Simple Scalar as an eLearning Mashup web application (eScalar)

Aiman Abu Samra[1] and Adwan Abdelfattah [2]
[1] Computer Engineering Department, The Islamic University of Gaza, Palestine
[2] Computer Science Department, The Arab American University of Jenin, Palestine

Abstract: *In recent years Mashups are increasingly found in the eLearning field to offer instructors the opportunity of linking students to multiple sources of information as well as offering students the ability of accessing learning tools and services in an easy manner. In this paper we describe the design and implementation of our Mashup web application that offer the SimpleScalar simulator as an eLearning web-based service for the students of Computer Architecture course. This Mashup overcomes the problem of platform and hardware dependency of SimpleScalar and can be accessed from anywhere. We added new service to SimpleScalar in our Mashup web application that is generating visual statistical results.*

Keywords: *SimpleScalar, eLearning, Web 2.0, Web Application Mashup.*

Received April 27, 2009; Accepted February 17, 2010

1. Introduction

The traditional learning methodologies revealed many limitations in different fields, especially when that related to teachers and students existing in distant locations, the world rapid development in international information and teaching exchange pushed strongly toward distant learning, and here, eLearning services are good solutions. Applying distance learning solution does not prevent the students training on simulators or any application that is used in campus-based education [11]. An eLearning system is a group of functions, elements and components mixed to give a good and robust learning experience, eLearning systems form a huge agglomerative class room where students are able to discuss issues, share opinions, share media and share anything is kept logged, this means nothing is forgotten or lost [6, 7].

Another aspect of Web 2.0 concepts interesting in the context of E-Learning are Web Sharing Applications [2].

Usually cooperative work reaches better results because it places bigger potential on a defined case, eLearning gives this opportunity by offering place, management tools, and a big library of resources[4,6]. Simulators are used in the learning process, many simulators were produced before the concept of e-Learning was revealed, and thus these simulators are used solely by students and it is relatively hard to share results and to get benefits from others experiences, so we looked for combining a simulator with eLearning features to get a powerful eLearning system. Especially, when we give students the ability to use simulators online, and can share results and troubleshoot errors with no limitations on platform or on hardware dependency. In this paper we are talking about Simple Scalar simulator [8] that's very vital in Computer Architecture field and apply the techniques of eLearning to it. Besides simple scalar we also include the benchmark SPEC CPU2000 [9].

We will at first give a background about the concepts and technologies used, then we will talk about the simulator and its specifications, then we introduce our system and the steps of implementation. Finally we mention the future work and our vision of this system.

2. Mashups in eLearning

Mashups are applications that "combine multiple sets of data streams into a unified user experience". They access open APIs and data sources to produce a new web service or tool that was not originally provided by either source. Mashup web application has two parts: the new service delivered through web page and the combined data made available through API and other protocols such as HTTP, REST and SOAP [5]. Client accesses mashups through web browser displaying a web page containing the result of the service.

Mashups are found in many different fields, e.g. enterprise systems, news, and shopping. Recent development in eLearning is the ability to create new and powerful elearning tools for courses using mashups. They found their place in eLearning because they are:

- Alive: They are current, up to date, and so consistently relevant. For instance, live election mashups show you how an election is changing as those changes happen.

- Interactive: Mashups can immerse you in the educational experience. Google Earth, for example, provides the ability to traverse ancient Rome, visually follow the growth of the world's population or examine more than 50 years of U.S. Geographical Survey data.
- Engaging: Because they are always relevant and immersive, and because they bring students into the learning process, mashups have the potential to be highly engaging.

3. Simple Scalar Simulator

SimpleScalar LLC is a privately held corporation, founded by Todd Austin. The version used is revision 3.0d**.**

3.1. Simple Scalar tools set

The SimpleScalar tools set is a system software infrastructure used to build modeling applications for program performance analysis, detailed micro-architectural modeling, and hardware-software co-verification. Using the SimpleScalar tools, users can build modeling applications that simulate real programs running on a range of modern processors and systems [8]. The tools set includes sample simulators ranging from a fast functional simulator to a detailed, dynamically scheduled processor model that supports non-blocking caches, speculative execution, and state-of-the-art branch prediction. The SimpleScalar tools are used widely for research and instruction, for example, in 2000 more than one third of all papers published in top computer architecture conferences used the SimpleScalar tools to evaluate their designs. In addition to simulators, the SimpleScalar tool set includes performance visualization tools, statistical analysis resources, and debug and verification infrastructure.

