

Remote Laboratory Description Language Based On XML

[doi:10.3991/ijoe.v5s1.1007](https://doi.org/10.3991/ijoe.v5s1.1007)

Ian Andrew Grout¹ and Alexandre César Rodrigues da Silva²

¹University of Limerick, Limerick, Ireland

²State University of Sao Paulo (UNESP), Ilha Solteira, Sao Paulo State, Brazil

Abstract—In this paper, a language used to describe the structure and capabilities (attributes) of remote, or online, laboratories is presented. With reference to the Remote Access Laboratory (RAL) at the University of Limerick, the structure of this laboratory is presented in order to identify the use of the Remote Laboratory Description Language (RLDL). This is based on XML providing for a text based, standard format on which to build and extend RLDL. The structure of the language is presented and an example is given with reference to the specific case study remote laboratory. However, the language can be readily extended to describe current laboratory attributes in more detail and to extend the language in order to identify and present new laboratory attributes. Such a description can be used to exchange information between remote laboratory developers and maintainers.

Index Terms—Remote Laboratory, Standard, Descriptor

I. INTRODUCTION

As the development and deployment of remote (or online) laboratories increases, with the opportunities for laboratory providers to exchange information and establish national or global laboratory grids, then the need to exchange information in a standardized format increases. In this paper, then a standard description language based on XML (Extensible Markup Language) [1] is presented and discussed. This language can be used to describe the structure and capabilities of a remote laboratory. The features of an example remote laboratory based within the University of Limerick [2] is presented, highlighting the key features of the laboratory considered for description and information exchange. Specific features of the existing laboratory are presented, however additional features can be added using XML based syntax.

Remote (or online) laboratories are engineering and scientific laboratories that are accessible by a user from a remote location. Such laboratories can be utilized with great effect in education and research activities. The use of remote laboratories in the education domain has gained substantial popularity and use over the last few years, providing students access to specific laboratory facilities that they might not normally be able to access due to a number of constraints. In many cases, the remote laboratories have been utilized in electronic engineering (both software and hardware) design and test laboratories, and also in mechanical engineering laboratories (such as

telerobotics). Given the ability to develop and use a custom built laboratory, or to have access to an existing laboratory facility, then the educational community can readily integrate remote laboratories into the education curriculum. This has been demonstrated by universities around the world. Fig. 1 shows an example remote laboratory arrangement developed and used at the University of Limerick (Ireland).



Figure 1. Example remote laboratory set-up

To provide an example case study design, the Remote Access Laboratory (RAL) developed and used within the University of Limerick (Ireland) has been utilized since 2002 as a content management system (CMS) and also as a remote laboratory for students learning microelectronic circuit test engineering concepts [3]. The electronic hardware experiments for this particular laboratory arrangement are shown in Fig. 1 (showing the laboratory in its current form and the electronic hardware experiments). The top photograph shows the main server and electronic hardware experiments connected to the main server. The middle photograph shows additional electronic hardware experiments connected to the main server. The bottom photograph shows one of the two auxiliary servers and an electronic hardware experiment connected to this particular server.

Given that access to a range of remote laboratories which would be developed at different institutions is possible, and that the laboratories would have been developed with different requirements and priorities in mind, then the detailed structure of the laboratories would naturally differ. However, the underlying laboratory architecture for each laboratory identifying the key modules that are combined to form the overall laboratory would show a common set of attributes. If these attributes can be described in a standard form using a standard syntax, then this would lead towards a standardized approach to remote laboratory development and facilitate information exchange.

This paper provides a possible structured language based on XML that could be used to describe the common attributes for all remote laboratories. The paper is an extension of the paper presented at the Remote Engineering and Virtual Instrumentation Conference (REV 2009, Bridgeport (USA), June 2009). The paper is structured as follows. Section II will provide an introduction to the use of XML. Section III will present the remote laboratory description language (RLDL) which is based on the XML syntax. Section IV will present a case study design of the remote laboratory description language as applied to the Remote Access Laboratory (RAL) at the University of Limerick. Section V will identify further work in language development and section VI provides the source for further information. Section VII concludes the paper.

II. USE OF XML

XML (Extensible Markup Language) is a **markup language** for use in documents that contain **structured information**, providing for:

1. A **markup language** that is a mechanism to identify structures within a document: XML defines a standard way to add markup to documents which contain structured information.
2. **Structured information** that contains both content (words, pictures, etc.) and an indication of what role that the content plays.

