

Webdemos: An Interactive, Web-Based Visualization and Simulation Framework for Open Access

David Rörich, Michael Bernhard, Thomas Handte and Stephan ten Brink

Institute of Telecommunications

University of Stuttgart

70569 Stuttgart, Germany

Email: {roerich, bernhard, handte, tenbrink}@inue.uni-stuttgart.de

<http://www.inue.uni-stuttgart.de/webdemo>

Abstract—We introduce the web-based simulation and visualization tool *Webdemo*, designed for supplementing science, technology, engineering and mathematics (STEM) courses in higher education with interactive examples. The flexible simulation system supports a great variety of visualizations and mathematical operations. To ensure open access, the web front end does neither require additional software nor user registration and can be viewed on every device capable of web browsing. The creation of a new *Webdemo* based on the presented framework is simple and the reuse of templates guarantees a uniform and extensible user interface. A user survey confirms that students appreciate the *Webdemos* as a valuable enrichment of lectures.

I. INTRODUCTION

Many concepts in STEM are best understood when visualized with the help of graphs and figures. Visualizations reveal relationships that are otherwise hard to grasp and convey information in a concise and instructive way. They are therefore of great importance in higher education, especially for teaching courses in the STEM field. Course content becomes more motivating for the learner when presented in an interactive manner, e.g. in such a way that the student can change parameters of a simulated system and receive a direct feedback about the impact of the parameter change [1]. In this way “what happens if” questions can be answered immediately without lengthy calculations.

Especially in engineering, simulations play an important role in the product development process because they can save money, time, and resources. Students should thus learn how to create computer models of physical effects and how to solve complex problems numerically. As simulation techniques are often exclusively taught in specialized courses, it would be desirable to offer both the theoretical concepts and their respective simulation models in the same engineering course.

Web-based technologies enable the deployment of interactive visualization and simulation tools with open access. In an early work, Masson has implemented a web-based simulation covering one specific topic of an electronics engineering course [2]. While some technologies used in that work are still applicable, some appear out-dated from today’s perspective and should be updated to improve flexibility and user experience. Another example of a single-purpose, web-based simulation tool has been presented in [3] and a web front end for the simulation of electronic devices has been introduced

in [4]. The latter work is more generic in that it allows the user to simulate an arbitrary model but still rather specific in that it only simulates the behavior of electronic devices and circuits. Additionally, the user interface relies on the upload of text files, thus requiring knowledge of the simulation core. The nanoHUB platform [5] is a highly generic and web-based simulation tool that supports a variety of programming languages and simulation tools but expects a Java installation on the client machine and requires the user to register. Its access is thus not fully open. Similarly, the demonstration project in [6] requires the user to install proprietary software to use the web-based simulation tools and potential authors of new simulations need a commercial software. In this paper we introduce the *Webdemo* framework: a novel interactive, web-based visualization and simulation environment [7]. It has been developed to supplement communications lectures of undergraduate and graduate students of the electrical engineering study course at the University of Stuttgart. The *Webdemos* are accessible by everyone and do not require a registration thus offering true open access. The framework is implemented by open source software and designed such that the client’s browser requires JavaScript (JS) only but no additional plug-ins. Moreover, the creation of a new *Webdemo* is simple and does not require any web programming knowledge. Consequently, the scope of *Webdemo* authors does not only include teaching staff but also students.

The remainder of this paper is organized as follows: Section II describes the implementation and the usage of the *Webdemo* framework. Some *Webdemo* examples are given in Section III to demonstrate the various possible applications and the flexibility of the framework. In Section IV we analyze the reception of the presented system by means of user evaluations and usage statistics. Finally, Section V concludes the paper.

II. WEBDEMO FRAMEWORK DESCRIPTION

The *Webdemo* framework allows the creation of interactive visualization and simulation tools accessible through a web browser. A *Webdemo* can contain several interactive and descriptive pages which are linked to each other in a fixed sequence (although random access of individual pages is possible through an auto-generated table of contents page). This arrangement resembles a slide show presentation and hence an individual web page of a *Webdemo* is referred to as a slide in the following. The framework is intended to be

