the DAM’s functions; the file covers the measured data, metadata and control values affecting the measurement. The viewer presents the three panels, the Attributes panel showing attributes of the “experiment” branch, the XML source panel listing the data structure and the branches tree panel 20

Figure 6 Example of the XML structure that presents the 1506161040530410.xml file to point out the subsection elements. The file comes from the RL “Transients in RLC circuits”. The content is highlighted by the colored lines to distinguish particular subsections; the identical colored line pairs indicate subsections at the same level. The yellow elements, “control” and “observation”, expose the control value and two measured buffers 21

Figure 7 Example setup of the user interface that presents the XML parameters used for the direct connection to the data warehouse to dispatch generated xml files 22

Figure 8 Example record of the clients’ activities located in the Users.log file to archive to the data warehouse 23

Figure 9 Example setup of the user interface that presents the LOG parameters used for configuring of the clients’ activities logging component... 23

Figure 10 Example of the “update” command dispatched from the DS to MS unit to obtain back the complex information describing the experiment 26

Figure 11 Example settings of the MUD module available in the MS unit to configure the direct connection to the DS and the messaging process 26

Figure 12 Notification and faulty messages generated by the MS unit when events occur and the control commands coming from the DS to use for the management of the unit in case of problems or updates 27

Figure 13 Exemplary piece of the physical modules list stored in the Modules.txt file used for the detection 29

Figure 14 Example of the MUD records; correctly running (gray), two failed (red); A) setup changed caused by the range of amperemeter, and B) one amperemeter was disconnected 30

Figure 15 Example of the detailed list presenting the current status of connected physical modules to the ports on the ISES PCI control board..........31

Figure 16 Example settings of the PMD user interface to present the involved referential physical modules and detected information with current status........31

Figure 17 Adjustment and entry of the second order ODE to a readable form processed by the EjsS as the basic idea used for entering such the mathematical expressions to the EPS interface...32

Figure 18 Example of the visualized embedded simulation provided by the EPS module that is integrated into the MS unit; it presents the RL “The electric and electromagnetic phenomena in the RLC circuit with the variable damping” where two colored curves are displayed; 1) blue: measured voltage, and 2) red: simulated voltage; both the approaches have the same values of the input parameters R_1, R_2, R_L, L and C; curves are similar.............34

Figure 19 Example of the visualized embedded simulation; the L coefficient differs as the orange arrow points out; the curves have unmatched behavior 34

Figure 20 Example of the visualized embedded simulation; the C coefficient differs as the orange arrow points out; the curves have unmatched behavior 35

LIST OF SYMBOLS, ACRONYMS AND ABBREVIATIONS USED

ADDA – Analog to Digital/Digital to Analog

CFG – Configuration

DAM – Data Archiving Management

DS – Diagnostic Server

EjsS – Easy Java/JavaScript Simulations

EPS – Embedded real-world Phenomena Simulation

ER-L – Easy Remote Laboratory

FSM – Finite-State Machine

GUI – Graphical User Interface

HTTP – Hypertext Transfer Protocol

HW – Hardware

IP – Internet Protocol

ISES – Intelligent School Experimental System

LOG – Data Log

MS – Measureserver

MSXML – Microsoft XML Core Services

MUD – Measureserver Unit Diagnostics

ODE – Ordinary Differential Equation

PCI – Peripheral Component Interconnect

PMD – Physical Modules Diagnostics

RL – Remote Laboratory

RLC – Resistor, Inductor and Capacitor

RLMS – Remote Laboratory Management System

TCP – Transmission Control Protocol

TCP/IP – Transmission Control Protocol/Internet Protocol

XML – eXtensible Markup Language

LIST OF PUBLICATIONS BY THE AUTHOR

1. GERŽA, Michal and Pavel POKORNÝ. A Visualisation of the Results of a Thermoforming Process Simulation in the Plastics Industry. In: 21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: WSCG2013 - Poster Proceedings. Plzeň: Západočeská univerzita v Plzni, Katedra informatiky a výpočetní techniky, 2013. ISBN 978-80-86943-76-3.

2. GERŽA, Michal, František SCHAUER and Roman JAŠEK. Security of ISES Measureserver Module for Remote Experiments against Malign Attacks. International Journal of Online Engineering, Wien, Austria, 2014. ISSN 1868-1646.

3. GERŽA, Michal, František SCHAUER and Karel VLČEK. Communication Principles between Client and Physical Hardware of ISES Remote Laboratory. AMCSE, Varna, Bulgaria, 2014. ISBN 978-1-61804-246-0.

4. GERŽA, Michal, František SCHAUER and Ivan ZELINKA. Artificial Intelligence in ISES Measureserver for Remote Experiment Control. NOSTRADAMUS, Ostrava, Česká Republika, 2014. ISSN 2194-5357.

5. SCHAUER František, Michal KRBEČEK, Pavel BEŇO, Michal GERŽA, Lukáš PÁLKA and Petra ŠPILÁKOVÁ. REMLABNET - Open Remote Laboratory Management System for e-Experiments. REV, Porto, Portugal, 2014. ISBN 978-1-4799-2024-2.

6. GERŽA, Michal and František SCHAUER. Advanced Modules Diagnostics in ISES Remote Laboratories. In: The 10th International Conference on Computer Science & Education. Fitzwilliam College, Cambridge University, UK, 2015. ISBN 978-1-4799-6599-1.