XML was developed from the earlier language SGML (Standard Generalized Markup Language) [4], as was HTML (Hypertext Markup Language) [5]. XML is very similar in appearance to HTML. For example, it has tags which identify elements. However, HTML is used for displaying information (with a focus on the appearance of data), whereas XML is used for transporting (carrying) and storing information (with a focus on what the data is). Essentially XML is plain text which can be read by any software application that can handle plain text. However, any software application that is "XML-aware" can handle XML tags (the tags identify elements and also contain attributes about these elements) in a particular way – the handling of the tags dependent on the particular software application. An XML document contains both markup and data. The text between the start and end tags are defined as data and the text within the tags defined to be markup. Data is typically character data (such as letters, numbers and punctuation), but it can also be binary data. An XML document which has been created for a particular purpose would be **well formed** but can also be **valid**, where:

- **Well formed XML documents** contain text and XML tags which conform to the XML syntax,
- **Valid XML documents** must be well formed and are additionally error checked against a Document Type Definition (DTD). A DTD is a set of rules that outline which tags are allowed, what values those tags may contain and how the tags relate to each other.

Given the ability to define structured data, then this can be applied to describing the structure of a remote laboratory. Each module (experiment, PC, software application, etc.) within the laboratory can be identified and described. Fig. 2 shows an example generalized remote laboratory structure which has a single web server which the remote learner accesses. Here, the web server contains the necessary software applications and source code for the server to operate and communicate with the remote laboratory modules. In addition, the server utilizes a database for actions such as laboratory configuration data, laboratory access control, results handling and data for general laboratory administrative tasks. Connected to the web server are electronic hardware experimentation, computer software experimentation and mechanical structure experimentation, along with other additional experimentation PCs. These additional experimentation PCs can also host their own set of experiments. Based on this, then the specific structure of such a laboratory would be dependent on the particular laboratory requirements.

III. REMOTE LABORATORY DESCRIPTION LANGUAGE

Given the ability to utilize XML in describing structure, then this is used to form the basis of a language that would describe the structure and also the capabilities of a remote laboratory. Such a language can represent the

laboratory at different levels of abstraction. For example, a high level of abstraction would just describe the functional modules (these are described by elements in XML) that form the overall laboratory structure and their interconnections. A more detailed (lower) level of abstraction would describe the overall laboratory structure and their interconnections as well as provide information on how the modules operate and their intended purpose, along with additional information that could be utilized to analyze the structure of the laboratory (described by data within the element). In this case, the language has been proposed to consider the ability to incorporate laboratory design details as well as structure. It should be noted that given the wide range of possible remote laboratory designs possible, then the language described here would provide a generalized laboratory description which could be added to as and when necessary.

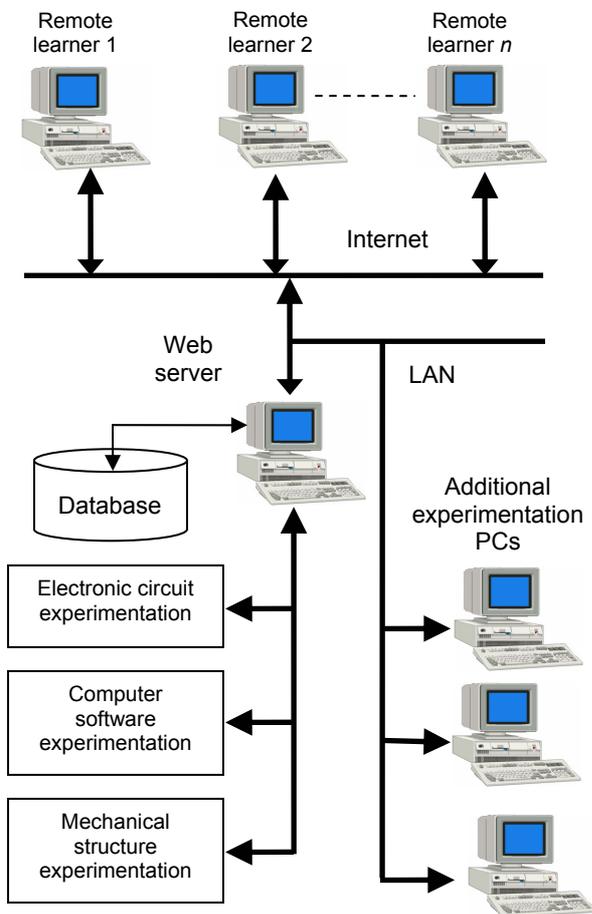


Figure 2. Generalized remote laboratory structure

As in XML, then the first step is to create a **root element** which will identify the modules that form the overall remote laboratory. Fig. 3 shows the root element as defined in the associated document type definition (DTD). The root element name is always to be set to `remote_laboratory`.

