World Academy of Science, Engineering and Technology
International Journal of Humanities and Social Sciences
Vol:6, No:5, 2012

E-Learning Platform with SPICE web service

A. Braeken, L. Sterckx, A. Touhafi, Y. Verbelen

***Abstract*—**When studying electronics, hands-on experience is considered to be very valuable for a better understanding of the concepts of electricity and electronics. Students lacking sufficient time in the lab are often put at disadvantage. A way to overcome this, is by using interactive multimedia in a virtual environment. Instead of proposing another new ad-hoc simulator for e-learning, we propose in this paper an e-learning platform integrating the SPICE simulator as a web service. This enables to make use of all the functions of the de-facto standard simulator SPICE inelectronics when developing new simulations**.**

***Keywords*—**E-learning, SPICE, Virtual experiments, Webservice

I. Introduction

UNDERSTANDING the behavior and interactions of fundamental variables like voltage and current in inductors and capacitors of electrical circuits requires thorough insight. In practice, many students have difficulties with it, since these concepts are not directly observable. They should be measured with meters and oscilloscopes. Animations and simulations provide the perfect way to demonstrate the behavior of power electronic circuits. When these animations and simulations are available by means of a global medium, like for instance the Internet, a student can study at his own pace, independent of time and place. In the past, several systems offering simulations and animations in the area of electronics have been successfully proposed [1]–[6]. General remarks for most of the existing systems are that the multimedia is often ad-hoc generated, stand-alone as closed applications, only applicable for power electronics, and interactivity is implemented by means of Flash or Java. In this paper, we develop a learning platform, which integrates a web service for the SPICE simulator (Simulation Program with Integrated Circuit Emphasis). SPICE is considered to be the de-facto standard for computer animations of electronic circuits. By offering the SPICE simulator as a web service, other software developers can focus on the implementation of improved graphical user interfaces. The paper is organized as follows. First, we explain previous work in this area. Then, the different aspects for introducing SPICE into a web context are described. In the next section, we discuss the learning platform in which the SPICE web service is integrated. Finally, we end with some conclusions and future work.

A. Braeken, L. Sterckx, A. Touhafi, and Y. Verbelen are with the Erasmushogeschool Brussel, Department Industrial Engineering and Science, Nijverheidskaai 170, 1070 Anderlecht, Belgium (phone: 0032 2 5590253; fax: 0032 2 5207123; e-mail: {An.braeken,, Abdellah.touhafi, Lucas.sterckx, Yannick.verbelen}@ehb.be).

II. Previous work

The original SPICE simulator was designed by L. Nagel and D.O. Peterson, during the late sixties. The program evolved to the standard for analog and mixed-mode circuit simulation. As an early open source program, SPICE was widely distributed and used in almost all large universities of Northern America. Due to its open source license, wide ranges of versions were adapted, each with their own specific application or updates. Nowadays, many off-line simulation programs for academia and industry, and also commercial products are developed based on SPICE. Compiled SPICE code can be used as a command line or as text-based simulation program. In order to make it more attractive, many commercial and open source software packages were developed to offer a GUI (graphical user interface). Such interfaces offer a more effective interaction between code and visualization. Two examples of commercial SPICE software with extensive GUIs are LTSpice and Circuitlogix.

In order to use the SPICE simulator in an e-learning situation, the Internet needs to fulfill the role of GUI for the simulator. Therefore, communication should take place between the computer of the student (client) and the server hosting the website. In the past, two implementations of such interfaces were developed ([7] and [8]). In both implementations, the SPICE code runs on the network server. The user sends requests for simulation to the server, which in turn responds with the results or graphs in the form of a web page. The compiled form of SPICE is addressed by means of network tools such as CGI [8] and Perl script [7]. Thanks to this client-server interaction, users do not need to install extra software or plug-ins, and platform independency is obtained. However, both websites ([7], [8]) were proposed almost 10 years ago. Especially in the Internet area, this is a very long period in which many new technologies were developed and existing ones improved. The implementation methods from [7] and [8] are also closed and offer no broader communication outside the website on which they were developed. Moreover, the input files of both websites are too extended and do not offer the fast interaction that can be obtained with the off-line software.

III. SPICE in web context

First we explain how a SOA (service oriented architecture) can solve the problem of reusability. Then, we describe how web services can be used to implement a SOA. Next some details are given on the implementation of SPICE as a web service. Finally, test results of the SPICE web service are discussed.

