Integrated Web Interface for RESOLVE

Chuck Cook and Grant Giles
ctcook@clemson.edu, ggiles@clemson.edu
School of Computing, Clemson University
Clemson SC, 29634

Abstract—E**mergence of widespread broadband internet access and technologies capable of creating a rich and dynamic user experience provides a platform to introduce RESOLVE to a larger audience. These technologies can be utilized to offer a web browser-based environment for development and verification of REOLVE components. This integrated system is ideal for no-hassle RESOLVE development and use in the classroom environment.**

Index Terms— **Development Tools, Education, Software Engineering, Verification**

I. Introduction

One of the primary goals of the Clemson Reusable Software Research Group (RSRG) is to increase the use and recognition of the RESOLVE programming language. One of the tools in our arsenal is the open-source Clemson RESOLVE Verifying Compiler. This tool must, however, be installed locally, a tricky procedure that requires external libraries to function properly. We realized early on that in order for the RESOLVE and the Verifying Compiler to gain popularity and usage, something simpler would be required. The recent increasing acceptance of cloud computing and internet based applications seemed to present a solution. Creating a web browser-based portal to the Verifying Compiler could provide the ease of use and functionality that could potentially increase the level of exposure to the tool and the RESOLVE in general.

II. Development and Design

The RESOLVE Web Interface in its current form is an evolution of past efforts at Clemson. The first iteration of the online tool was strictly designed to demonstrate the current abilities of the Verifying Compiler at the time. This version, written in PHP, successfully integrated the Verifying Compiler with a commonly used browser. However, there were some shortcomings that would eventually need to be addressed. Firstly, different web browser implementations meant that some important code did not function properly or at all in some browsers. Secondly, this version only allowed users to use RESOLVE files that were hard-coded into the system.

This work was supported in part by the National Science Foundation under Grants DUE-0633506, CCF-0811748, and DMS-0701187.

To address issues like these and make improvements to the system, we decided to make use of more modern web technologies to provide improved functionality [1]. Although the basic look and feel presented to the user remained mostly unchanged, the core technologies were updated. The main improvement to the user experience that we were looking for was to allow users to be able to make real-time changes to and write their own RESOLVE code that would then be sent to the Verifying Compiler. We chose to make use of AJAX (Asynchronous Javascript and XML) to create the dynamic experience that we were looking for. To make use of AJAX for the Web Interface the implementation needed to be split into two parts, a server backend and a web site frontend. To maintain compatibility with the Verifying Compiler we decided to use Java Servlets for the server backend and HTML with Javascript for the frontend. In this version editable textboxes allow the web user to modify or create their own RESOLVE source code which is then transferred using AJAX to the Java Servlet. The Servlet writes the user code to the file system, instantiates the Verifying Compiler, and returns the output to the user through the web frontend.

The current version of the Web Interface employees the same technologies as the previous version, but the web frontend has been redesigned. The impetus for the redesign was to create a more user-friendly interface. Both prior iterations of the interface have presented existing RESOLVE source files and Verifying Compiler output in simple textboxes. This version provides a more intuitive and simpler design; RESOLVE files in the pre-existing library are presented in an easier to navigate fashion. Additionally, input and output files are no longer displayed as plain text. Raw text files are converted to HTML and stored locally in the browser for later use. Presenting in HTML allows us to add additional formatting such as bold-faced keywords that make RESOLVE source code and the Verifying Compiler output more readable.

III. Education

Another area of emphasis of the RSRG is the integration of the RESOLVE language and good software development practices with the classroom environment. The Web Interface provides an ideal platform to use to introduce Computer Science students to the connections between mathematical reasoning and software development. The interface will allow students to easily experiment with RESOLVE code, verification condition (VC) generation, and the resulting VC proofs without losing time trying to install additional development tools.

IV. Future work and Conclusions

Modernization of the Clemson RESOLVE Web Interface to be more dynamic and understandable has enhanced its overall usability. As the tool continues to evolve we hope to provide a higher level of integration with the existing Verifying Compiler, allowing a tighter meshing between the two systems. This would allow the web interface to continue improving how the Verifying Compiler output and results are integrated and presented within the web browser. We hope the tool as it is and with future improvements will help make RESOLVE, in particular, and contract-based software development, in general, more accessible.

V. References

[1] Cook, C. and Smith H., "Developing and Verifying RESOLVE Components Via a Web Interface," *Workshop: RESOLVE 2009*, 22 Sep. 2009; http://www.cse.ohio-state.edu/rsg/RESOLVE-2009/Cook_Smith.pdf.