Here, there are twelve identified modules which would form a typical remote laboratory structure. Of course, it is

possible to either use a sub-set of these modules or to identify additional modules to include in the laboratory.

```
<!ELEMENT remote_laboratory
(laboratory_server,
auxillary_server*,
digital_circuit*,
analogue_circuit*,
mixed_signal_circuit*,
mechanical*,
software*,
web_cam*,
test_equipment*,
self_test_capability*,
location*,
database*)>
```

Figure 3. Remote laboratory root element

Within this structure, then the modules are defined as XML ELEMENTS and the purpose for each of these elements is identified in table 1.

TABLE I.
REMOTE_LABORATORY ELEMENT AND CHILD ELEMENTS

Element	Purpose
<code>remote_laboratory</code>	The laboratory root element name.
<code>laboratory_server</code>	The main server which is accessed by a remote user.
<code>auxillary_server</code>	Other computers (servers) which are part of the laboratory which are normally accessed through the main server.
<code>digital_circuit</code>	Digital electronic circuit laboratory experiment.
<code>analogue_circuit</code>	Analogue electronic circuit laboratory experiment.
<code>mixed_signal_circuit</code>	Mixed-signal electronic circuit laboratory experiment.
<code>mechanical</code>	Mechanical structure laboratory experiment.
<code>software</code>	Software application that can be accessed for software laboratory experiments.
<code>web_cam</code>	Web cams connected to specific experiments for the remote user to view the real experiment.
<code>test_equipment</code>	Test and measurement equipment connected to experiment such as oscilloscopes.
<code>self_test_capability</code>	Details of any self-test capability built into the remote laboratory.
<code>location</code>	The physical location of the laboratory.
<code>database</code>	The database hosted on the main server which will hold laboratory information and user data.

As an example of structure of the defined elements, the `laboratory_server` element is shown in Fig. 4. This describes the main server (the access server for remote users). Here, this element is defined to have ten child elements.

Each child element identifies a specific aspect of the main server. The child elements within the `laboratory_server` element are identified in table 2. This set of information provides a basis on which the laboratory can be identified for access and integration into a grid of remote laboratories.

```

<laboratory_server>
  <server_name>
    RAL Main Server
  </server_name>
  <server_number> 1 </server_number>
  <ip_address> 193.1.97.77 </ip_address>
  <url>
    http://www.ral.ul.ie
  </url>
  <organisation>
    University of Limerick
  </organisation>
  <web_master_name>
    Ian Grout
  </web_master_name>
  <web_master_email>
    Ian.Grout@ul.ie
  </web_master_email>
  <web_server> Apache 2.0 </web_server>
  <operating_system>
    Microsoft Windows 2000 Professional
  </operating_system>
  <additional_information>
    Remote Access Laboratory
  </additional_information>
</laboratory_server>
    
```

Figure 4. `laboratory_server` description element

TABLE II.
LABORATORY_SERVER CHILD ELEMENTS

Element	Purpose
<code>server_name</code>	The local name to identify the main server in the laboratory structure.
<code>server_number</code>	The number of the server (starting at 1) and this number is a unique number for each of the computer servers in the laboratory.
<code>ip_address</code>	The IP address of the remote laboratory.
<code>url</code>	The URL of the remote laboratory.
<code>organisation</code>	The name of the organization hosting the remote laboratory.
<code>web_master_name</code>	Name of the remote laboratory administrator.
<code>web_master_email</code>	Email address for the remote laboratory administrator.
<code>web_server</code>	The web server software application (e.g. Apache 2.0).
<code>operating_system</code>	The computer operating system (i.e. Windows/UNIX/Linux).
<code>additional_information</code>	Any additional information can be added which has not been captured in the above elements.