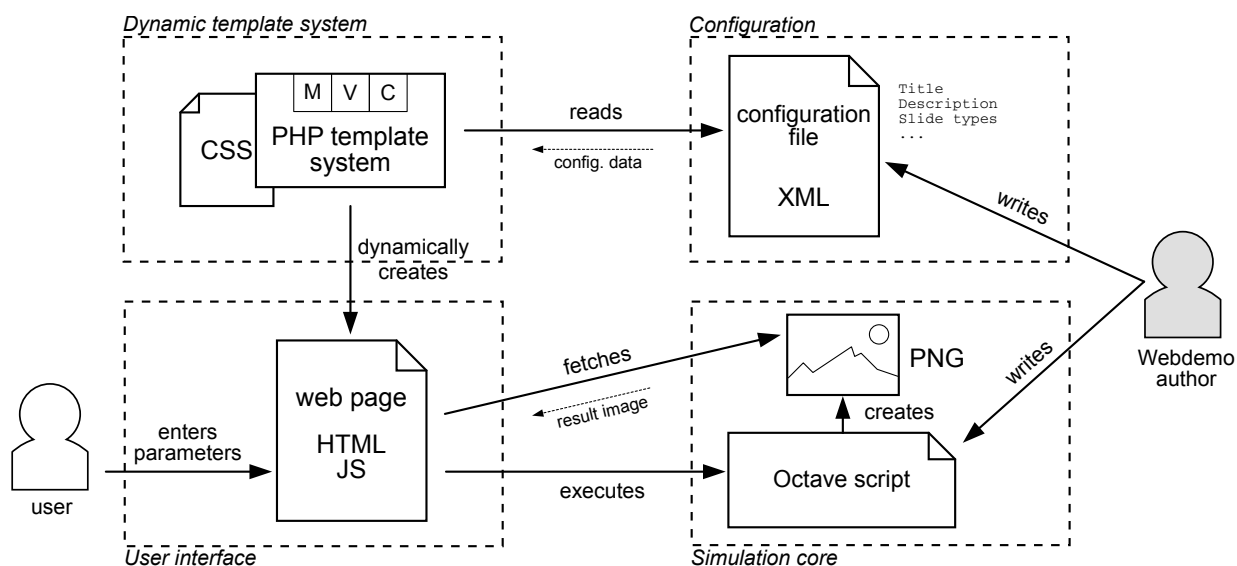


Fig. 1. Diagram of the different components of a Webdemo and their interactions which each other, the user, and the Webdemo author.

used by various authors including faculty members, students and possibly third persons from other institutions. A lot of attention was therefore paid to the simplicity of creating a new Webdemo. Especially, no web programming skills are required. However, it is assumed that a potential Webdemo author has the basic knowledge of scientific programming languages or libraries such as Matlab, GNU Octave or Python’s SciPy that is necessary to write a numerical simulation. The Webdemo framework can be divided into four components which are shown in Fig. 1 and are described in detail in the following.

A. Dynamic template system

The central component of the Webdemo framework is a template system written in PHP controlling the creation of the web pages that are delivered to the client, i.e. the Webdemo user. It is common to all Webdemos and thus guarantees a consistent “look and feel” for the user while hiding the details of the framework implementation from the Webdemo author. As shown in Fig. 1 it consists of three components: model (M), view (V) and controller (C). This architecture separates data management, user interface, and control flow of the dynamic web page creation. The model component parses the configuration file which contains various information about the specific Webdemo such as title, number of slides, input fields, etc. (see section II-C) and provides this information to the controller. According to the requested URL, the controller decides which slide is to be displayed, loads the respective template file and invokes the view component with the Webdemo specific data. The template files store the structure of different slide types and contain PHP instructions on how to build the web page with the Webdemo specific data provided by the model component. As a part of the template system, the templates are common to all Webdemos and their modular design facilitates future extensions of the system, e.g. by new slide types. Finally, the view component creates and delivers the web page which is an HTML document including JS scripts, as depicted in Fig. 1. The appearance, i.e. the colors, sizes, and positions of web page elements as well as font types,

is defined in a cascaded style sheet (CSS) which is common to all Webdemos as well.

B. Simulation core

In a web-based, interactive simulation tool the user enters a set of generally arbitrary parameter values and requests a graphical or numerical representation of the simulation result. Since the number of possible parameter sets can be huge¹, the precalculation of all possible simulation results is impossible because computational and storage resources are limited. Subsequently, all calculations that depend on the user input have to be executed on demand. In principle, there are two design options for implementing such on-demand calculations: (i) on the server side or (ii) on the client side.

