Simulation eLearning environment for digital circuits test and diagnosis

P. Radoyska\textsuperscript{1}, T.Ivanova\textsuperscript{2}, N.Spasova\textsuperscript{3}

\textsuperscript{1,2,3}Technical University of Sofia

Key words: distributed learning, digital circuits, test and diagnostics, Rich Web Client (RWC), Rich Internet Applications (RIA)

Abstract:

In this paper we present the first step of our project for building a distributed learning environment for studying digital circuits Test and Diagnostics at TU-Sofia. We describe the main requirements for this environment, substantiate the developer platform choice and present our simulation and circuit parameter calculation tools.

We choose the Eclipse Rich Ajax Platform regarded the following main reasons: it is Java based, which makes it compatible with the most of the open source e-learning platforms; it has powerful interaction and drawing functionality; it permits to use well developed Java Development tools.

1. Introduction

The experiments are significant part of learning and research activities. The laboratories are required for the experiments. Conventional hands-on laboratories have rich tradition but they are not enough for the contemporary needs. The self study and long life study become more and more important in the 21 century. Free experiment planning and performing are part of a self study process. Innovative projects implementation become available for small groups or even individuals due to the technology progress. Web-based technologies are prerequisites for collaboration of the researchers from the all sides of the world. Virtual labs are the tool, which are based on web-technologies and allow remote and/or simulation experiment performance.

Virtual labs are the tool, which are based on web-technologies and allow remote and/or simulation experiment performance. They break the geographical, the time and the price constrains. The Laboratory types can be classified in 3 categories [1]:

- **Hands-on laboratory** – all the equipment required to perform the laboratory is physically set up and the person who perform the experiment are physically present in the lab.
- **Remote laboratory** – all the equipment required to perform the laboratory is physically set up and the person who perform the experiment are geographically remote and get access by the web.
- **Simulation laboratory** – a software-based environment that is built as an imitation of real experiments. The all infrastructure, required for laboratories is not real, but simulated on computers.

The simulation labs offer significant advantages. They are cheapest according to the hardware equipment and extra resources, necessary for the experiment performing. They have not access time restrictions. This kind of labs allows full possibilities for control the parameters of the experiment and unbounded number of repeating the experiment. They are preferably for dangerous experiment performance. The simulation labs allow observation the internal processes, which are not observable in real experiments. For example the signal level of the
VLSIs internal lines. The simulation labs are useful for the project verification at the design time. In the digital circuits test and diagnosis aspect simulation labs are very useful for the testability check and for generation the optimal test and diagnosis input vectors sets. In this paper we offered the web-based environment for simulated experiments in area of digital circuits test and diagnosis. The rest of the paper is organized as follows. In Section 2 are discussed the related works. In Section 3, are outlined the main project objectives and requirements and draw the conceptual model. In Section 4, are discussed the selected software platform. In Section 5, publishing as Semantic Web Service is discussed.

2. Related Works

In the world there is a wide interest in learning digital circuits test and diagnostic. Such kinds of courses are offered in many universities. In many of them the experiments are performed on powerful PC-based simulation environments (PSPICE, Xilinx). We have found free Web-based logic simulators, used for the digital circuits e-learning: LogiFlash [2], WELD [3], Digital Works [4], DLSim 3 [5], Breadboard simulator [6], Turbo Tester (TT) [7] [8]. Their key features are compared in table 1. The examined simulators are well done and have build in behavior simulators. But only TT is adequate to the test and diagnosis purposes. Only TT has build in fault simulators. Turbo Tester has been developed on the European project REASON (REsearch And Training Action for System On Chip DesigN) by the Tallinn Technical University, Estonia, Technical University of Ilmenau, Germany, Warsaw University of Technology, Poland, Institute of Informatics of the Slovak Academy of Science, and others. Turbo Tester is PC-based tool set, which consists of test generators based on various algorithms, logic and fault simulators, a test optimizer, a module for hazard analysis, a simulator and test generator for defects, built-in self-test simulators, design verification and design error diagnosis tools. Turbo Tester is based on Java applets.

