The 14th International Scientific Conference
eLearning and Software for Education
Bucharest, April 19-20, 2018
10.12753/2066-026X-18-000

WEB-SERVICE BASED THIN CLIENT FOR TELE-MEASUREMENT

George-Alex Stelea, Vlad Fernoaga, Cristinel Gavrila, Dan Robu
Department of Electronics and Computers, Transilvania University of Brasov, Romania
george.stelea@unitbv.ro, vlad.fernoaga@unitbv.ro, cristinel.gavrila@unitbv.ro, robu.dan@unitbv.ro

Abstract*: As online education services grown in popularity, the need to offer an alternative solution to hands-on measurement activities, especially in the engineering related fields, became one of the top priorities. Blended learning approach, in which both theoretical notions and experimental procedures are available, is, nowadays, a must for students, teachers and researchers. This paper presents a thin client implementation for a web-service browser based tele-measurement of electronic devices and circuits. The thin client offers a web-service based visualization of tele-measurement results as well as multiple-choices in aspects of formats and representations and displays a customized appearance adapted to the device screen size, the available input methods (touch screen, mouse) and the available bandwidth, covering the demands in respect of quality of service and quality of experience. The paper also presents key aspects regarding the thin client structure and implementation, the communication with a LabVIEW based workbench server as well as the results obtained after performing tele-measurements using different terminals (graphical representation and experimental and simulated results).*

Keywords: *Thin clients; tele-measurement; responsive web design; content to terminal adaption; presentation layer; web services.*

I. INTRODUCTION

Modern Web and Cloud Computing are compliant with a broad range of mobile terminals (smartphones or tablets) that run "thin clients" [1]. These clients are portable, cross-platform and cross-device compatible, can centralize the information and management tasks, are less expensive and - what defines them as ”thin” - they are designed to consume less computing power and energy for their tasks. A thin client uses web-services to connect with applications running on a server, in a way that only their *presentation* layer is accessed via web browsers [2]. Operating a thin client requires then a lower resource consumption. A thin client merely represents the user interface, while data processing is done by the server.

Compared with “fat clients” (for "rich content"), the biggest advantage of thin clients is their ease of operation. Thin clients only run tasks needed to *access* the centralized applications. Thus, they operate consistently, with minimal dependencies on the allocation of computational resources. This also allows a very simple management of centralized or decentralized control systems, with reduced overhead - an almost unlimited number of thin clients can be managed by simply assigning / validation of configurations (turning their applications on and off quickly results in a significant service advantage for the end user, especially compared with remote clients needing lengthy installations).

II. EDUCATIONAL USE-CASES

We have developed a "thin client" for a web-service browser based presentation (which does not require applications or other dependencies to be installed) for the tele-measurement of electronic devices and circuits, available at: *http://vlab.unitbv.ro/velab* .

The use-case we focused-on is the real and emulated tele-measurement of BJT – Bipolar Junction Transistors. Remote-experiment stimuli are parsed into a Spice simulation that enables the students to compare “reality with expectations” – ”practical with theoretical”. The “Rich Content” starting web-page, *https://vlab.unitbv.ro/VELab/bjt/bjt.stimuli.php*, includes the schematic under test, device data sheets, recommended stimuli, explanations on manipulations. There is also a “Mini web-page”, *https://vlab.unitbv.ro/VELab/bjt/bjt.stimuli.mini.php* recommended initially as thin-client starting point of the tele-measurement.

III. TECHNOLOGIES AND DEMONSTRATOR

The LabVIEW based workbench server – running in the Intranet of the vlab.unitbv.ro machine (that represents the Internet web server) – is publishing TXT and XLS outputs. The thin client we developed offers a web-service based visualization of tele-measurement results, a multiple-choice of formats and representations based on a graphical user interface (GUI) and can be accessed simply via common browsers without having to install any additional plugins or third party dependencies. This approach has two key advantages: security problems (which may occur when using third party software) are eliminated and resource consumption is drastically decreased. Operating a thin client application allows low cost hardware as it requires less computing power. A thin client merely represents the user interface as the data processing is done by a server which allows a highly individual and centralized deployment of workspaces.