SimpleScalar LLC is unique in the EDA community. This software is distributed as an open source model, trusting that the users will license the software after that they use it. This type of distribution gives the users the ability to extend SimpleScalar, and to adapt existing models to their own ideas.

3.2. Simple Scalar instruction sets emulation

SimpleScalar simulators can emulate the Alpha, PISA, ARM, and x86 instruction sets. The tool set includes a machine definition infrastructure that permits most architectural details to be separated from simulator implementations. All of the simulators distributed with the current release of SimpleScalar can run programs from any of the above listed instruction sets. Complex instruction set emulation (e.g., x86) can be implemented with or without microcode, making the SimpleScalar tools particularly useful for modeling CISC instruction sets.

The PISA instruction set (a.k.a. the portable instruction set architecture) is a simple MIPS-like instruction set maintained primarily for instructional use. A GNU GCC-based cross-compiler and pre-built libraries are also available for this target. The PISA target is particularly useful for computer engineering instruction as the tools can be built on a wide range of host platforms, including Linux/x86, Win2000, SPARC Solaris, and others.

3.3. Simple Scalar Running platforms

SimpleScalar builds on most 32-bit and 64-bit flavors of UNIX and Windows NT-based operating systems. The internal software architecture of the tool set includes a host interface module, permitting fast porting to other host platforms. The host interface module permits cross-endian emulation, thus it is possible to use emulate a target on a host platform with a different endian, e.g., running Alpha ISA emulation on a SPARC Solaris host platform. Most SimpleScalar users and developers (including SimpleScalar LLC) use SimpleScalar on Linux/x86. [8].

4. The Limitations in the Current Simple Scalar

Simple scalar is not the only simulator that does this job, however, it is the best one because of many factors, its flexibility and open-source license are of these factors, but unfortunately simple scalar is hardware and OS Dependent, and it is hard to install and use on some platforms, in addition to the results manipulating you can get the results are in a text file, and without any structure for the text, there are no analytical tools provided, and the most important limitation no GUI for this simulator.

This simulator Support a lot of functions but the lack of GUI makes it harder to discover all the features of this powerful simulator, and for sure makes it harder for it to be used, not all users are familiar with command line or shell scripting.

Simple scalar has the following limitations that make it sometimes difficult to use or deploy.

- Platform dependency: simple scalar installed over Linux operating system which is sometimes considered as a difficulty for students unfamiliar with Linux.
- Standalone software: simple scalar is installed on a single machine as a standalone so every student wants to use it has to have a Linux machine with Simple Scalar installed.
- Not cooperative: using the simulator is limited to the single user on his machine. There is no interaction or cooperation between students by distributing their simulation results and tests.

- No statistical information**:** simulator is limited to giving results for users without taking care of results history and data mining for previous tests.

5. Why EScalar should be a mashup

Because we need powerful tools that make eLearning simple, easy and comfortable to all students. Sometimes there are difficulties in configuring some tools due to lack of experience for some students or platform dependency problems as some tools may be dedicated to a specific operating system. Using *mashups* things can be changed, service are deployed once and provided as a service. Students then concentrate on the learning process rather than configuring and deploying the tool.

Mashups also make the learning process cooperative; results and statistics can be collected from many students for comparison purposes which enable them to create a complete community specified courses, a learning tools, or systems.

5.1. eScalar Description.

eScalar is a Mashup web application that offers an eLearning service for using the simple scalar simulator in addition to analysis tools, eScalar can be installed and used locally inside a university or can be deployed internationally and used by remote students from a number of universities.

eScalar offers an easy to use and comprehensive user interface for the simple scalar simulator, this revealed all the features of this simulator. The system also includes the benchmark SPECCPU2000. The SPEC CPU2000 benchmark suite is a collection of 26 compute-intensive, non-trivial programs used to evaluate the performance of a computer's CPU, memory system, and compilers. The benchmarks in this suite were chosen to represent real-world applications, and thus exhibit a wide range of runtime behaviors' [9]**.** The system also gave a fast way to set up a simple scalar testing lab, using one pc as a server and n- clients you can start using the simple scalar for n-clients, and with a robust queuing technique these n-clients can be mixed between remote and local clients. A special algorithm was designed to serve in queuing and giving priorities to clients to use the resources, this way, clients won't feel left behind.

