

Scripting Diagrams with EyeSee

Matthias Junker, Markus Hofstetter

Software Composition Group, University of Bern, Switzerland

Abstract

Presenting numbers in the right way is crucial for understanding their meaning. We present EyeSee, a diagram drawing engine that allows for programatic specification of the presentation, while offering default values that produce uncluttered diagrams.

Keywords: diagrams, visualization, meta-modeling, tooling, metrics, testing, static analysis, dynamic analysis, evolution analysis

1 Introduction

Using diagrams to reason about quantitative data is a common practice in today's age of information. However, reading and understanding diagrams is not always as easy as it could be. In many cases, embellishments capture more attention than data [1]. In this paper, we present EyeSee, a free diagram drawing engine that aims to provide default values that produce uncluttered diagrams. In particular, EyeSee focuses on a simple motto proposed by Tufte: *Minimize chart-junk and maximize data ink* [2].

Furthermore, most diagram drawing tools are concentrated on the user interface, and typically they require input data in a certain format. With EyeSee we address the researcher that builds the analysis, and that is comfortable with his programming environment. Thus, EyeSee does not require the data to be passed in a fixed format, but rather it allows the user to specify programmatically

how to extract the data from the model he is using.

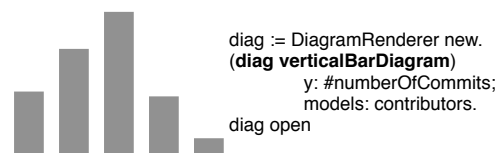
While on the one hand we strive to produce clutter-free diagrams, on the other hand, we also focused on the ability of the user to control all the details of the presentation with as little effort as possible. For this purpose we provide scripting methods that allow for changing most of the properties of a diagram (e.g., the color of the elements, the type of axis, the size).

Our prototype is built in VisualWorks Smalltalk and works with Smalltalk domain models, but the same approach can be applied to any programming language.

2 EyeSee by example

In this section we show the basic facilities of EyeSee using hands-on examples. As domain model, we use a collection called *contributors* containing the contributors to a certain project, as obtained from a versioning system. Every contributor has the attributes name, number of commits, lines of code, team and versions.

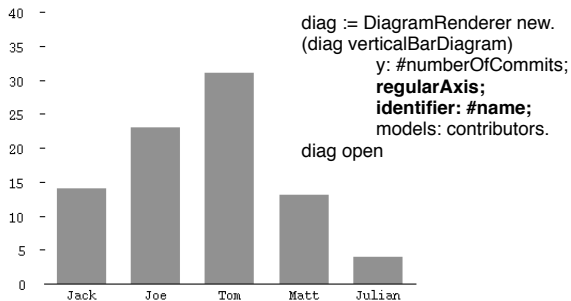
We start with a very simple script that opens a vertical bar diagram where each bar shows the number of commits of the contributor. In the script, *numberOfCommits* is a method in the Contributor class:



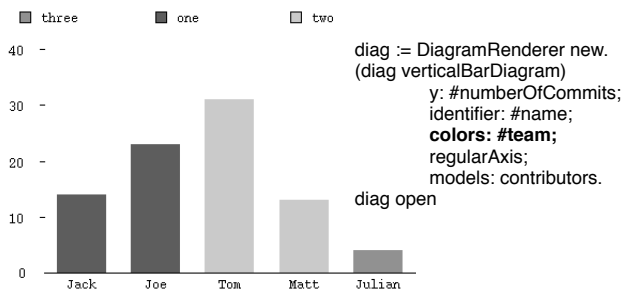
In the above diagram we cannot see which bar

is from which contributor, and we also do not know the actual values of the bars. Thus, we want to add axis and labels to the bars.

We accomplish this with 2 additional lines of code. The new line *regularAxis* adds axis with ticks and labels. In regular axis, the distance between two ticks is always the same. Instead, we could also use *valueAxis*, which draws ticks dependent on the values which get displayed in the diagram. For putting a label below every bar, we can use the *identifier* keyword, to tell the diagram, how it can get the data for the labels. In our example, we use the *name* of the contributor to identify the bars (where *name* is a method in the Contributor class):

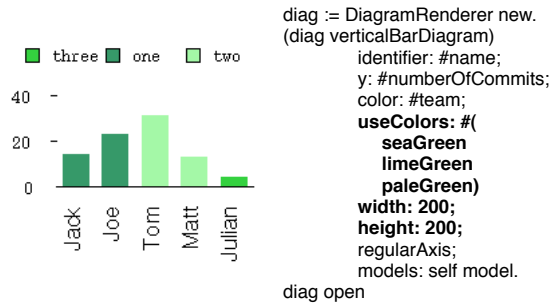


We might also want to encode other information in our diagram. We have this possibility by using colors or shades of gray for the bars. In our example, knowing for each contributor the team he belongs to, we also want to reveal this information. Thus, we add one line to specifying that we want a different color for each new *team*.