3.1. Content to terminal adaption

The thin client we developed has a responsive design [3] (using standardized HTML5, CSS3, JavaScript [4], jQuery [5] and the Bootstrap Framework – shown in Figure 1) and takes into account the different requirements of the terminals. Our solution is proving how applications can adapt their presentations to the particular terminals, in a clear and user-friendly visualization. The criteria we took into consideration for the customized appearance are, in addition to the size of the display device, the available input methods (touch screen, mouse) and the bandwidth of the Internet connection.

```html
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

<title>Web-service Based Thin Client for Tele-measurement</title>

<meta content='width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0' name='viewport' />
<meta name="viewport" content="width=device-width" />
<meta name="theme-color" content="#31b9d8" />

<!-- Bootstrap core CSS     -->
<link href="assets/css/bootstrap.min.css" rel="stylesheet" />
<link href="assets/css/animate.min.css" rel="stylesheet"/>
<link href="assets/css/light-bootstrap-dashboard.css?v=1.4.0" rel="stylesheet"/>
<link href="assets/css/demo.css" rel="stylesheet" />


<!--     Fonts and icons     -->
<link href="http://maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css" rel="stylesheet">
<link href='http://fonts.googleapis.com/css?family=Roboto:400,700,300' rel='stylesheet' type='text/css'>
<link href="assets/css/pe-icon-7-stroke.css" rel="stylesheet" />

<!--   Core JS Files   -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery.form/4.2.2/jquery.form.min.js" in"></script>
<script src="assets/js/jquery.3.2.1.min.js" type="text/javascript"></script>
<script src="assets/js/bootstrap.min.js" type="text/javascript"></script>
```

Figure 1. Bootstrap and CSS3 media queries integration

From a historical perspective on building web applications, the desktop version provides the normal view of the site (as mentioned above). For mobile devices (smartphones and tablets) an additional independent template had to be added to the classic method [6]. But, in our approach, the Responsive Web Design/Development results in only one version of the site - this independently adapts itself to the available environment. This is particularly visible in the layout, which changes according to the width and resolution of the browser window.

The graphical structure of the responsive "thin client" is based on the requirements of the particular device that displays it. This applies, in particular, to the arrangement and display of individual elements, such as navigations, page columns and texts, but also the use of different input methods of mouse (click, drive over) or touch screen (type, swipe). The technical basis for this is represented by the newer web standards HTML5, CSS3 (particularly the media queries) and JavaScript.

The size and resolution of displays on laptops, desktops, tablets, smartphones, and even TV screens can vary considerably. The aim of this solution is to have a "thin client" which adapts its presentation so that it is as clear and user-friendly as possible to each viewer. The criteria for the customized appearance are, in addition to the size of the display device, for example, available input methods (touch screen, mouse) or the bandwidth of the Internet connection - as already mentioned.

In particular, block elements are treated differently. In the sections where the design grid uses several columns as shown in Figure 2, the blocks variably adapt to the width of the respective viewport. For small resolutions, if a representation no longer fits next to each other, it is alternatively re-positioned. Images are scaled to a maximum enabled by their surrounding elements - but not beyond their native size - and textures for backgrounds have the overflow hidden CSS 3 property. Backgrounds with motif are also omitted in small representations. Logos are scaled down like pictures or replaced by a smaller signet for large and detailed graphics. The vertical menu bar that needs a lot of width is repositioned as a collapsed list viewable only on click as presented in Figure 3.