IV. RLDL CASE STUDY APPLICATION

The language to describe a remote laboratory can be applied to a case study laboratory design. The remote laboratory developed at the University of Limerick (the Remote Access Laboratory (RAL)) is considered in this

paper. In its current form, the laboratory consists of the main server and two auxiliary servers (one being a Windows operating system PC hosting an electronic circuit experiment for research and the other being a Linux operating system PC hosting software applications (simulation software for both research and teaching)). The main server hosts a MySQL database and five electronic circuit hardware experiments (for both research and teaching). There are four web cams attached to the main server. This structure is shown in Fig. 10. In this figure, then each of the modules attached to the servers (the main laboratory server and the auxiliary servers) is identified. In this, the main laboratory server (RAL Main Server) hosts the database (RAL_Database), connects to two auxiliary servers (RAL Windows Server 1 and RAL Linux Server 1), two software experiments (MATLAB® and MVHDL), two digital circuit experiments (Remote Test Laboratory and Programmable Logic Unit), two mixed-signal circuit experiments (PLL Tester and Optical Fibre tester) and four web cams (RTL_Cam, ULTEDB_Cam, PLL_Cam, and Conference_Cam). Here, the web cam is connected to the same server as the experiment. To describe the laboratory, then three files are required:

1. The **RLDL (XML based) laboratory description file** (RAL_RLDL.xml (.xml file extension)) that describes the particular laboratory structure and attributes (the file name would be specific to the laboratory). The structure of this file is shown in Fig. 11.
2. The **XSL stylesheet file** (RLDL.xsl) which is the XML stylesheet that allows the XML file to be viewed in an Internet browser such as Internet Explorer with formatted text (the file name is set as RLDL.xsl). The structure of this file is shown in Fig. 12.
3. The **DTD file** (document type definition file – RLDL.dtd) that sets the RLDL file validity checking with a suitable parser (the file name is set to RLDL.dtd). This file is shown in Fig. 13.

Fig. 5 shows the RAL_RLDL.xml file as viewed in Microsoft® Internet Explorer®. The style layout here is simple, identifying each of the laboratory modules and the details for each module. The color coding is used to identify each module type.

Given that the laboratory description can be represented in various forms, then a suitable software application (for example) could be developed in order to view the laboratory structure as a graphical representation of the form shown in Fig. 10. A suitable software application would then read in the RLDL file and then graphically display (and potentially allow a user to manipulate) the laboratory structure. This would then provide a means to view and to analyze/modify the laboratory structure.

The attempt here is to show the structure of the laboratory in such a way as to identify the laboratory hierarchy (if one exists) and to visualize how the laboratory resources are distributed across the laboratory as a whole.

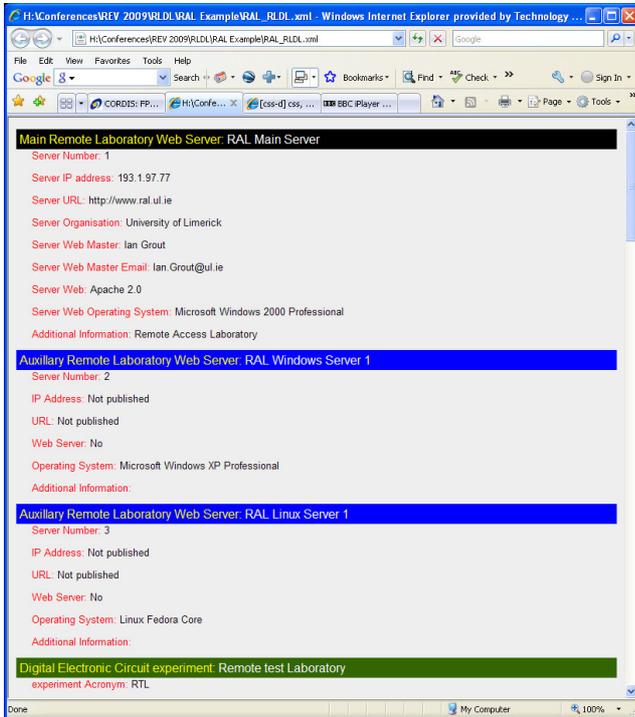


Figure 5. RLDL file viewed in Internet Explorer

The remote laboratory would consist of a number of resources that would be accessible by a user. For example, an experiment based on a digital electronic circuit could be created. Fig. 6 shows one digital electronic circuit experiment connected to the remote laboratory.