Executing calculations on the client side allows a prompter response to the user input, for once the web interface to the simulation has been loaded, no further communication with the server is required and thus no transmission delay occurs. However, the software available to execute the required calculations is limited by the user’s computer configuration. The browsers of most Internet users support JS which allows for plotting data using special libraries [8]. Nevertheless, these libraries do not offer all features of dedicated scientific software in terms of mathematical operations and graphical output capabilities. Furthermore, the reuse of existing simulation models would be limited with JS. On the other hand, the user should not be confronted with barriers by requiring the installation of additional software. This software could be unavailable for the user due to license restrictions or due to incompatibility with the user’s hardware or operating system.

Additionally, the user’s hardware configuration is unknown and can range from a smartphone to a powerful personal computer. Accordingly, the computational time required for client side simulations can vary considerably. In contrast, the computational time of server side simulations is predictable,

¹Example: two double precision floating-point parameters result in $2^{128} \approx 3.4 \cdot 10^{38}$ possible parameter sets

since the hardware is known in advance, and can thus be kept in acceptable limits.

For these reasons, implementation option (i) has been chosen for the Webdemo framework, thus reducing the software requirements to a minimum that can be assumed to be fulfilled on virtually every client. As a consequence, the Webdemos can also be viewed on mobile devices like smartphones or tablet computers.

It is one aim of the Webdemo framework to make students familiar with scientific software and simulation models by providing them with practical examples which they can study immediately through the Webdemo itself but also in more detail through inspection of the source code of the underlying simulation script. In order to make the source code usable by everyone, the necessary compiler or interpreter should be free. In addition, it should exist for a large number of computer platforms and operating systems. Therefore, GNU Octave [9] has been chosen as simulation software because it is open source and possesses powerful built-in visualization routines as well as mathematical functionalities like vector algebra, filters, and transforms. The simulation scripts written in Octave are therefore independent of external libraries and still compact with respect to the source code length which makes them both easily deployable and understandable for the user. Every Webdemo exhibits an individual, single Octave function which can be downloaded and thus run independently from the Webdemo. In addition, Octave is largely compatible to the widespread commercial software Matlab. Consequently, the Webdemo simulation scripts developed for Octave can in principle be executed with Matlab as well.

As shown in Fig. 1, the simulation core of a Webdemo consists of an Octave function that is called by the JS script² when the user requests the simulation result. All simulation parameters and a file name are passed to that function which is executed on the sever side. The result of the function is a portable network graphics (PNG) image file with the given name. This image file is then sent back to the client and displayed in the user's browser.

C. Configuration

The content of a Webdemo is defined in a configuration file using the extensible markup language (XML). Among others, it contains meta data about the Webdemo like author, title, date of creation, and the associated lecture or project. An extract of a configuration file is given in Listing 1 which demonstrates that the structure of the XML file is self-explanatory.

```
<meta>
  <author>David Rörich</author>
  <author_mail>roerich@inue.uni-stuttgart.de</author_mail>
  <teaching_module>Übertragungstechnik II</teaching_module>
  <title>Mach-Zehnder Modulator</title>
  <subtitle>Operational modes and...</subtitle>
  <version>1.2</version>
</meta>
```

Listing 1. Definition of the Webdemo meta data in the XML configuration file.

²The JS script actually calls a PHP function on the server which in turn calls the Octave function.

Additionally, an arbitrary number of slides can be defined where every slide can either be a description or a demo slide. Description slides can contain text, images and mathematical equations. Latex markup can be applied for setting equations which are rendered using MathJax [10]. A demo slide contains various input fields (here: controls), a button that triggers the server side simulation, and an area for displaying the simulation result in the form of an image. Currently, the following control types are available

- Text field for arbitrary numbers or strings
- Drop down menu for selecting an option
- Check box for Boolean parameters
- Group of radio buttons for selecting an option
- Slider for choosing a value out of a predefined range.

For every control the Webdemo author can define a short text (XML tag <text>) appearing next to the field and a tooltip text (XML tag <tooltip>) which offers more detailed information. As shown in Listing 2, also the short text can include Latex math code.

```
<slide>
  <type>demo_wide</type>
  <header>Transfer characteristic</header>
  <control>
    <type>text</type>
    <id>U_dc</id>
    <value>-0.5</value>
    <text>$U_{\mathrm{1,DC}}/U_{\pi}$</text>
    <tooltip>Bias voltage norm...</tooltip>
  </control>
  <script_interface>
    octave_fun(U_dc, filename)
  </script_interface>
</slide>
```

Listing 2. Definition of a demo slide in the XML configuration file. This example includes a text input field and the function interface of the simulation core.