By default, the values for the color get encoded with gray scale colors. By adding the expres-

sion *lightColors* or *strongColors*, we can use a set of ten strong or light Colors which are distinct enough so they can be easily separated [3]. However, custom colors can also be used for the encoding, by using the *useColors*: message:



In the above example, not only did we specify the colors we want to use, but we also specify the width and height of the diagram. In our example, as there is not enough space for the identifiers to be displayed horizontally, they automatically get rotated and are displayed vertically. Of course we can also configure to have them rotated manually by using the *rotatedIdentifiers* message.

3 Conclusion

The goal of *EyeSee* was to create a diagram drawing engine that lets the user visualize his data without forcing him to convert it into a fixed format. The provided default values produce clear and clutterfree diagrams. ¹

References

- [1] S. Few. *Show me the numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, 2004.
- [2] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2nd edition, 2001.
- [3] C. Ware. *Information Visualization*. Morgan Kaufmann, 2000.

¹The latest version of *EyeSee* can be obtained on the Cincom Public Repository, see <http://smallwiki.unibe.ch/moose/tools/eyesee/> for more details.

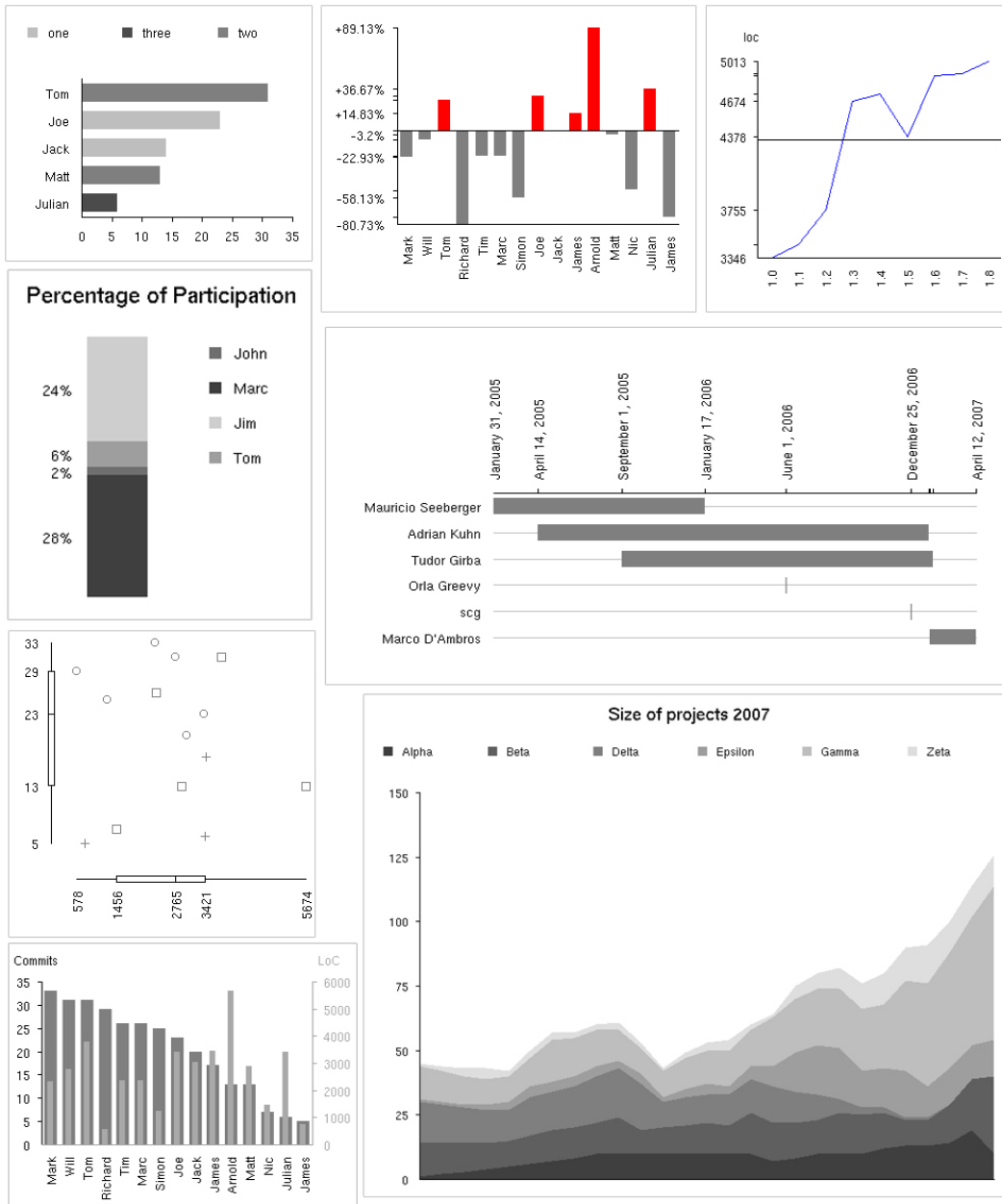


Figure 1. Several diagrams supported by EyeSee