A. Service Oriented Architecture (SOA)

Following the principles of a client-server model, there exists one closed series of sequential processes to visualize a

graph or data in the web browser. The user can only execute simulations by means of the methods imposed by the site. This implies that for an outsider to implement a new application based on SPICE simulations, (s)he has to rewrite the code for obtaining interaction between the web and the simulator. Such problem of reusability can be easily solved by means of a 'new' principle in software design, SOA. In a SOA, a piece of software is considered as a system that offers a certain service to a user through a public interface. More formally (following [9]), a service is generally implemented as a course-grained, discoverable software entity that exists as a single instance and interacts with applications and other services through a loosely coupled (often asynchronous), message-based communication model. The philosophy behind SOA is not recent, but it took some time for technologies to support this idea. The first research on SOA was reported by E. Erl [10] of SOA Systems Inc.

B. Web services

A technology to implement a SOA principle into the network is a web service. Following the definition of the World Wide Web Consortium, a web service is a software system to support inter-machine communication through a network. The interface is described by a machine interpretable text format (WSDL Web Description). It defines the rules for the communication with the web service. Other systems communicate with a web service by means of SOAP (Simple Object Acces Protocol) messages, using HTTP (Hypertext Transfer Protocol) with serialization through XML (eXtensible Metadata Language). SOAP is a protocol that determines the structure of the exchanged information. Finally, services can profile themselves through UDDI (Universal Description Discovery and Integration). UDDI centralizes all web services into a register such that their description and address can be released. Fig. 1 shows the main principles of a web service.

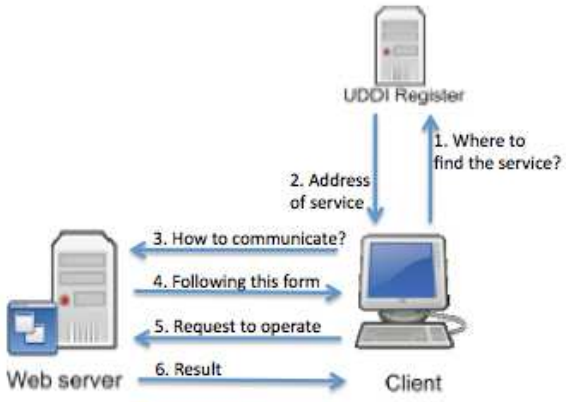

Fig. 1 Protocol overview in a webservice

Note that a web service has two additional advantages over other technologies that support communications over the network (COBRA, RMI,...). Thanks to the use of XML, web services are independent of the language on the platform. Consequently, it is perfectly possible that a program with the client is programmed in C++ running on a windows platform, while the web service is a Java program running on a Linux system. Secondly, web services using HTTP to transfer messages do not suffer interferences of proxies and firewalls. However, a disadvantage of web services is the loss of efficiency due to the overhead added by XML to the transmitted data. On the other hand, thanks to the actual bandwidths, this overhead is still acceptable.

In our application, web services are built with the API (Application Programming Interface) of the .NET Framework of Microsoft, called WCF (Windows Communication Foundation). WCF can be considered as the "Mercedes" for the development of web services, since all components and functionalities of the web services are largely configurable by the programmer and thus reaching maximum availability for the different clients. The language C# is used for programming the web service.

C. SPICE as web service

In order to address the methods of the simulator in the .NET environment, there are two possibilities. Firstly, the C code of SPICE could be rewritten into C#. Secondly, the SPICE code could be compiled into a DLL (Dynamic Link Library), a library of functions that can be addressed by means of C#. Both options require a thorough reconstruction of the SPICE code. Due to the enormous size of the code, this work would take an immense amount of time. The focus of our work is not on the optimization of the underlying software, but on the visualization of the result. Therefore, we address the complete SPICE simulator as a compiled executable. The disadvantage of this method is the increased latency that will be caused with increased parallel access of multiple clients.

The implemented web service contains one method and one argument. Clients can submit a netlist that will be simulated by the web service. Note that netlists are the representations of circuits in SPICE. A netlist consists of a list of all components together with their parameters and indexed connection points.

Microsoft offers in its Visual Studio, the integrated development environment for the .NET

Framework, A very simple interaction with other web services.The web service can be added to the library of functions by simply including the location of the WSDL file. In this way, the functions and data types can be immediately addressed.

D. Test results

The computer, used for the tests, has 2 processors with each a clock frequency of 2 GHz and a work memory of 4 GB. We simulate a simple circuit with an opamp as differentiator. By analysis of the traffic over the network, the time and size of the SOAP messages can be retrieved. The results of the test are described in Table I.