Also a discussion thread can be open on any results to discuss, and chat about specific results, and the results information are archived and indexed for searching.

eScalar can be also used to post any problems regarding using the Simulator itself, and course labs can be held on eScalar, groups can be formed, and times can be arranged, authentication can be made, and the log of the lab lecture is sent to all the corresponding students.

6. eScalar characteristics

- Uses the architecture of Mashups, making it easier to develop and scale.
- Powerful internal architecture.
- New queuing technology that guarantees priority and execution in reasonable time.
- Simple and Rich user interface, designed as a RIA (HTML and AJAX) this technology requires no additional plug in for the web browser and runs on the native web browser.
- Can be used either by its user interface or as a web service to be integrated in other systems.
- Authentication is required and behaviors are logged.
- Discussion threads can be opened.
- Chat sessions can be held.
- Groups can be formed and lab lectures can be given.
- Analytical tools like graphs and statistics.
- Graphs can be bars, curves, or any other type of graph.

7. Deployment and Technical Background of eScalar

eScalar is a Mashup web application combines information from two data sources to create the new simulation service accessed from anywhere and it has the following main components:

- Simple Scalar instance installed on a Linux machine: This is the execution engine of simulation requests sent by students which runs on a Linux machine.
- Google charts API:Results generated by execution engine are sent to Google charts API to dynamically generate charts and statistics. eScalar will be deployed on Sun Glassfish Application Server as a web application accessed by web browsers.
- Model View Controller: (MVC) is an architectural pattern used in software engineering. Successful use of the pattern isolates business logic from user interface considerations, resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business rules without affecting the other [3]. In MVC, the model represents the information (the data) of the application; the view corresponds to elements of the user interface such as text, checkbox items, and so forth; and the controller manages the communication of data and the business rules used to manipulate the data to and from the model. System architecture is compliant to MVC design pattern as shown in the figure 1.

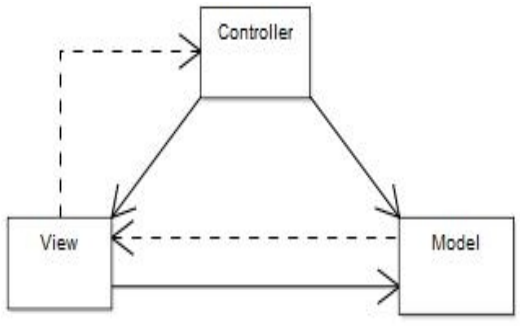

Figure 1. MVC design pattern.

Where

- Model: are the two data sources that create the new mashup service. Simple Scalar engine executes command sent by users on a linux machine and generates results that will be sent to Google API to generate charts and statistical data.
- Controller: is a part of system which has the responsibility of handling users requests and forwarding them to data sources as well as managing user queuing and access to the server.
- View: is the display of eScalar mashup. HTML pages view the result of simulation and the generated charts.

The architecture of the proposed eScalar System as shown in figure 2 is flexible and modular which makes the adaptation of this system to the future requirements very easy and without any changes to the core of the system. View can be easily modified without making any change in the controller or the data source (model). This architecture also allows the business logic on the controller to be changed and still interfacing with the same view without making any changes to them.

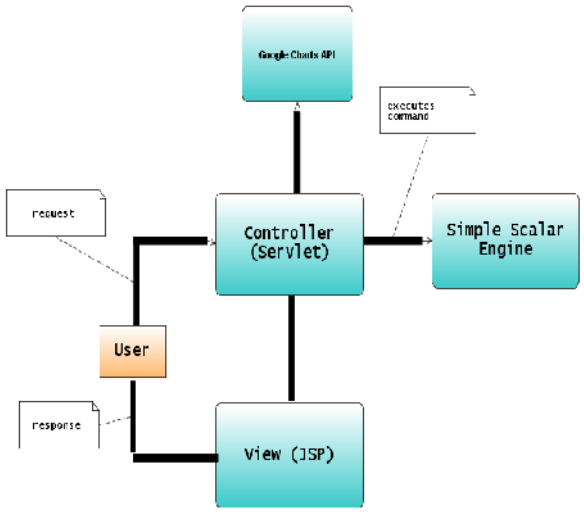

Figure 2: eScalar system architecture

8. Additional Modules

The architecture of eScalar as a Mashup opened the way of easily integrating and using other services, this way also made eScalar ready to integrate with other modules for analysis, visualization, or any other module, eScalar itself can be used inside a Mashup to create a new service that does a specific job.

