\begin{thesisabstract}

JavaScript is one of the most important and prevalent programming languages today. It is the language of the web browser and it has seen a surge in active development since the rise of the Ajax approach for developing highly interactive web applications. Often misunderstood as a simple scripting language, JavaScript is in fact powerful and expressive. As web applications have grown in sophistication, so have their user interfaces, which are predominately implemented as client-side JavaScript. This growth in complexity has resulted in a vast array of JavaScript frameworks, best practices and design patterns.

Since JavaScript applications involve creating user interfaces, adopting a visual development approach would seem a natural choice to help manage complexity and improve program understanding. One possible approach is to allow the developer to visually relate semantically meaningful elements on a web page to the actual JavaScript source code that implements its behavior. This can be accomplished by leveraging context information during JavaScript execution. Additionally, control-flow graphs are dynamically generated to help the developer understand how a series of function calls might be related based on events triggered in the user interface.

Inspired by work from Li and Wohlstadter, we implement the above approach by creating a program called FireInsight. It integrates with the Mozilla Firefox web browser and the Firebug development tool. We use an HTTP proxy to instrument JavaScript source code with our own analysis code. We analyze this approach and evaluate its effectiveness by applying FireInsight to an open-source web application called Java Pet Store 2.0.

\end{thesisabstract}