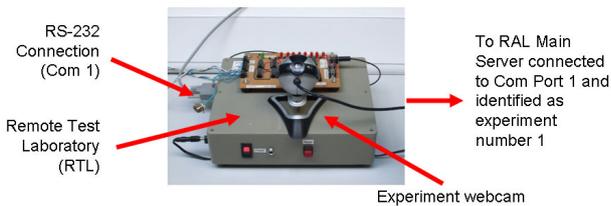


Figure 6. Digital electronic circuit experiment example

Fig. 6 shows a digital electronic circuit experiment (top circuit board – Remote Test Laboratory (RTL)) with an interface circuit (located within the box, not visible) connecting the experiment to the appropriate PC via an RS-232 connection. The description for this experiment in RLDL is shown in Fig. 7. There are seven identified attributes for this particular type of experiment (the digital electronic circuit experiment). These attributes relate to (i) the experiment name, (ii) the experiment purpose, (iii) how the experiment is connected to the server, and (iv) additional experiment information not previously captured but the laboratory description provider provides.

```
<digital_circuit>
  <experiment_name> Remote Test Laboratory
  </experiment_name>
  <experiment_acronym> RTL
  </experiment_acronym>
  <experiment_number> 1
  </experiment_number>
  <experiment_purpose> Teaching and learning
  </experiment_purpose>
  <server_connection> RAL Main Server
  </server_connection>
  <connection_type> RS-232 (Com 1)
  </connection_type>
  <additional_information>
  </additional_information>
</digital_circuit>
```

Figure 7. Remote Test Laboratory description

An additional attribute of the remote laboratory is its physical location. Whilst the general user does not necessarily need to know where the laboratory is actually located (they would primarily be interested in using the laboratory resources), it would be important for laboratory and laboratory grid developers to have access this information. The RLDL file element <location> allows the developer to include the details of the location of a KML (Keyhole Markup Language) file for the laboratory (if one exists). An example for this attribute for the Remote Access Laboratory is shown in Fig. 8 [6]. This particular element provides the location of an on-line resource providing information on the KML file (or can be the KML file directly if so required).

```
<location>
  <organisation>
    University of Limerick
  </organisation>
  <kml_file_location>
    http://www.ral.ul.ie/LabID/RAL_ULimerick_Information.php
  </kml_file_location>
  <additional_information>
  </additional_information>
</location>
```

Figure 8. <location> element

The KML file would normally be viewed in Google Earth™ as shown in Fig. 9. However, care must be taken with this location file as a laboratory might be relocated at some point in time and therefore access to current information on the laboratory location for all users would be important.

