Visualizing the Word Structure of Java Class Names

Craig Anslow   James Noble
Stuart Marshall

Victoria University of Wellington, New Zealand
{craig,kjx,stuart}@mcs.vuw.ac.nz

Ewan Tempero

University of Auckland, New Zealand
ewan@cs.auckland.ac.nz

Abstract

Large amounts of software have been written since the Java language was created. There is little known about the word structure Java within class names. We have created visualizations of words used in class names from the Java API specification and 91 open-source Java applications. Our visualizations will help expose which words are used in practice.

Categories and Subject Descriptors   D.2.6 [*Programming Environments*]: Interactive environments

General Terms   Design

Keywords   Class Names, Evolution, Java, Software Visualization

1. Introduction

We are interested in understanding what Java software looks like to help with software reuse, maintenance, and re-engineering. We believe creating software visualizations will help to assist end-user developers to learn the Java API specification and understand the structure of Java software.

We have a corpus of Java software[1] (1) that contains 91 open-source Java applications. The corpus is used for conducting empirical studies to help understand how software engineers create code and the relationship between the code structure and quality attributes. Our project requires better techniques for understanding the software from the corpus. In this paper we explore creating visualizations of the words used in Java class names from the Java API specification and our software corpus using the Many Eyes[2] (2) visualization web application.

---

[1] http://www.cs.auckland.ac.nz/~ewan/corpus/

[2] http://www.many-eyes.com
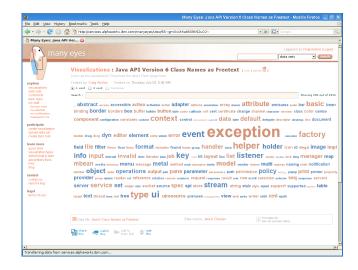
2. Software Visualizations

Figure 1. Tag Cloud Visualization of the words used in the class names from the Java API Specification version 1.6.

We have conducted experiments of visualizing the words used in class names of the Java API specification and from our software corpus. Firstly, we have visualized the words in the class names from the Java API Specification version 1.6. Secondly, we have visualized the evolution of words in class names from Java version 1.1 and Java version 1.6. Thirdly, we have visualized the ordering of words in the Java class names from the specification. Finally, we have visualized the words in the class names from the 91 open-source Java applications in our software corpus. We do not distinguish between interfaces and classes.

Figure 1 shows a Tag Cloud visualization of the words used in the class names from the Java API specification version 1.6. For example the Java class name AbstractColorChooserPanel becomes Abstract (position one), Color (two), Chooser (three), and Panel (four). The tag cloud shows the most common words used in Java class names are Exception (381 occurrences), UI (133), Helper (128), Type (120), Event (116), and Factory (99). The word Exception is used twice as many times as any other word in the specification

which shows that the design of Java relies quite heavily on capturing exceptions at run-time.

Figure 2(a) shows a comparison of the class names between Java version 1.1 (red colour) and version 1.6 (blue colour). Java 1.1 contains 477 classes and Java 1.6 contains 3777 classes. This is an interesting visualization as it shows how the words used Java class names have evolved between different versions of the specification. All of the words used in Java 1.1 have also been used in Java 1.6, there is no word that has been so called deprecated. There are, however, a number of additional words used in Java 1.6 which is to be expected being a more recent version. The word Exception is the most prominent word in both versions. There are no words associated with XML (e.g. XML, XPath) in the version 1.1 tags which suggests that this version of Java did not have any XML libraries at that time.

Figure 2(b) is a Tree Map visualization which shows the ordering of the words in class names from Java 1.6. The current screen-shot of the visualization shows the words in the class names ordered as in the order you would find them in a Java class name (e.g. AbstractColorChooserPanel represented as Abstract, Color, Chooser, and Panel). There are 1217 unique words in position one, 761 in position two, 409 in three, 186 in four, and 70 in five. The tree map allows a user to change the order of the words in the class name to see which words are the most prominent in each position in the class name. The tree map shows that the most prominent word in position one is the word Metal followed by Basic, Default, Order, and then Key. However, the most prominent word in positions two, four and five are variants on the word Border. One thing to observe is that the word Exception does not appear prominently at all in this visualization given the frequency it appears in Figure 1.

Figure 2(c) shows a visualization of the class names from the 91 open-source Java applications in our software corpus. We only considered the class names of Java files. The visualization contains approximately 51,000 classes. The most common words that are used are Test (3847 occurrences), Action (1541), Impl (1451), Factory (1333), Exception (1089), and Data (948). This suggests that there is an emphasis of testing in these applications and perhaps a test driven development approach was followed.

3. Summary

We have created visualizations with Many Eyes of the words used in class names of the Java API specification and 91 open-source Java applications. Our visualizations have exposed that the most common words used in Java class names are Test, Action, Impl, and Exception. Knowing what the most common words in Java are can help developers create coding standards for their class names. In the future we plan to explore visualizing package, method, and field names. We would also like to experiment with different visualization

(a) Tag Cloud Visualization of the words used in the class names of Java 1.1 (red) and Java 1.6 (blue).

(b) Tree Map Visualization of the ordering of words used in the class names from the Java API Specification version 1.6.

(c) Tag Cloud Visualization of the words used in the class names from our software corpus which contains 91 applications.

Figure 2. Using the Many Eyes web application to create software visualizations.

techniques and tools to see if we can expose other interesting patterns of word usage, and conduct some user testing.

Acknowledgments

References

[1] Hayden Melton and Ewan Tempero. The CRSS metric for package design quality In Proceedings of *Australasian Conference on Computer Science (ACSC)*, pages 201–210. ACS, 2006.

[2] Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. ManyEyes: a Site for Visualization at Internet Scale In *Transactions on Visualization and Computer Graphics (TVCG)*, pages 1121–1128. IEEE, 2007.