Every slide definition must include a script interface defining the name of the Octave function that implements the simulation and the parameter list of that function. The values of the slide controls are identified by their <id> tag. They are passed to the function along with the file name that is chosen randomly by the template system in order to prevent conflicts between multiple user sessions.

D. User interface

Figure 2 shows an example of a Webdemo description slide. Its layout is optimized for a screen resolution of 1024 px × 768 px which is compatible to most projectors and can thus be used in talks and lectures. The Webdemo user interface is organized as a slide show presentation with a navigation footer and a content area. The navigation footer contains links to the previous and the next slide as well as the current slide number that links to a table of contents page. Beneath the navigation footer, meta information about the Webdemo and links to the imprint and privacy policy are displayed. A special feature is the usage statistics that can be accessed through the footer as well.

The screenshot in Fig. 3 shows a detail of a demo slide. Footer and header are omitted here for brevity as they are

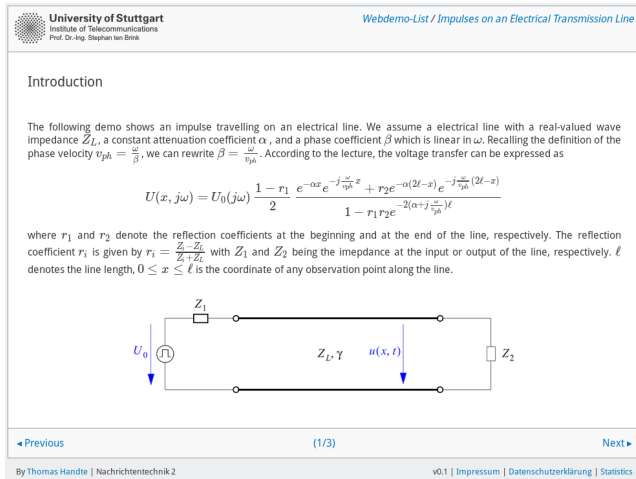


Fig. 2. Screenshot of a description slide of the Webdemo “Impulses on an Electrical Transmission Line”. Online: <http://www.inue.uni-stuttgart.de/webdemo/etl>

similar to Fig. 2. The central area of the slide is reserved for the image representing the simulation result. Beneath the image area, controls are arranged that allow user input. Whenever a demo slide is loaded, the simulation is executed using the default parameter values. A JS script shows a progress bar providing feedback about the elapsed and remaining calculation time. In order to estimate the calculation time, the template system measures the elapsed time for every call of the Octave function and calculates the mean of all measurements. While actual simulation times in general differ from that mean, it has been found that they are sufficiently exact in most cases. The simulation time of the existing Webdemos ranges from 2 to 20 seconds and is thus larger than the usual response time of Internet pages. The animated progress bar indicates to the user that the calculation is ongoing and that this behavior is no error but normal. As soon as the simulation core has generated the result image, the JS script requests the image from the server and displays it in place of the progress bar. The user can now change the simulation parameters through the controls and initiate a recalculation using the “Update” button. Beneath the update button a download link provides access to the source code of the underlying Octave function.

III. WEBDEMO EXAMPLES

The following Webdemo examples give an overview of the versatile applications which can be realized with the Webdemo framework and demonstrate how students can profit by them.

Figure 3 shows the screenshot of a Webdemo that calculates the output signal of a single-mode optical fiber for a given input signal. Both signals are depicted in the same graph to make the signal distortion caused by the physical effect of chromatic dispersion (CD) clear. The user can change the fiber length, the chromatic dispersion coefficient of the fiber and the symbol rate of the information signal and observe their impact on the signal distortion. In addition, the pulse shape of the input signal can be chosen from three options which affects the signal bandwidth and the signal distortion. While the effect of CD can be described by a fairly simple expression in the frequency domain, its mathematical time domain representation is complicated and gives little insight into the actual form