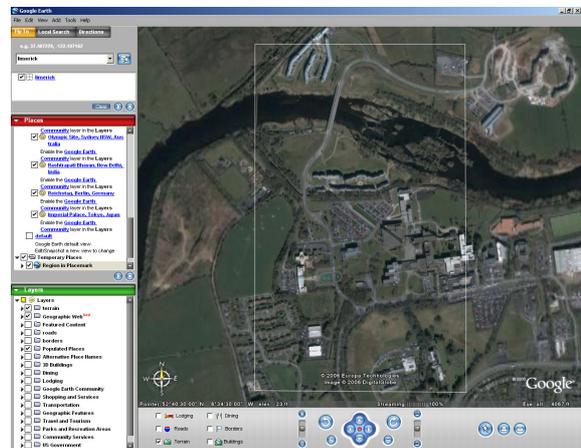


Figure 9. RAL location in Google Earth™

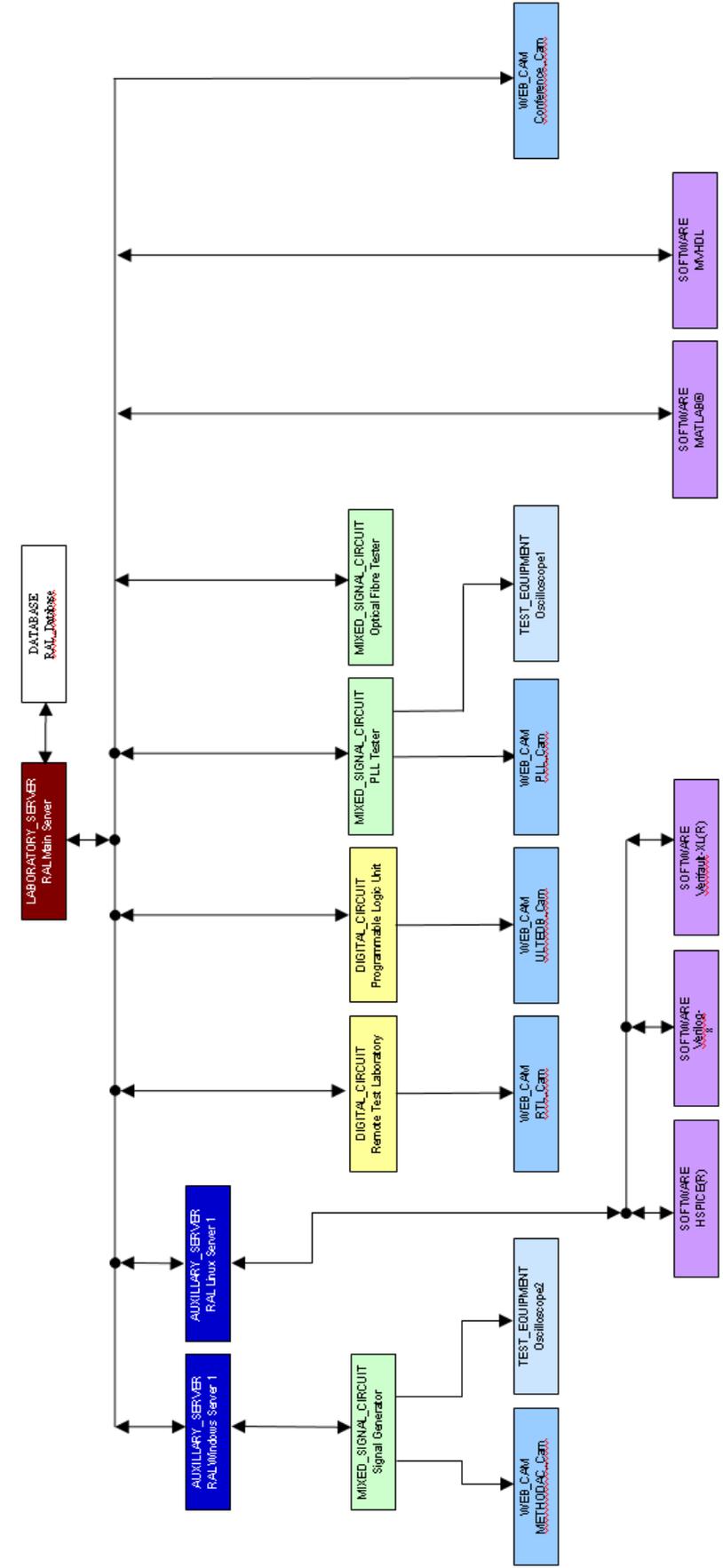


Figure 10. Remote Access Laboratory structure

REMOTE LABORATORY DESCRIPTION LANGUAGE BASED ON XML

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<?xml-stylesheet type="text/xsl" href="RLDL_Stylesheet.xsl"?>

<!-- ***** -->
<!-- Remote Laboratory Description Language (RLDL) -->
<!-- File creation date: 15th February 2009 -->
<!-- ***** -->
<!-- File created using version 1.0 of the RLDL format -->
<!-- Last update: 15th February 2009 -->
<!-- ***** -->

<!-- ***** -->
<!-- Remote Laboratory attributes -->
<!-- ***** -->

<remote_laboratory>

    <laboratory_server>
    .
    .
    </laboratory_server>

    <auxillary_server>
    .
    .
    </auxillary_server>

    <digital_circuit>
    .
    .
    </digital_circuit>

    <mixed_signal_circuit>
    .
    .
    </mixed_signal_circuit>

    <software>
    .
    .
    </software>

    <web_cam>
    .
    .
    </web_cam>

    <test_equipment>
    .
    .
    </test_equipment>

    <self_test_capability>
    .
    .
    </self_test_capability>

    <location>
    .
    .
    </location>

    <database>
    .
    .
    </database>

</remote_laboratory>

```

Figure 11. Remote Access Laboratory RLDL file code structure (.xml file structure)

```

<?xml version="1.0" encoding="ISO-8859-1" ?>

<html xsl:version="1.0"
      xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
      xmlns="http://www.w3.org/1999/xhtml">

<body style="font-family:Arial;font-size:12pt;background-color:#EEEEEE">

<xsl:for-each select="remote_laboratory/laboratory_server">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/auxillary_server">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/digital_circuit">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/mixed_signal_circuit">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/analogue_circuit">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/software">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/mechanical">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/web_cam">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/test_equipment">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/self_test_capability">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/location">
.
.
</xsl:for-each>

