COMETS: A Dataset for Empirical Research on Software Evolution using Source Code Metrics and Time Series Analysis

Cesar Couto[1,2], Cristiano Maffort[1,2], Rogel Garcia[1], Marco Tulio Valente[1]

[1]Department of Computer Science, UFMG, Brazil
[2]Department of Computing, CEFET-MG, Brazil
{cesarfmc,maffort,rogelgarcia,mtov}@dcc.ufmg.br

Abstract

This paper documents a time series dataset on the evolution of seventeen object-oriented metrics extracted from ten open-source systems. By making this dataset public our goal is to assist researchers with interest in software evolution analysis and modeling.

1 Introduction

COMETS (Code Metrics Time Series) is a dataset collected to support empirical studies of source code evolution. The dataset includes information on the evolution of seventeen well-known software metrics for the following open-source Java-based systems:

1. Eclipse JDT Core: compiler and other tools for Java
2. Eclipse PDE UI: components to create Eclipse plug-ins
3. Equinox: OSGi implementation
4. Lucene: text search engine library
5. Hibernate: persistence framework
6. Spring: application development framework
7. JabRef: bibliography reference manager
8. PMD: source code analyzer
9. TV-Browser: electronic TV guide
10. Pentaho Console: console for business intelligence suite

2 Time Series in COMETS

The time series provided in the dataset have been collected in intervals of bi-weeks (14 days). For each system, Table 1 shows the time frame considered to create the time series. The figure also shows the number of source code versions considered in this time frame (always with an interval of one bi-week between consecutive versions).

For each system, the dataset includes the source code of its versions, in intervals of bi-weeks, during the time frame indicated in Table 1. Moreover, the dataset includes time series with the values of the following source code metrics measured at the level of classes [1, 4]:

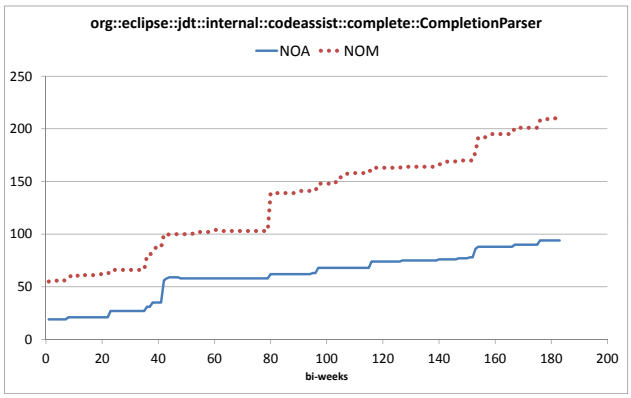
- Size Metrics: Number of attributes (NOA), Number of public attributes (NOPA), Number of private attributes (NOPRA), Number of attributes inherited (NOAI), Number of lines of code (LOC), Number of methods (NOM), Number of public methods (NOPM), Number of private methods (NOPRM), Number of methods inherited (NOMI).
- Coupling Metrics: fan-in, fan-out.
- CK Metrics: Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number Of Children (NOC), Coupling Between Objects (CBO), Response For a Class (RFC), and Lack of Cohesion in Methods (LCOM).

System	Period	Versions
Eclipse JDT	07/01/2001 - 06/14/2008	183
Eclipse PDE	06/01/2001 - 09/06/2008	191
Equinox	01/01/2005 - 06/14/2008	91
Lucene	01/01/2005 - 10/04/2008	99
Hibernate	06/13/2007 - 03/02/2011	98
Spring	12/17/2003 - 11/25/2009	156
JabRef	10/14/2003 - 11/11/2011	212
PMD	06/22/2002 - 12/11/2011	248
TV-Browser	04/23/2003 - 08/27/2011	221
Pentaho	04/01/2008 - 12/07/2010	72
Total	-	1571

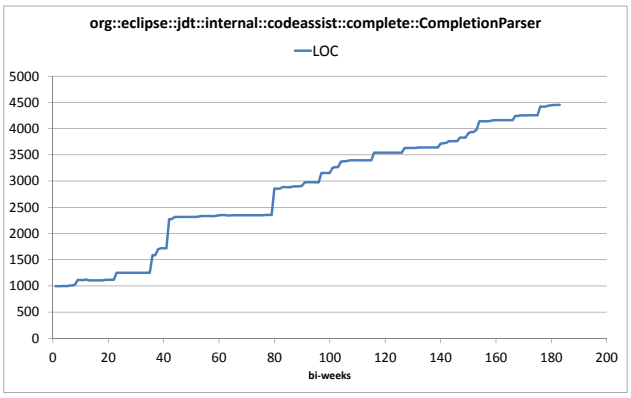
Table 1: Dataset

Basically, for each system S and metric M, there is in the COMETS dataset a csv file whose lines represent the classes of S and whose columns represent the bi-weeks considered when extracting the versions of S. A cell (c, t) in this file contains the value of the metric M, measured for the class c, in the bi-week t.

Figure 1 shows examples of the time series provided in the dataset. Figure 1(a) shows the time series with the values of two metrics (NOA and NOM) collected for one of the classes in the Eclipse JDT system. Figure 1(b) shows the time series describing the evolution of this same class in terms of lines of code. Finally, Figure 2 shows the evolution of the number of attributes (NOA) and number of methods (NOM) considering all classes in the Eclipse JDT.