7. SCHAUER František, Michal KRBEČEK, Pavel BEŇO, Michal GERŽA, Lukáš PÁLKA and Petra ŠPILÁKOVÁ. REMLABNET II - Open Remote Laboratory Management System for University and Secondary Schools Research Based Teaching. In: 12th International Conference on Remote Engineering and Virtual Instrumentation [online]. Bangkok, Thailand: IEEE, 2015, p. 109-112 [cit. 2016-09-26]. DOI: 10.1109/REV.2015.7087273. ISBN: 978-1-4799-7839-7.

8. GERŽA, Michal, František SCHAUER and Karel VLČEK. Advanced Communication Diagnostics in ISES Remote Experiment. International Journal of Communications [online]. North Atlantic University Union, 2015, p. 43-52 [cit. 2015-10-12]. ISSN 1998-4480.

9. GERŽA, Michal and František SCHAUER. Intelligent Processing of Experimental Data in ISES Remote Laboratory. International Journal of Online Engineering [online]. Wien, Austria, 2016, (Vol 12, No 03), p. 58-63 [cit. 2016-09-26]. DOI: 10.3991/ijoe.v12i03.5538. ISSN 1861-2121.

10. GERŽA, Michal, František SCHAUER and Petr DOSTÁL. Embedded Simulations in Real Remote Experiments for ISES e-Laboratory. In: 9th EUROSIM Congress on Modelling and Simulation [online]. Oulu, Finland: IEEE, 2016, p. 653-658 [cit. 2016-09-26]. DOI: 10.1109/EUROSIM.2016.66. ISBN 978-1-5090-4119-0.

11. SCHAUER, František, Michal KRBEČEK, Pavel BEŇO, Michal GERŽA, Lukáš PÁLKA, Petra ŠPILÁKOVÁ and Lukáš TKÁČ. REMLABNET III - Federated Remote Laboratory Management System for University and Secondary Schools. In: 13th International Conference on Remote Engineering and Virtual Instrumentation [online]. Madrid, Spain: IEEE, 2016, p. 232-235 [cit. 2016-09-26]. DOI: 10.1109/REV.2016.7444471. ISBN 978-1-4673-8245-8.

12. SCHAUER, František, Michal GERŽA, Michal KRBEČEK, Pavel BEŇO, Lukáš PÁLKA, Petra ŠPILÁKOVÁ, Tomáš KOMENDA, Miroslava OŽVOLDOVÁ, Žaneta GERHATOVÁ and Lukáš TKÁČ. REMLABNET IV - LTI Federated Remote Laboratory Management System with Embedded Simulations. In: 14th International Conference on Remote Engineering and Virtual Instrumentation [online]. New York, USA: IEEE, 2017, p. 340-349 [cit. 2017-06-19].

13. SCHAUER, František, Michal GERŽA, Michal KRBEČEK and Miroslava OŽVOLDOVÁ. Remote Experiment Wave Laboratory for Wave Phenomena Teaching. In: 14th International Conference on Remote Engineering and Virtual Instrumentation [online]. New York, USA: IEEE, 2017, p. 350-356 [cit. 2017-06-19].

AUTHOR'S PROFESSIONAL CURRICULUM VITAE

Michal Gerža, MSc.

Address:	Jílová 4571 760 05 Zlín Czech Republic
Phone:	+420 603732198
Email:	michal.gerza@email.cz

Personal data:

Date of birth:	22. November 1974
Marital status:	Married
Mother tongue:	Czech language

Education:

2012 – 2017	Thomas Bata University in Zlín, Faculty of Informatics, Postgraduate study (PhD), Field of study: Engineering Informatics
2007 – 2012	Thomas Bata University in Zlín, Faculty of Informatics, Bachelor and graduate study, Field of study: Engineering Informatics
1989 – 1993	Secondary school of Electrotechnic in Vsetín Field of study: Mechanical engineering

Professional experiences:

2010 – 2017	Accuform, Zlín, Programmer
2009 – 2010	Edhouse, Ltd., Zlín, Programmer
1993 – 2009	Glass Service, Inc., Vsetín, Programmer

Language:

- English
- Russian

Skills:

- Flexibility
- Independence
- Responsibility
- Communicableness
- Interest in communication with people
- Leading with human resources
- Project management
- Computer skills

Projects:

- SCOPES project (Swiss National Science Foundation and Swiss Agency for Developments and Cooperation) aimed at the remote laboratories community

Hobby:

- Fitness
- Tourism
- Psychology
- Computer technology

Other knowledge:

- Programing languages: C, C++, C#, Java, JavaScript, Visual Basic
- Web applications: PHP, CSS, XML, XSL, XAML, HTML
- Database systems: MSSQL, MySQL, PostgreSQL
- Computer systems: Windows, Linux

Inteligentní Measureserver pro řízení vzdálených experimentů s vestavěnou simulací a pokročilou diagnostikou

Intelligent Measureserver for Controlling Remote Real Experiments with Embedded Simulations and Advanced Diagnostics

Doctoral Thesis Summary

Published by: Tomas Bata University in Zlín,

nám. T. G. Masaryka 5555, 760 01 Zlín.

Edition: published electronically

Typesetting by: Michal Gerža

This publication has not undergone any proofreading or editorial review.

Edition 1st

Publication year: 2017

ISBN 978-80-7454-707-2