<xsl:for-each select="remote_laboratory/database">
.
.
</xsl:for-each>

</body>

```

Figure 12. Remote laboratory description language XSL stylesheet code structure

REMOTE LABORATORY DESCRIPTION LANGUAGE BASED ON XML

```

<?xml version="1.0"?>
<!DOCTYPE remote_laboratory [
  <!ELEMENT remote_laboratory      (laboratory_server,      auxillary_server*,      digital_circuit*,
    analogue_circuit*,             mixed_signal_circuit*,  mechanical*,
    software*,                     web_cam*,               test_equipment*,
    self_test_capability*,         location*,              database*)>
  <!ELEMENT laboratory_server      (server_name,           server_number,           ip_address,
    url,                            organisation,           web_master_email,       web_master_name,
    web_server,                     web_server,            operating_system,
    additional_information)>
  <!ELEMENT server_name            (#PCDATA)>
  <!ELEMENT server_number          (#PCDATA)>
  <!ELEMENT ip_address             (#PCDATA)>
  <!ELEMENT url                   (#PCDATA)>
  <!ELEMENT organisation           (#PCDATA)>
  <!ELEMENT web_master_name        (#PCDATA)>
  <!ELEMENT web_master_email       (#PCDATA)>
  <!ELEMENT web_server             (#PCDATA)>
  <!ELEMENT operating_system       (#PCDATA)>
  <!ELEMENT additional_information (#PCDATA)>
  <!ELEMENT auxillary_server       (server_name,           server_number,           ip_address,
    url,                            web_server,            operating_system,
    additional_information)>
  <!ELEMENT digital_circuit        (experiment_name,       experiment_acronym,      experiment_number,
    experiment_purpose,               server_connection,      connection_type,
    additional_information)>
  <!ELEMENT experiment_name        (#PCDATA)>
  <!ELEMENT experiment_acronym     (#PCDATA)>
  <!ELEMENT experiment_number      (#PCDATA)>
  <!ELEMENT experiment_purpose       (#PCDATA)>
  <!ELEMENT server_connection      (#PCDATA)>
  <!ELEMENT connection_type        (#PCDATA)>
  <!ELEMENT analogue_circuit       (experiment_name,       experiment_acronym,      experiment_number,
    experiment_purpose,               server_connection,      connection_type,
    additional_information)>
  <!ELEMENT mixed_signal_circuit   (experiment_name,       experiment_acronym,      experiment_number,
    experiment_purpose,               server_connection,      connection_type,
    additional_information)>
  <!ELEMENT mechanical            (experiment_name,       experiment_acronym,      experiment_number,
    experiment_purpose,               server_connection,      connection_type,
    additional_information)>
  <!ELEMENT software              (software_name,         software_number,         software_purpose,
    server_connection,               connection_type,        additional_information)>
  <!ELEMENT software_name          (#PCDATA)>
  <!ELEMENT software_number        (#PCDATA)>
  <!ELEMENT software_purpose         (#PCDATA)>
  <!ELEMENT web_cam               (web_cam_name,         web_cam_number,         experiment,
    server_connection,               additional_information)>
  <!ELEMENT web_cam_name           (#PCDATA)>
  <!ELEMENT web_cam_number         (#PCDATA)>
  <!ELEMENT experiment            (#PCDATA)>
  <!ELEMENT test_equipment        (test_equipment_name,  test_equipment_number,  experiment,
    server_connection,               additional_information)>
  <!ELEMENT test_equipment_name    (#PCDATA)>
  <!ELEMENT test_equipment_number  (#PCDATA)>
  <!ELEMENT experiment            (#PCDATA)>
  <!ELEMENT self_test_capability   (self_test,             digital_circuit_test,   analogue_circuit_test,
    mixed_signal_test,               mechanical_test,         software_test,
    additional_information)>
  <!ELEMENT self_test              (#PCDATA)>
  <!ELEMENT digital_circuit_test   (#PCDATA)>
  <!ELEMENT analogue_circuit_test  (#PCDATA)>
  <!ELEMENT mixed_signal_test      (#PCDATA)>
  <!ELEMENT software_test          (#PCDATA)>
  <!ELEMENT mechanical_test        (#PCDATA)>
  <!ELEMENT location              (organisation,         kml_file_location,      additional_information)>
  <!ELEMENT organisation           (#PCDATA)>
  <!ELEMENT kml_file_location      (#PCDATA)>
  <!ELEMENT database              (database_used,         database_name,           database_application,
    database_location,               additional_information)>
  <!ELEMENT database_USED          (#PCDATA)>
  <!ELEMENT database_name          (#PCDATA)>
  <!ELEMENT database_APPLICATION  (#PCDATA)>
  <!ELEMENT database_location      (#PCDATA)>
]>

```

Figure 13. Remote laboratory description language DTD file