(a) NOA and NOM



(b) LOC

Figure 1: NOA, NOM, and LOC (for a Eclipse JDT class)

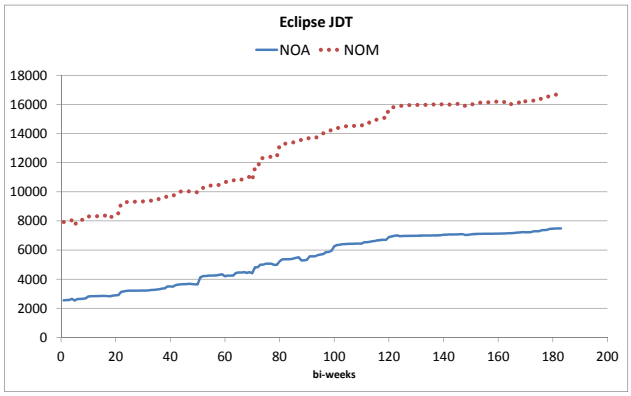


Figure 2: LOC (for Eclipse JDT)

3 Extraction Process

To create the dataset, we extracted in intervals of bi-weeks the source code of each system from its revision control platform. We used the Moose platform[1] to extract the metrics values for each class of each considered version, excluding only test classes. Particularly, we relied on VerveineJ—a Moose application—to parse the source code of each version and to generate MSE files. MSE is the file format supported by Moose to persist source code models. Because Moose's current version does not calculate three CK metrics (CBO, LCOM, and RFC), we extended the platform with new routines for this purpose. In the dataset, we also included the MSE files we used to extract the metrics time series.

[1]http://www.moosetechnology.org.

4 Related Datasets

Our dataset was generated from a dataset originally conceived by D'Ambros et al. to evaluate bug prediction models [3]. The D'Ambros dataset includes temporal time series with metrics values for the following systems: Eclipse JDT Core, Eclipse PDE UI, Equinox, and Lucene. We extended this benchmark in two ways: (a) by extracting again all source code versions and recalculating the metrics; (b) by expanding the original time series of two systems: Eclipse JDT Core (from 91 to 183 bi-weeks) and Eclipse PDE UI (from 97 to 191 bi-weeks), and (c) by including historical information from six more open-source systems (Hibernate, Spring, JabRef, PMD, TV-Browser, and Pentaho).

Helix is another dataset that provides temporal information on the values of source code metrics [7]. However, the dataset lacks information on some of the metrics included in COMETS, including the CK metrics (with the exception of NOC and DIT) and coupling metrics.

The Qualitas Corpus is another well-known dataset for empirical studies in software engineering [6]. It contains information on 111 systems, but it only provides information on the evolution of 14 systems (only one with more than 70 versions). On the other hand, in our dataset all systems have at least 72 versions and five systems have more than 180 versions. Moreover, the Qualitas Corpus does not include temporal information on the values of source code metrics.

5 Final Remarks

By making this dataset public our goal is to assist other researchers with interest on source code evolution analysis. For example, we have used the dataset on a study on the relation between class evolution categories (e.g. supernovas, white dwarfs, pulsars, etc) and source code metrics [5].

The COMETS dataset is available for download at:

http://www.java.llp.dcc.ufmg.br/comets

As future work, we are planning to extend COMETS with information on the number of defects at the level of classes, which is the central data need to evaluate, for example, bug prediction models [2, 3].

Acknowledgments: This research was supported by grants from CNPq, CAPES, and FAPEMIG.

References

[1] S. R. Chidamber and C. F. Kemerer. A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20:476–493, 6 1994.

[2] Cesar Couto, Christofer Silva, Marco Tulio Valente, Roberto Bigonha, and Nicolas Anquetil. Uncovering causal relationships between software metrics and bugs. In *16th European Conference on Software Maintenance and Reengineering (CSMR)*, pages 223–232, 2012.

[3] Marco D'Ambros, Michele Lanza, and Romain Robbes. An extensive comparison of bug prediction approaches. In *7th Working Conference on Mining Software Repositories (MSR)*, pages 31–41, 2010.

[4] Michele Lanza, Radu Marinescu, and Stéphane Ducasse. *Object-Oriented Metrics in Practice*. Springer-Verlag, 2005.

[5] Henrique Rocha, Cesar Couto, Cristiano Maffort, Rogel Garcia, Clarisse Simoes, Leonardo Passos, and Marco Tulio Valente. Mining the impact of evolution categories on object-oriented metrics. *Software Quality Journal*, 2013. To appear.

[6] Ewan Tempero, Craig Anslow, Jens Dietrich, Ted Han, Jing Li, Markus Lumpe, Hayden Melton, and James Noble. Qualitas corpus: A curated collection of Java code for empirical studies. In *Asia Pacific Software Engineering Conference (APSEC)*, pages 336–345, December 2010.

[7] Rajesh Vasa, Markus Lumpe, and Allan Jones. Helix - Software Evolution Data Set, 2010. http://www.ict.swin.edu.au/research/projects/helix.