<table>
<thead>
<tr>
<th>Simulator</th>
<th>Web-access</th>
<th>Platform</th>
<th>Input</th>
<th>Simulation type</th>
<th>Free experiment</th>
<th>Fault simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogiFlash</td>
<td>Yes</td>
<td>Macromedia Flash</td>
<td>Interactive graphics</td>
<td>Circuit behavior</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Digital Works</td>
<td>No</td>
<td>Java</td>
<td>Interactive graphics</td>
<td>Circuit behavior</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DLSim 3</td>
<td>Yes</td>
<td>Java plug-ins</td>
<td>Interactive graphics</td>
<td>Circuit behavior</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>WELD</td>
<td>Yes</td>
<td>Java applet</td>
<td>VHDL</td>
<td>Circuit behavior</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Breadboard simulator</td>
<td>Yes</td>
<td>Java applet</td>
<td>VHDL</td>
<td>Circuit behavior</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Turbo Tester</td>
<td>Yes</td>
<td>Java applet</td>
<td>VHDL</td>
<td>Fault and fault-free circuit behavior</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1

Establishing our private simulating environment we will get possibilities to implement all demos and algorithms, needed for education and make experiments with new developed algorithms.

3. Conceptual model

The main task of the e-learning fault simulation environment is to help student to understand fault modeling, fault simulation algorithms, test and diagnosis input vector set
mechanisms. The main components of the environment are described in fig.1. Some of the components are specified in fig.2, 3, 4, 5 and 6. A part of components are implemented and build in the environment. The rest, which are written with italic symbols, are under implementation.

![Diagram of e-learning environment](image)

**Fig. 1. Conceptual model of e-learning environment**

![Diagram of Administrative tools](image)

**Fig. 2. Conceptual model of Administrative tools**

![Diagram of Simulation tools](image)

**Fig. 3. Conceptual model of Simulation tools**

The diagram of simulation process is presented on fig.7. The circuits’ descriptions are stored in database in any acceptable format. Parser reads the circuit description and converts it to the circuit model. Any of the embedded algorithms can be performed on the circuit model and results will be presented on the screen.

The data flow permissions for all persons, using learning environment, are presented on fig. 8. All learning resources are divided into groups: the main resources and the private resources. The access to them is restricted based on the roles and belongings. Only instructor has permission to use and manipulate learning items in the main resource group. The students have only permission to use these items without any changes. Every student has own resource space and can upload, use and manipulate private resources. He can share these resources with other students. The full access to all resources has only administrator. This resource organization will give more learning flexibility especially for students’ with kinaesthetic learning style.
4. Software platform determination

For our project we decide to stand on the Rich Web Client ideology [9]. Rich Internet Applications (RIAs) or Rich Web Client strive to leverage the Web with the engaging interactivity found in traditional desktop applications. Combined with the Software-as-a-Service (SaaS) delivery model, web applications are empowered to compete with desktop interfaces. Moreover, RIAs provide a new client-server architecture that reduces significantly the network traffic using more intelligent asynchronous requests. Faster performance, readiness and engaging interactivity are the hallmarks of this new crop of applications.

For the research platform we choose the Eclipse Rich Ajax Platform (RAP) [10], which is the open source platform. It enables us to build rich, Ajax-enabled Web applications by using the Eclipse development model, plug-ins with the well known Eclipse workbench extension points and a widget toolkit with SWT API (plus JFace). We choose RAP because it is very similar to Eclipse RCP and allowed us to build in simulation tools, already created for experimental purposes. In addition RAP is running on a server and clients can access the application with standard browsers. And the last reason is that RAP takes advantage of the award winning Java Development tools and the Plug-in Development tools provided by Eclipse.org.
5. Publication as a Semantic Web service