TABLE I
TEST RESULTS OF THE SPICE SERVICE

Size of the netlist	297 bytes
Size of raw data	36 kB
Time between reception of SOAP request and transmission of data	0.48 s
Size of SOAP response	56 kB
Total time for data-plot	1.56 s

From the test results, we can conclude that the SOAP protocol contributes to an overhead of almost 50% of the original data size. The size of the simulation data is strongly dependent on the time frame and the number of components in the analysis. The services have been tested with simulations consisting of maximum 200 kB processing data. We here assume that simulations for educational purposes are rarely more complex and heavy. Consequently, the processing will always be in the range of tenths of seconds for one simple request. Afterwards, the speed of the execution is dependent of the network and of the visualization process of the recipient. With the actual bandwidths and computer systems, one can expect that a simulation will last no longer than several seconds in order to be transmitted, processed, and executed. This time span is clearly below the physiological bound of 10-12 seconds to prevent perception as annoying [11].

IV. LEARNING PLATFORM

A. Website

Up to now, only the tools and methods are discussed to bring visualizations of simulations to a web context. Now we explain the implementation of the learning platform. The website will serve as a test platform for new multimedia and promotes the further implementation of the visualizations. The site can be found on http://iwt2.ehb.be/ElektroSims. Currently, the learning platform contains besides the SPICE web service, also two other existing systems for e-learning.

The first system is called videomodels and was developed by Guening [12]. Videomodels are video clips that offer a three-dimensional view of a working circuit by using the voltage across the circuit elements as a measure for their altitude in the three-dimensional space. Due to its three-dimensional representation, it is a very attractive model that has proven its efficiency in motivating students. The other method is an existing Java-applet created by Falstad [13]. This system contains a very high level of user interactivity by offering the opportunity to fully design the circuit. Instead, it is less attractive, it uses a two-dimensional approach for the display of voltage and current.

The actual learning platform also contains educational content with corresponding questions in each of the three e-learning systems for four basic circuits:

- The laws of Kirchoff
- A bridge rectifier
- A forward control of an NPN transistor
- An amplifier in a common emitter configuration with biasing

The main reason for the development of the website was to test the efficiency and perception of these three different methods. However, this evaluation is considered to be out of scope in this paper.

B. Technologies

As web technology, we mainly use ASP.NET (Active Server Page in .NET). Just like WCF, ASP.NET represents an extension of the .NET Framework. It enables programmers to develop web application that use .NET classes. ASP.NET is also programmed with one of the .NET languages, e.g. C# or Visual Basic.NET. ASP.NET is a technology that is applied by the server. Consequently, the user does not need to download extra data on his system, besides the HTML code. Note that the application is highly portable due to the existence of .NET implementations on most server platforms.

We have implemented the system with a strong separation between content and view. Therefore, the technology XML is used since it lowers the production time and efforts to add new content. When the content is extended with XML, the document can be interpreted by other systems by means of a XLST (eXtensible Markup Language) processor. XLST is a language based on XML, used to transform XML documents into other XML documents, web pages, or PDF documents. All style elements are declared in one or more XLST files. The transformation can be performed on client or server side.

C. SPICE simulator

In the context of this paper, the main aim was to bring the SPICE simulator to the web context, such that other programmers in their simulation applications can easily adopt it. Therefore, we only developed a basic GUI for it.

The most important part of the SPICE software is the ability to change the different parameters of the circuits and to plot the output in a graph. A simple implementation for offering these functionalities can be realized by showing an image of the circuit with indications, corresponding to nodes in the netlist. Together with the image, the student can complete a form, containing all the components in the circuit. By activating a button, all parameters are written into the netlist. This information is then sent to the SPICE web service and is simulated. The obtained data can be interpreted by the server side as programming language or visualized by a data visualization software.

Consequently to offer the SPICE simulations, not only the transformation of an XML document is required. Also a program on the server side should be written to provide the exchange of simulation data. The XML document will contain all the information such that the server is able to create a form with all parameters of the netlist and from which a new netlist can be created. All text of the netlist should be contained in the XML representation.

Fig. 2 shows the GUI of the SPICE simulator on our website. All components of the circuit with corresponding value are shown in the right corner of the site. Some values can be changed. When the student clicks on the Plot-button, an empty graph is shown. The graph is created as soon as the parameters are selected in the checkbox list. This stimulates the trial-and-error feeling for the students.