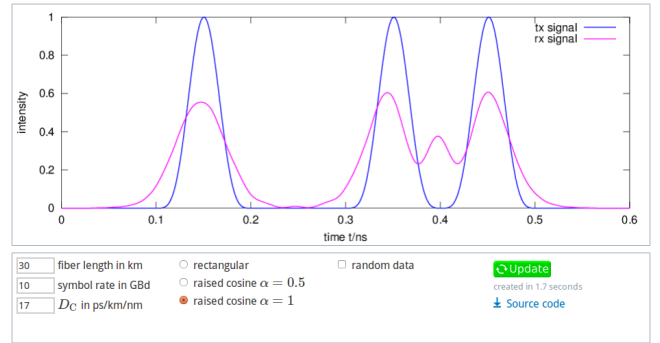


Fig. 3. Screenshot of one of the demo slides of the Webdemo “Chromatic Dispersion”. Online: <http://www.inue.uni-stuttgart.de/webdemo/cd>

of a signal affected by CD. This Webdemo helps students to understand this effect better and makes clear what role different parameters play. For example, it is an instructive exercise to gradually increase the fiber length for a given symbol rate until the transmitted impulses become indistinguishable at the receiver. This is the maximum transmission distance that can be reached for that symbol rate (without CD compensation).

The layout of the CD Webdemo features a wide aspect ratio which has been chosen because more resolution is required on the time axis than on the signal intensity axis. There are other cases in which a different aspect ratio is favorable, especially when the horizontal and vertical resolution should be approximately equal. The Webdemo framework therefore supports two types of demo slides, that can be configured in the XML configuration file: one with a wide aspect ratio and one with a 4:3 aspect ratio. The Webdemo shown in Fig. 4 is an example of a 4:3 aspect ratio demo slide.

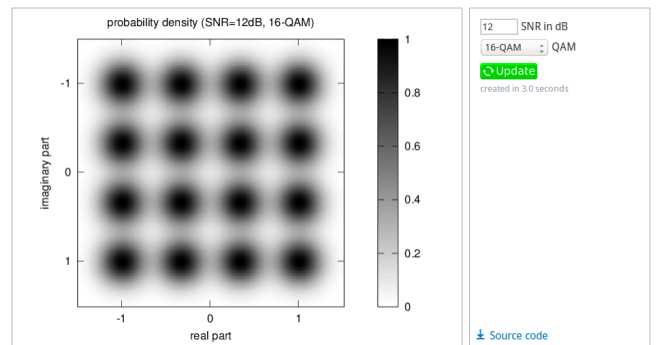


Fig. 4. Screenshot of one of the demo slides of the Webdemo “QAM”. Online: <http://www.inue.uni-stuttgart.de/webdemo/qam>

This simulation calculates the probability of the inphase and quadrature components of a quadrature amplitude modulation (QAM) signal affected by additive white Gaussian noise (AWGN). The range of probability values is normalized to 1 and mapped to greyscale colors while the horizontal and vertical axis represent the inphase and quadrature signal component, respectively. Both signal components have the same range of values, which is why a 4:3 aspect ratio has been chosen here. This demo slide has two input parameters: the signal-to-noise ratio (SNR) in dB and the QAM order. The SNR can be any real number and is therefore entered through a text field. The QAM order is, by contrast, a positive integer

number and a power of two. Since practical values for the QAM order usually do not exceed 2048, a drop down menu offering all possible values for this parameter is considered the most suitable choice. In communications, the SNR of a signal is a basic concept of great importance. It is, however, an abstract quantity and mapping it to a qualitative information requires experience. This Webdemo illustrates the SNR in a way that helps students to interactively develop a feeling for the meaning of a certain SNR value. It also provides an intuitive explanation of the fact that a higher order QAM signal conveys more information but at the same time is more susceptible to noise.

The above Webdemos are examples of classical graphical representations of mathematical functions of one and two variables, respectively. More sophisticated visualizations are possible, however, using the varied plotting functions of Octave. Especially low level functions that plot basic graphic elements like lines, circles, points, and text allow to create arbitrary graphical output in principle. The demo slide shown in Fig. 5 is an example of such an advanced plotting application.

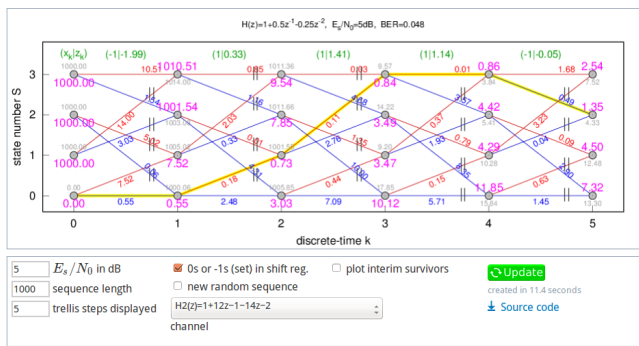


Fig. 5. Screenshot of the Webdemo “Equalization using the Viterbi algorithm”. Online: <http://www.inue.uni-stuttgart.de/webdemo/viterbi>

This Webdemo visualizes the famous Viterbi algorithm for the maximum-likelihood estimation of the transmit sequence in a communication system [11], [12]. It is an important algorithm in this field and is used in GSM³ mobile telephones, for example. It is also a popular exam problem. The plot mimics the way one would solve this problem on a sheet of paper and thus can help students to check their own solutions as an exercise.

