Tool-assisted Code-splitting for GWT

Alan Hicks, Supervisor: Dr. Joseph R. Kiniry
MSc in Advanced Software Engineering - Thesis Proposal

Overview

- *Subject Area:* Software Engineering
- *Pre-requisite:* Good knowledge of Java and Web Technologies
- *Co-requisite:* Knowledge of Google Web Toolkit, GWT Code-Splitting, GWT Story of Your Compile (SOYC), Eclipse Platform, Eclipse Plug-in Architecture, Quantitative Data Analysis, will be obtained during the course of this project
- *Subject Coverage:* Java, Google Web Toolkit, Quantitative Data Analysis, Integrated Development Environments, Eclipse Plug-in Development
- *Project Type:* Research, Design and Implementation
- *Hardware:* PC or workstation (on nearly any operating system)

Description

Google Web Toolkit (GWT) allows developers to build JavaScript applications using the Java language. The toolkit converts Java code into JavaScript. GWT provides abstraction from the low-level browser quirks of JavaScript development. The comprehensive set of features provided by GWT represents a viable alternative to JavaScript programming.

Over the last several years GWT has become increasingly popular. The resulting tools, which have been developed by the GWT community, are proof of this popularity. One example of such a tool is the GWT Eclipse Plug-in which enables simple compilation and deployment of GWT-based applications within the Eclipse environment.

Despite advances in GWT, e.g. compiler optimisations, a potential bottleneck at application start-up is still relatively common. The cause of this bottleneck is the need to download the entire JavaScript application. This methodology represents a poor use of resources as the end user may not require the full feature set of the application. Future releases of GWT will have a feature that aims to alleviate this bottleneck. The GWT team is introducing a technique called code-splitting. Code-splitting gives developers an API to support splitting their application up into chunks that are loaded on demand, as triggered by subsequent events. The use of code-splitting separates the download impact over the entire user's session and thus is intended to reduce the number of download bottlenecks. An additional benefit is that the user will download only the code they require for the features of the application they use.

The goal of this project is to extend the code-splitting technique by providing a mechanism that can help objectively identify code segments for code-splitting. This analysis will be

based upon either recorded user actions or code analysis. The user will benefit because their required downloads will be staggered according to the architecture of the application and the application will feel more like a responsive desktop application rather than a web application.

An Eclipse Plug-in will be employed to implement this advancement of code-splitting techniques. The intention of the Plug-in is to analyse the application source code and quantitative data and display some graphical information to the developer. The Plug-in will thus highlight to the developer where code-splitting and or pre-emptive code loading can be inserted.

Mandatory

- Familiarisation with GWT in particular the new code-splitting and SOYC features.
- Familiarisation with Eclipse Plug-in development.
- Familiarisation with Eclipse development environment and profiler tools.
- Familiarisation with tools for analysing web application flow and usage.
- Develop at least one algorithm, either via source code or user interaction analysis, that suggests to the developer how their code might be split.
- Perform requirements analysis and design plug-in.
- Implement the GWT Code-splitter Assistant plug-in.
- Release Eclipse plug-in that will help GWT developers build web applications that use GWT code-splitting technology intelligently.

Discretionary

- Demonstrate working plug-in.
- Develop the complementary algorithm to that mentioned in the mandatory requirements.

Exceptional

- Publish a peer reviewed paper on the results.

Sources of information and preparatory reading

- The GWT web site http://code.google.com/webtoolkit/
- GWT Code Splitting Wiki http://code.google.com/p/google-web-toolkit/wiki/CodeSplitting
- Google Developer Conference Session Videos http://code.google.com/events/io/sessions.html
- GWT in Action Manning Publications 2007, Robert Hanson and Adam Tacy.
- Eclipse http://www.eclipse.org