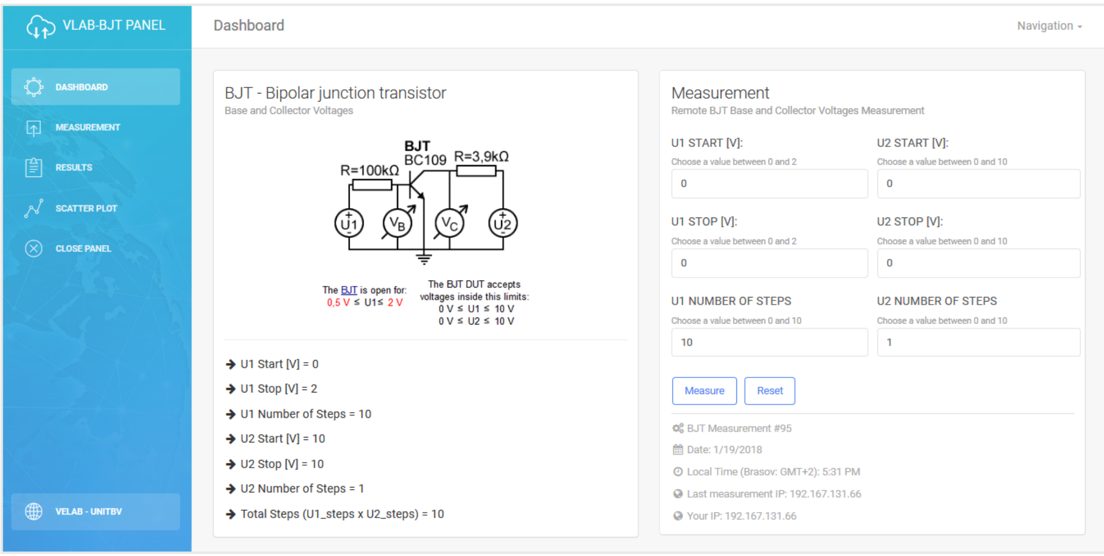

Figure 2. “Thin client” display on a laptop/desktop device resolution

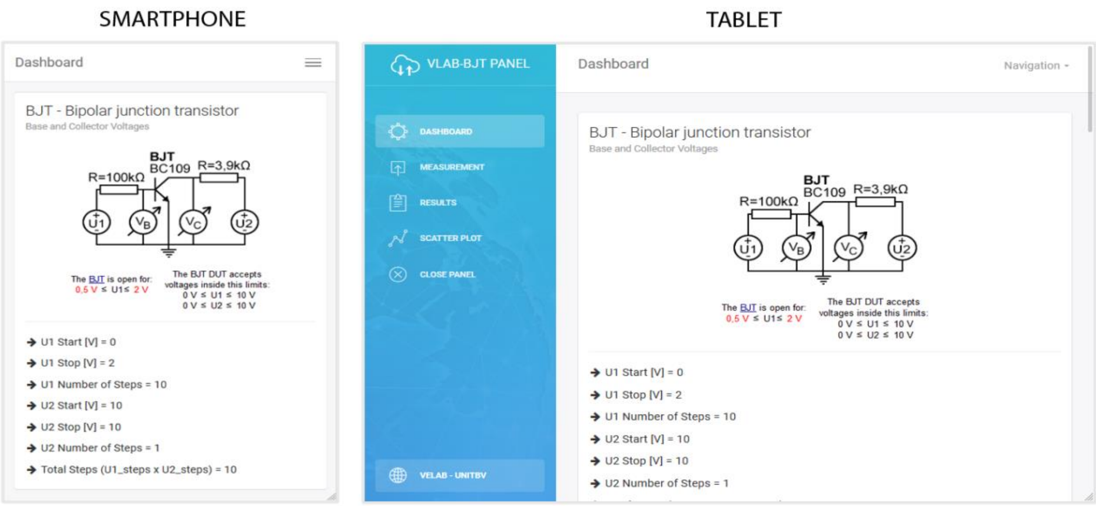

Figure 3. “Thin client” display on a smartphone (left) and tablet (right) device resolution

3.2. Tele-measurement & presentation layer

The demonstrator "thin client" is available at http://vlab-bjt.9net.ro . It was equipped with PHP and JavaScript parsing libraries for XLS [7], CSV [8], XML/JSON and even plain text formats in order to be able to expose data and to generate analytics and reports (see Fig.1) in a large spectrum of graphical representations. Using AJAX technology, the possibility to perform remote measurements from the "thin client" directly to the server is also supported, as shown in Figure 4.

The measured data is collected and parsed using web services and displayed in parallel, as measured and simulated results, as shown in Figure 5.