The examples presented in this section are just a subset of the variety of visualizations that are possible with the Webdemo framework. Subplots and inset plots enable better exploitation of the available image area by displaying several plots next to each other, for instance. The interested reader is pointed to <http://www.inue.uni-stuttgart.de/webdemo> for a list of all Webdemos available.

IV. USER EVALUATION AND STATISTICS

The Webdemos have first been introduced to the lectures of the Institute of Telecommunications at the University of Stuttgart in October 2013 as supplementary material and a visualization tool supporting the presentation of the lecture

content. In order to evaluate the reception of the Webdemos we have conducted a user survey and analyzed Internet traffic statistics.

A. User evaluation

We asked students to complete a questionnaire containing, among others, one question about the usefulness of the Webdemos. A Likert-type scale has been used to analyze the participants’ level of agreement towards the statement “Webdemos help to illustrate the topics of the lecture”. The symmetric scale consists of five possible answers ranging from 1 (“agree”) to 5 (“disagree”) which are assumed to be equidistant. Our results are taken from surveys in the winter term of 2013 and the summer term of 2014. The lectures in which the questionnaires have been issued contained undergraduate and graduate courses as well as international and German courses. Our survey thus covers a wide range of educational and cultural backgrounds. The evaluation of the questionnaire, which has been completed by 149 students in total, is shown in Fig. 6.

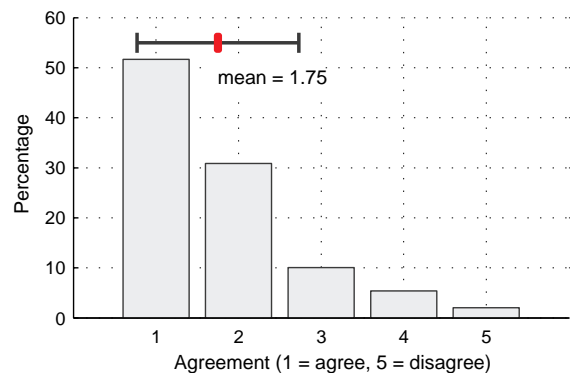


Fig. 6. Percentage of levels of agreement to the statement “Webdemos help to illustrate the topics of the lecture” in a survey of 149 students. The horizontal line indicates the standard deviation which is 0.98.

The results show that the majority (83%) of students fully or partially agree that the Webdemos are a helpful extension of the lecture. A more detailed analysis revealed that the acceptance in advanced courses was 95% and thus higher than in introductory courses (74%). This could indicate that the level of some Webdemos needs adaption to the prior knowledge of these students. Nevertheless, the overall acceptance is high and confirms the usefulness of the presented system.

B. Visitor statistics

To gain insight into the frequency of usage of the Webdemos, the free web analytics tool Piwik [13] has been installed on the server hosting the available Webdemos. The analytics tool is configured to filter out all Internet protocol (IP) addresses from inside the Institute of Telecommunications so that the statistics are not biased by visits from the Webdemo developers and authors. All page requests from the same IP address that have a time difference of less than 30 minutes are considered one visit. It should be noted that the actual number of visits could be higher since some browsers block Piwik by default. Tab. I summarizes some statistics that have been recorded during approximately one year of operation.