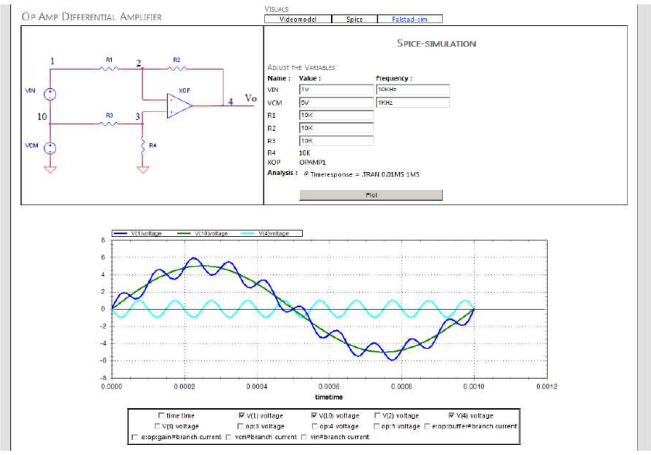

Fig. 2 SPICE simulation on the website

V. Conclusions and Future Work

We have presented the different techniques and technologies that are used for creating an e-learning platform with SPICE as a web service. The focus in the design of the learning platform was on expandability and modularization of the educational content. By offering the SPICE simulator as a web service, software developers can now focus on the implementation of improved graphical user interfaces.

There are several directions for future work. Firstly, the software can be offered as a standalone package and further extended and graphically improved. Secondly, the software can be standardized following the SCORM (Sharable Content Object Reference Model) norms. This would enable the immediate integration to an existing LMS (Learning Management System).

Acknowledgment

This paper is a result of the project ”Bilingual learning environment for 3D visualizations of current and voltage in electric and electronic circuits”, sponsored by the Prins Filip Fonds in Belgium.

References

[1] J. Hospodka, and J. Bicak,, Web-Based Application for Electric Circuit Analysis, *Fourth International Multi-Conference on Computing in the Global Information Technology 2009*, pp. 157–160.

[2] S. C. Wang and Y. H. Liu, Software-Reconfigurable E-Learning Platform for Power Electronics Courses, *IEEE Transactions on Industrial Electronics 2008*, vol. 55, nr. 6, pp. 2416–2424 .

[3] U. Drofenik and J. Kolar, Interactive Power Electronics Seminar (iIPES)- a Web-Based Introductory Power Electronics Course Employing Java-Applets, *IEEE 33rd Annual Power Electronics Specialists Conference 2002*, vol. 2, pp. 443448.

[4] J. Hamar, H. Funato, S. Ogasawara, O. Dranga, and C.K. Tse, Multimedia Based E-Learning Tools for Dynamic Modeling, *IEEE International Conference on Industrial Technology 2005*, pp. 07803.

[5] L. Siragusa, , Quality E-Learning: Designing Pedagogically Effective Web Based Environments for Enhancing Student Online Learning in Higher Education, *Proceedings Western Australian Institute for Educational Research Forum*, 2006.

[6] F. Alonso, G. Lopez, D. Manrique, and J.M. Vies, An Instructional Model for Web- Based E-Learning Education with a Blended Learning Process Approach, *British Journal of Educational Technology*, 2005, vol. 36, nr. 2, pp. 217235.

[7] B. Wilamowski, A. Malinowski, and J. Regnier, Internet as a New Graphical User Interface for the SPICE Circuit Simulator, *IEEE Transactions on Industrial Electronics*, 2001, vol. 48, nr. 6, pp. 1266–1268.

[8] B. Swiercz, L, Starzak, M., Zubert, and A. Napieralski, , An Interactive Website as a Tool for CAD of Power Circuits, Tech. Proc. 2003 Nanotech. *Conf. Trade Show 2003*, pp. 346349.

[9] A. Brown, A. Johnston, K. Kelly, Using Service-Oriented Architecture and Component-Based Development to Build Web Service Applications.

[10] T. Erl, , Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, augustus, 2005.

[11] C. H. Muntean, , Fine Grained Content-Based Adaptation Mechanism for Providing High End-User Quality of Experience with Adaptive Hypermedia Systems, *Proceedings of the 15 International World Wide Web Conference*, 2006.

[12] F. Gueuning, Circuits Electroniques Visualises par Videomodeles, *J3eA, Journal sur l’Enseignement des Sciences et Technologies de l’Information*, 2005, vol. 4, nr. 2, pp. 8.

[13] P. Falstad, Electronic Circuit simulator, http://www.falstad.com/circuit/