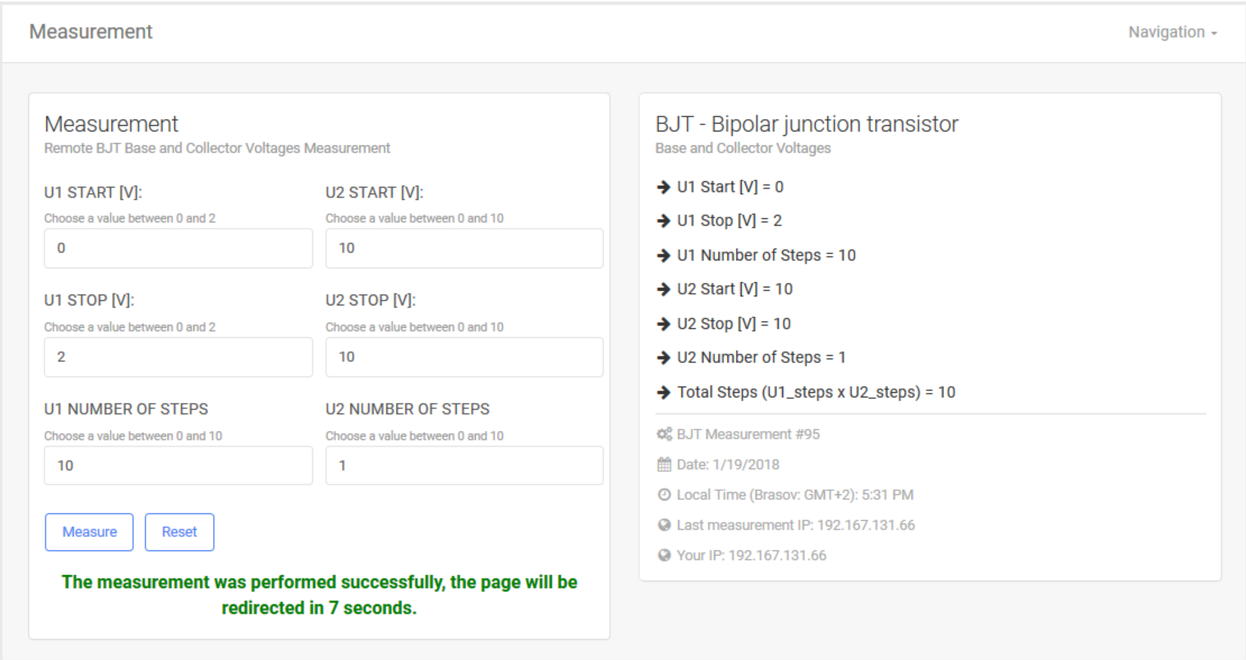

Figure 4. Remote “thin client” measurement


BJT Results

Navigation

Experimental Results

Bipolar junction transistor

U1 (DATA IN)	V_C (DATA OUT)	V_B (DATA OUT)	U2 (DATA IN)
EXPERIMENTAL			
0.00	10.00	0.035	10.00
0.20	10.00	0.235	10.00
0.40	10.00	0.435	10.00
0.60	9.39	0.586	10.00
0.80	6.57	0.630	10.00
1.00	3.36	0.650	10.00
1.20	0.65	0.660	10.00
1.40	0.20	0.665	10.00
1.60	0.17	0.665	10.00
1.80	0.16	0.665	10.00

Simulated Results

Bipolar junction transistor

U1 (DATA IN)	V_C (DATA OUT)	V_B (DATA OUT)	U2 (DATA IN)
SIMULATED			
0.00	10.00	0.001	10.00
0.20	10.00	0.200	10.00
0.40	10.00	0.400	10.00
0.60	9.77	0.585	10.00
0.80	7.10	0.652	10.00
1.00	3.72	0.674	10.00
1.20	0.68	0.687	10.00
1.40	0.24	0.689	10.00
1.60	0.21	0.689	10.00
1.80	0.19	0.689	10.00
2.00	0.18	0.689	10.00

Figure 5. Experimental and Simulated results are presented side by side

To generate the scatter plots, *pChart* - a PHP class oriented framework, designed to create anti-aliased charts - was used, as shown in Figure 6. The analytics and reports can be previewed and downloaded in a large spectrum of graphical representations [9] as diagrams, graphs, scatter plots

(Figure 7), bar charts, line charts, X-Y charts, to display tabular numeric data, functions or other qualitative structures.