We plan to publish the functionalities of our application as web service to make them easily searchable and usable from the internet. Web services [17] are application programs (called web APIs) that are accessed via HTTP and executed on a remote system hosting the requested services. Web Services use Simple Object Access Protocol (SOAP [19],[22]) and Extensible Markup Language (XML [23]) syntaxes to exchange messages. There is often a machine-readable description of the operations offered by the web service, written in the Web Services Description Language (WSDL [18]). There are several problems in the plain Web Services approach: Insufficient coverage of service properties; free-form terminology in service descriptions, leading to ambiguities in property descriptions; static binding to predetermined services. The main approach to cope with these problems is adding semantic to Web service technologies. Augmentation of Web Service and Semantic Web technologies leads to establishing the Semantic Web Service recommendation standards (see fig. 9).

As our Simulation e-Learning environment propose rich feature set, and there are several web-based related simulation environments, we believe that describing and publishing it as semantic web service will be the best way for finding and usage its services and algorithms. That is why we analyze and compare Semantic Web service (SWS) technologies for choosing the best one for our application.

We have found three large projects for Semantic Web Service representation frameworks: OWL-S[24], WSMO[21], and SWSF[12] (Table 2).

<table>
<thead>
<tr>
<th>SWS framework</th>
<th>Ontology Language</th>
<th>Logical ground</th>
<th>Service composition, planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWL-S</td>
<td>OWL, RDF(S)[16]</td>
<td>DL</td>
<td>Static, HTN planning [25]</td>
</tr>
<tr>
<td>WSMO</td>
<td>WSML-Rule, WSML-Core, WSML-DL, WSML-Full</td>
<td>DL +Rules</td>
<td>Dynamic, Abstract State Machines (ASMs), Concurrent Transaction Logic. [26]</td>
</tr>
<tr>
<td>SWSF</td>
<td>SWSL-FOL, SWSL-Rules</td>
<td>first-order logic (FOL), situation calculus</td>
<td>Process Specification Language (PSL)</td>
</tr>
</tbody>
</table>

OWL-S[11] (Web Ontology Language for Services) is a ontology-representation language, developed to provide a set of ontologies to describe services using the Web Ontology Language (OWL[15]). OWL-S combines the representational technologies of the Semantic Web (RDF and OWL) with the dominant Web services standards, such as WSDL[20], UDDI and SOAP. The rich semantic service descriptions allowed by OWL (Ontology Web Language) are grounded to WSDL-based messages for service invocations. OWL-S is an OWL ontology, including three subontologies[24]: Profile subontology, specifying what a service does; Process model subontology, describing how does a service work, and Grounding specifying how to use a service by mapping the constructs of the process model onto concrete message formats and protocols typically expressed in WSDL. A service provider should create a service profile for advertisement and a service requester should create his own profile.
to explain what services he searches for. The best service is found through semantic mapping between these two profiles. The process model will be the basis of interaction among the requester and provider during the service usage. The process is characterized by its inputs, outputs, preconditions, and effects. The input is a set of parameters that the client needs to provide for the service provider. The output is a set of variables for which the service provides values. Both the input and output variables are identified by URI:s that need to be defined in an OWL ontology. The preconditions are requirements for the words state before the service can be executed and the effects are the changes in the words state that the service does when it is executed. One drawback of OWL-S is that it does not contain variables that are often required to specify planning operators.

WSMO (Web Service Modeling Ontology) [13], [14] is an ontology, being used in Web Services Modeling Framework (WSMF). WSMF includes ontologies that define the shared terminology, goals that define the space of problems to be solved (request types), Web services that define the elements of the solutions (services), and mediators to solve the interaction problems between the other elements. The main part of the web services descriptions in WSMO constitutes of preconditions, postconditions, assumptions and effects, which correspond to the inputs, outputs, preconditions and effects in OWL-S respectively. The main characteristic of WSMF is the aim at maximum de-coupling between its components. Another strength of WSML is that it has five different logical levels, based on the stacking principle: the most basic one is WSML-Core that is defined as an intersection of logic programming and description logic, WSML-Flight is a language that extends WSML-Core with non-monotonic reasoning, WSML-Rule is the extension of WSML-Flight into a logic programming language, and WSML-DL extends WSML-Core to description logic (comparable to OWL-DL).