Google graphs were used as a module in this Mashup, the system itself is designed to interface these services without diving into the implementation of the service itself.

9. Implementation

To implement the proposed EScalar System we have to use an Apache server as a web server to serve users. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation [1]. For accessing the simple scalar system which installed at the web server we have to used Java Servlet technology which similar to java applets [10]. Java servlets runs on the server side. So servlets are connected with simple scalar to execute the input data. They also connected with Google chart API to transform the results of the simulation to readable charts. To interact with users dynamically we have to use Java Server Pages (JSP) which is a server side language [10], to create our dynamically generated web pages. JSP is responsible for the response to user requests. The users of our system may use it remotely. The only application they need is the web browser.

10. The Future of eScalar

At the same time that eScalar is a service, it can be used in future as another data source for new *Machup* services to create new applications that were not planned to create. For eScalar to be scalable and serves the increasing number of users, cluster of machines can be deployed with a controller to organize requests sent by users and choosing the best machine to serve that request.

11. Conclusion

Our proposed eScalar system architecture is an effective way to make simulations available online for *Computer Architecture* course. eScalar could overcome some limitations in which students had to be involved in ,when they use current SimpleScalar simulator. These limitations like hardware and OS dependency, hardness to install, getting the results in text file only, the absence of analytical tools, and the absence of GUI. eScalar system architecture is compliant to MVC design pattern. It is also presented as a new *Mashup* e-Learning web service to participate

in Web 2.0 community and to be a potential data source to *Mashup* world.

References

[1] Apache Software Foundation, www.apache.org, 2009.

[2] Ebner M.,"E-Learning 2.0 = e-Learning 1.0 + Web 2.0", *The Second International Conference on Availability, Reliability and Security* (ARES'07), IEEE, pp. 1235-1239, 2007.

[3] Erich G., Richard H., Ralph J., John V., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional Computing Series, 2007.

[4] Lennon, J., Maurer, H., "Why it is Difficult to Introduce e-Learning into Schools And Some New Solutions", *Journal of Universal Computer Science* Vol. 9, no.10 , pp. 1244-1257, 2003.

[5] Maximilien E. M., Wilkinson H., Desai N., and Tai S., "A Domain Specific-Language for Web APIs and Services Mashups", *Proc. of 5th Intl Conf. on Service-Oriented Computing (ICSOC)*, LNCS 4749, pp. 13–26, Vienna, Austria, Sept, 2007.

[6] Neumann, G., "XoWiki-Towards a Generic Tool for Web 2.0 Applications and Social Software", *OpenACS and .LRN Conference* , Vienna, 2007.

[7] Safran, C., Helic, D., Gütl, C., "E-Learning practices and Web 2.0", *International Conference* on *Interactive Computer Aided Learning*, ICL2007, 2007.

[8] SimpleScalar, LLC. www.simplescalar.com, 2009.

[9] Standard Performance Evaluation Corporation http://www.spec.org/cpu2000/research/umn, 2009.

[10] Sun Developer network, www.java.sun.com /products/, 2009.

[11] Swedish National Agency for Higher Education, www.hsv.se/download/18.8f0e4c9119e2b4a60c800028057/0811R.pdf, 2009.

Aiman Abu Samra is an IEEE and Computer Society member. He received his PhD degree from the National Technical University of Ukraine in 1996. He is currently an assistant professor at the Computer Engineering Department at the Islamic University of Gaza. His Research interests include computer architecture, computer networks and software engineering. He managed several funded projects in cooperation with industry. He teaches several courses on computer architecture and computer networks.

Adwan AbdelFattah is an assistant professor at the Computer Science Department of the Arab American University of Jenin, Palestine. Previously he worked at Philadelphia and Zarka Private University. He received his PhD degree from the National Technical University of Ukraine in 1996. His research interests include computer networks, computer architecture, cryptography and networks security.