The “thin client” has been built Class-based in such a way that the presentation of the graphs can be extended using interface solutions and modules as Kibana, Microsoft Power BI, etc.

```php
<?php
 /* CAT:Scatter chart */
 /* pChart library inclusions */
 include("class/pData.class.php");
 include("class/pDraw.class.php");
 include("class/pImage.class.php");
 include("class/pScatter.class.php");
  /* Create the pChart object */
 $myPicture = new pImage(530,590,$myData);
 /* Turn of Anti-aliasing */
 $myPicture->Antialias = FALSE;
 /* Add a border to the picture */
 $myPicture->drawRectangle(0,0,599,599,array("R"=>255,"G"=>255,"B"=>255));
 $myPicture->setFontProperties(array("FontName"=>"fonts/verdana.ttf","FontSize"=>9));
 /* Set the graph area */
 $myPicture->setGraphArea(50,30,520,520);
 /* Create the Scatter chart object */
 $myScatter = new pScatter($myPicture,$myData);
 /* Draw the scale */
 $scaleSettings = array("XMargin"=>15,"YMargin"=>15,"Floating"=>TRUE,"GridR"=>200,"GridG"=>200,
 "GridB"=>200,"DrawSubTicks"=>TRUE,"CycleBackground"=>TRUE);
 $myScatter->drawScatterScale($scaleSettings);
 $myScatter->drawScatterLegend(330,20,array("Mode"=>LEGEND_HORIZONTAL,"Style"=>LEGEND_NOBORDER));
 /* Draw a scatter plot chart */
 $myPicture->Antialias = TRUE;
 $myScatter->drawScatterPlotChart();
 /* Render the picture (choose the best way) */
 $myPicture->autoOutput("pictures/example.example.drawScatterBestFit.png");
 ?>
```

Figure 6. pChart - PHP charting library object oriented integration

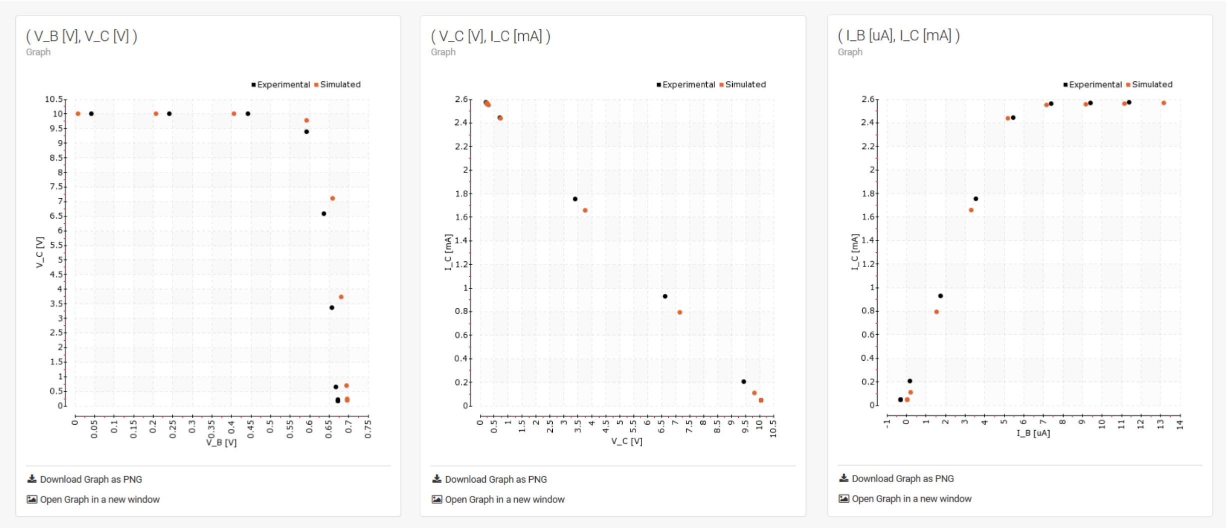

Figure 6. Scatter plots – chart presentation of BJT real and emulated “parallel” tele-measurement - students can detail low signal voltage gain on the right side of the 1st plot, the ”static load” oblique (with a slope of exactly -1 / 3.9 kΩ) on the 2nd plot or the β factor on the left side of the 3rd plot.