Semantic Web Services Framework (SWSF) [12] attempts to build a more comprehensive framework by defining a larger set of concepts over OWL-S. The main parts of SWSF are Semantic Web Services Language (SWSL) [13] and Semantic Web Services Ontology (SWSO) [14]. The essential novelties of SWSF over OWL-S are usage of the first-order language and more comprehensive process model. SWSO contains a conceptual model of a Web Service and an axiomatization of that model.

SWS technology is an emergent research realm. Best of our knowledge, there are no real-world working applications in e-learning, based on Semantic Web Service architecture. We believe the SWS is appropriate for describing and publishing our learning environment capabilities because of SWS propose a rich model for detailed semantic description of system (and its subsystems) properties and logical model for reasoning in the process of finding and choosing the most appropriate tool or algorithm for given learner or task. We discuss the method of service publication. We may publish it as a whole application, including all the functionalities (demo tools, service tools, simulation tools, testing algorithms, etc.), or publish its main components as relatively independent services. Our application is modular and it will be easy to make semantic description, independent interfaces and propose its main modules as separate services. The main strength of the second approach is its flexibility. It would make possible building personal learner’s assistant-recommender for dynamic personalized choosing the most appropriate for concrete learner sequence of learning and simulation resources usage. For example, if the learner is new for the system, the assistant will recommend him first to see the demo, but if advanced learner makes a request for performing specific test, recommender should propose him usage of appropriate for his task algorithms, sequenced in the best order. Learners will discover and use not only the whole application, but needed its parts and semantic web service technology can propose flexible easy to use interfaces and possibility for quick and easy introduction to the environment and preliminary evaluation of usefulness for concrete task without registration and reading long documentation.
Comparing the three above explored SWS approaches, we conclude that WSMO is the most appropriate for our task, as it has layered logical complexity and goal-oriented architecture. This make possible using learning goals as goals for service-composition in the process of dynamic choosing the needed order of service invocation for specific learner or learning goal.

6. Conclusion

In this paper is offered one project for a web-based environment for simulated experiments in area of digital circuits test and diagnosis. This simulating environment provides possibilities for: (1) implementation the demonstrations and algorithms, needed for education and (2) providing the experiments with new developed algorithms. Based on the main project objectives and requirements is drawn the conceptual model of the simulation environment. A developing software platform and a Semantic Web Service representation framework are chosen. The presented simulation environment is used in several learning courses, offered in TU-Sofia. The students can upload, download and delete custom digital circuits, described in ISCAS-85 format. The students use the simulation environment for the next tasks: testability calculations; fault dictionary generation and fault collapse ratio observation; Z, A and AB sets calculations; test sets generation using ATPG (Automatic Test Pattern Generator), based on pseudo random test vectors and user defined fault coverage ratio; test sets generation with 100% fault coverage based on deterministic approach; test sets generation for bridge faults; diagnosis set generation.

References


Authors:

Pavlinka Radoyska, Technical University – Sofia, College of Energy and Electronics, 31 Blvd. Bulgaria, 2140 Botevgrad, Bulgaria, email: pradoiska@abv.bg

Tatyana Ivanova, Dr.,Technical University – Sofia, College of Energy and Electronics, 31 Blvd. Bulgaria, 2140 Botevgrad, Bulgaria, email: ivanova_tatiana@abv.bg

Nadejda Spasova, Technical University – Sofia, College of Energy and Electronics, 31 Blvd. Bulgaria, 2140 Botevgrad, Bulgaria, email: nadia.spasova@mail.bg