³Global System for Mobile Communications

TABLE I. VISITOR STATISTICS OF THE WEBDEMOS

Date range	22.08.2013 to 11.09.2014
Visits	3461
Average visit duration	5m21s
Visits from Stuttgart	67%
Top referrer / ratio	www.inue.uni-stuttgart.de / 50%
Distinct countries	79
Users with Java plug-in	76%
Users with Flash plug-in	84%

The number of visits may appear small but is actually relatively high when compared to the number of students. The majority of visits originates in the region of Stuttgart where the University and most of its students are located. Additionally, half of the visitors has been referred to the Webdemos from the homepage of the Institute of Telecommunications. It can therefore be estimated conservatively that about 50% of the visitors were students pointed to the Webdemos in one of the lectures. The number of students that have attended lectures supplemented by Webdemos in the considered time period is roughly 400. Accordingly, using the previous estimation this yields an average number of four visits per student. Interestingly, the Webdemos have been visited from a total number of 79 different countries showing their attractiveness beyond their application in the before-mentioned lectures. It is also noted that about a fourth of the visitors had no Java plug-in installed or activated in their browsers. A similar observation is made for the Flash plug-in. This confirms our assumption that no additional software should be required to view the Webdemos in order to maximize their accessibility.

V. CONCLUSION

We have shown the Webdemo framework to be a highly flexible tool for open access, web-based learning in the STEM field. Students are offered an interactive simulation environment which is freely accessible without registration and which only requires an Internet browser with JS enabled. As our survey results show, students regard the Webdemos as a useful extension of the lectures. To gain a deeper insight into the simulation models applied, users can download the underlying simulation script of a Webdemo and run it on their own computers using the free software Octave. The server side implementation of the simulations makes the Webdemos largely independent of the user's hard- and software configuration and keeps the simulation time predictable. Therefore, the Webdemos can be viewed on mobile devices as well. Moreover, the simplicity of creating a Webdemo encourages potential authors to write new Webdemos which is confirmed by the fact that 20 Webdemos have been created within one year. The framework implementation guarantees a uniform user interface which can be changed and extended for all Webdemos at once.

A possible extension is the introduction of new input types like matrices or file uploads which might be required by future Webdemo applications. Furthermore, the simulation results could take another form in versions to come, for instance that of an animation or an audio file.

ACKNOWLEDGMENT

The authors would like to thank Müge Bayraktaroğlu, Bushra Bashir Chaoudhry, Marc Gauger and Klaus Oestreich for contributing to the success of the Webdemo framework with their numerous and inspiring Webdemos.

REFERENCES

- [1] H. Lu and P. Lin, "Effects of interactivity on students' intention to use simulation-based learning tool in computer networking education," in *14th International Conference on Advanced Communication Technology*, 2012, pp. 573–576.
- [2] A. M. Masson, "Web-based simulations for computer-assisted learning in the higher education sector," *Engineering Science and Education Journal*, vol. 8, no. 3, pp. 107–114, 1999.
- [3] I. Belski, "Acquiring a holistic picture: The 4screens web-based simulator helping students to unify behaviours of electronic systems," *8th IEEE International Conference on Advanced Learning Technologies*, pp. 154–158, 2008.
- [4] H. Zhuang, J. He, W. Deng, X. Zhang, X. Zhu, Q. He, and M. Chan, "A web-based education platform for nanoscale device modeling and circuit simulation," in *4th Interdisciplinary Engineering Design Education Conference*, Mar. 2014, pp. 84–87.
- [5] G. Klimeck, M. McLennan, M. S. Lundstrom, and G. B. Adams III, "nanoHUB.org - online simulation and more materials for semiconductors and nanoelectronics in education and research," in *8th IEEE Conference on Nanotechnology*, Aug. 2008, pp. 401–404.
- [6] Wolfram Research. The Wolfram demonstrations project. [Online]. Available: <http://demonstrations.wolfram.com>
- [7] M. Bernhard, T. Handte, D. Rörich, and S. ten Brink. (2014, Sep.) Webdemos at the Institute of Telecommunications. Institute of Telecommunications, University of Stuttgart. [Online]. Available: <http://www.inue.uni-stuttgart.de/webdemo>
- [8] O. Laursen and D. Schnur. (2014, Sep.) Flot, a JavaScript plotting library. [Online]. Available: <http://www.flotcharts.org>
- [9] J. W. Eaton, D. Bateman, and S. Hauberg, *GNU Octave*, ed. 3 for octave version 3.0.2, 1. print., sept. 2008 ed. Bristol: Network Theory Ltd., 2008.
- [10] MathJax Consortium. (2014, Sep.) MathJax, a JavaScript display engine for mathematics. [Online]. Available: <http://www.mathjax.org>
- [11] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.
- [12] G. D. Forney, Jr., "The Viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
- [13] M. Aubry *et al.* (2014, Sep.) Piwik, an open analytics platform. [Online]. Available: <http://www.piwik.org>