IV. CONCLUSIONS

Tele-measurement is one of the key components for ensuring blended educational services - hands on and remote (online), guided / individual / peer-to-peer etc. - in which both theoretical notions and experimental procedures are available. Being able to remotely conduct measurements - via a remote and virtual laboratory (VL) - has a major impact in the process of learning basic and advanced concepts in various disciplines of Science and Engineering (domains where experimental skills and manipulations are of a specific educational importance).

One of the most feasible approaches to ensure the access to VL services through a network as well as to provide a friendly and intuitive interface for the users is by using “thin clients”. The clients we developed operate consistently, with minimal dependencies and with almost insignificant computational resources.

These features allow a very simple management of centralized or decentralized control, with reduced overhead, and an almost unlimited number of thin clients. The unlimited number of thin clients is a critical aspect in providing tele-measurement services, because the probability of having a high density of clients is significant, especially in University environments. In order to ease the problems of ”concurrent access”, the tele-measurement implemented is ”telegram” based (an ”wysiwyg” - ”what you see is what you get” approach, where stimuli are returned together with the results).

The “thin client” implementation described in this paper offers the possibility to instantiate and trigger remote measurements as well as a web-service based visualization of tele-measurement results and can adapt its presentations based on terminals characteristics, in a clear and user-friendly visualization.

When performing tele-measurements, the measurement results (usually numeric values) are the most important, however, having different graphical interpretations has a high contribution to a better understanding of the results retrieved. The implemented “thin client” provides various graphical representations such as diagrams, graphs, scatter plots, bar charts, line charts, X-Y charts. This higher degree of vizualisation, combined with the responsive design feature, has, from the user perspective, a crucial impact on the quality of experience (QoE).

Reference Text and Citations

[1] L. Deboosere, B. Vankeirsbilck, P. Simoens, F. De Turck, B. Dhoedt, P. Demeester: *Cloud-Based Desktop Services for Thin Clients*. In: IEEE Internet Computing, Vol. 16, No. 6, pp. 60-67, 2012.

[2] L. Deboosere, P. Simoens, J. De Wachter, et al: *Grid design for mobile thin client computing.* In: Future Generation Computer Systems, Volume 27, Issue 6, 2011, pp. 681-693, ISSN 0167-739X,

[3] W.Jiang, M.Zhang, B.Zhou, et al: *Responsive web design mode and application*, 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), Ottawa, ON, 2014, pp. 1303-1306.

[4] G. Richards, S. Lebresne, B. Burg, J. Vitek: *An analysis of the dynamic behavior of JavaScript programs*. In: Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '10). ACM, New York, NY, USA, 2010. pp. 1-12.

[5] E. Andreasen, A. Moller: *Determinacy in Static Analysis for jQuery*. In: Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications - OOPSLA '14, Portland, Oregon, USA. 2014, pp. 17-31.

[6] M. Nebeling, M.C. Norrie: *Responsive Design and Development: Methods, Technologies and Current Issues*. In: Daniel F., Dolog P., Li Q. (eds) Web Engineering. ICWE 2013. Lecture Notes in Computer Science, vol 7977. Springer, Berlin, Heidelberg, pp. 510-513.

[7] C. Zhao, L. Zhao, H. Wang: *A spreadsheet system based on data semantic object*. In: 2010 2nd IEEE International Conference on Information Management and Engineering, Chengdu, 2010, pp. 407-411.

[8] S. Adam, U.P. Schultz: *Towards Tool Support for Spreadsheet-based Domain-specific Languages*. In: SIGPLAN Not. Journal, New York, NY, USA, Vol. 51, No. 3, March 2016 pp. 95-98.

[9] S.J. Qin: *Process data analytics in the era of big data.* In: AIChE Journal, Volume 60, No. 9, 2014, pp. 3092-3100, ISSN: 1547-5905.