V. FURTHER WORK IN LANGUAGE DEVELOPMENT

This work has considered the elements required to describe a specific remote laboratory case study example. Whilst these would describe the structure and attributes of many remote laboratories in general, the language can be extended in order to both enhance the existing element details and also to include new laboratory elements. For example, science experiments such as those in physics and chemistry are not explicitly described in the current version of the language but there is no reason why they cannot be so suitably described. Given the structured nature of XML and therefore RLDL, the language can be readily extended to support these enhancements.

Further work would consider the extension of the language to allow for enhancements to existing laboratory elements and for new laboratory elements to be described and therefore shared. In addition, a review of how the description files are formed can be undertaken in order to identify whether the current file structures are the most suitable.

The extended use of the language other than simply to describe an existing remote laboratory could also be identified. For example, suitably defined and created software tools could be used in the creation of new laboratory structures providing the developer information on optimal laboratory configurations and the ability to visualize laboratory structures prior to building a physical laboratory. These aspects could come from a suitably formed user community.

Any suggestions on this work can be emailed to Ian.Grout@ul.ie.

VI. FURTHER INFORMATION

Specific code extracts for developed code have been included in the paper. Further information on this work, additional code and a description of the language (in the form of a language user guide available in PDF file format) are available at the following URL:

<http://www.ral.ul.ie/RLDL/index.htm>

VII. CONCLUSIONS AND FUTURE WORK

This paper has presented and discussed the development and structure of a language to describe the structure and capabilities (attributes) of remote, or online, laboratories.

The Remote Laboratory Description Language (RLDL) is based on XML providing for a text based, standard format on which information on remote laboratories can be provided and be exchanged between remote laboratory developers and maintainers.

Reference to the Remote Access Laboratory (RAL) at the University of Limerick was made and the structure of this laboratory was presented using RLDL.

However, given the structured nature of XML and therefore RLDL, the language can be readily extended to describe current laboratory attributes in more detail and to extend the language in order to identify and present new laboratory attributes.

REFERENCES

- [1] W3C, Extensible Markup Language (XML) 1.0 (Fifth Edition), <http://www.w3.org/TR/REC-xml/>
- [2] Remote Access Laboratory at the University of Limerick, <http://www.ral.ul.ie>
- [3] J. Murphy, I. Grout, J. Walsh, and T. O'Shea, "Local and Remote Laboratory User Experimentation Access using Digital Programmable Logic", *International Journal of Online Engineering*, vol. 1, October 2005
- [4] International Organization for Standardization (ISO), ISO Standard ISO 8879:1986, Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML), <http://www.iso.org>
- [5] Musciano C. and Kennedy B., HTML The Definitive Guide, O'Reilly & Associates Inc., 1996, ISBN 1-56592-175-5
- [6] I.A. Grout, "Remote Laboratory Standard Descriptor based on KML", Proceedings of the Remote Engineering and Virtual Instrumentation Conference 2007 (REV 200), Porto, Portugal, June 2007

GLOSSARY

CDATA	Character Data
CMS	Content Management System
DTD	Document Type Definition
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IP	Internet Protocol
KML	Keyhole Markup Language
LAN	Local Area Network
PCDATA	Parsed Character Data
PHP	PHP Hypertext Preprocessor
RAL	Remote Access Laboratory
RLDL	Remote Laboratory Description Language
SGML	Standard Generalized Markup Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XML	Extensible Markup Language

AUTHORS

Ian Andrew Grout is with the University of Limerick, Department of Electronic and Computer Engineering, Limerick, Ireland (Ian.Grout@ul.ie).

Alexandre César Rodrigues da Silva is with the State University of Sao Paulo (UNESP), Department of Electrical Engineering, Ilha Solteira, Sao Paulo State, Brazil (acrsilva@dee.feis.unesp.br)

This article is an extended version of a presentation at the REV2009 conference, held at University of Bridgeport, Bridgeport, CT, USA in June 2009. Submitted, July, 6, 2009. Published as resubmitted by the author(s) on Jul, 9, 2009.