Control flow in the wild : a first look at 13K Java projects (Extended abstract)

Control Flow in the Wild
A first look at 13K Java projects

Davy Landman*, Alexander Serebrenik*†, Jurgen Vinju*
* Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
{Davy.Landman, Jurgen.Vinju}@cwi.nl
† Eindhoven University of Technology, Eindhoven, The Netherlands
a.serebrenik@tue.nl

I. Introduction

We are interested in the understandability of software. Maintainability models such as the SIG model use cyclomatic complexity to measure understandability. However, doubts have been raised about the relation between cyclomatic complexity and understanding of the code. In a grounded theory approach we first observe control flow in a large corpus. Which in the long term will enable us to find categories and create well-founded metrics or indicators for understandability.

We present our early observations of Control Flow Patterns (CFPs) [1] in the Sourcerer Corpus [2], a set of 13 thousand Java projects. We observe saturations when CFPs belonging to two or more systems are considered, but no saturation when all patterns are considered. Most observed patterns are unique, only present in one system, moreover they are small, less than 20 statements. We conclude with questions for future research.

II. Experiment

We took the Sourcerer Corpus which contains 18K (13K non empty) Java projects. Using a Java grammar and RASCAL [3] we parsed all Java files. All methods were transformed [1] into CFPs.

A CFP is an AST created by removing all statements not related to control flow. Table I contains a list of Java's language constructs kept. The last step is to change all expressions inside the arguments of the constructs into an empty string.

TABLE I
JAVA LANGUAGE CONSTRUCTS USED IN A CFP.

if	if else	switch	case	labeled	continue	break
for	while	do while	return	throw	synchronized	try

Table II describes how large the Sourcerer corpus is, and how many CFPs we extracted and how many of those CFPs were unique to one system.

TABLE II
SIZE OF SOURCERER CORPUS AND EXTRACTED CFPS

Size	Files	LOC†	Methods	CFPs	$CFPs^{\ddagger}_{unique}$
19GB	2M	477M	23M	678K	516K

† measured using `wc -l`
‡ CFPs only observed in one system.

III. Observations

Figure 1 shows the amount of CFPs observed, where we see that almost every time when we add a new systems, we observe new patterns. Narrowing our definition of a pattern, only considering patterns present in 2 or more systems, we observe a saturation. Even more so for patterns shared by 3 or more and 4 or more. Figure 2 shows these narrowed definitions in more detail.

Unique CFPs are patterns only occurring in exactly one system. The almost linear growth in Figure 1 raises the question what causes it. Figure 3 shows that this is not primarily caused by large patterns, that most unique patterns are actually smaller then 20 control flow statements.

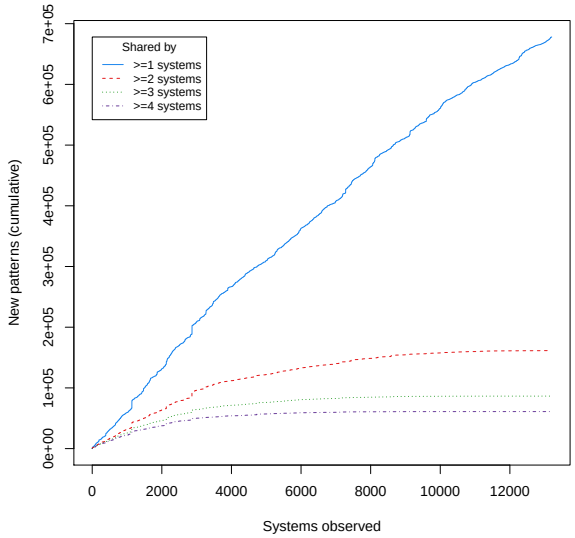

Fig. 1. Saturation of the patterns in the Sourcerer corpus. The four lines represent the saturation of patterns appearing in x or more systems.

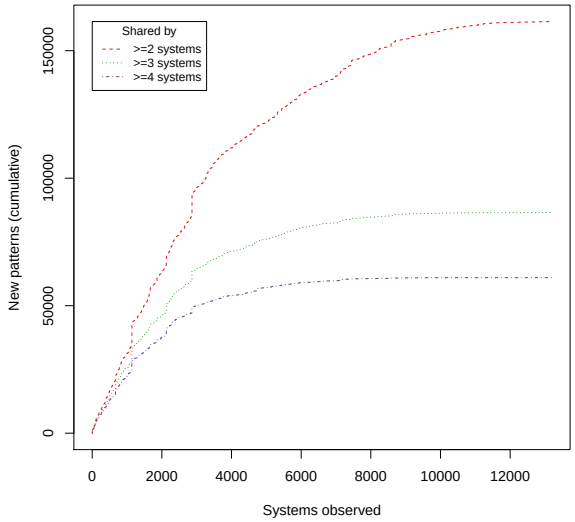

Fig. 2. Zoomed in on the patterns shared in more than one system.

The theoretical reason for so many small *unique* CFPs, is the exponential growth in possible patterns. For size 4 there are already 2.474.634 possible CFPs. Figure 4 shows how many different CFPs per size were observed and it shows the theoretical maximum.

Figure 5 shows the distribution of the size of a CFP and in how many systems it occurs. Here we can see that even the larger CFPs are shared. Eye-balling these larger shared CFPs revealed code clones and code generated by the same generator. We also observed code clones where the full library was embedded.

IV. Open Questions

As future work we have the following questions:

1) Are systems with a lot of CFPs not using OO constructs?
2) Can we find categories of CFPs?
3) Are CFPs abstract enough?
4) Can we find a relation between the naming of a method and it's CFP?
5) If we observe more systems, would the saturation change?
6) If we analyse non Java systems, would we find similar patterns and saturations?
7) What would be the impact of removing clones on the amount of shared patterns?

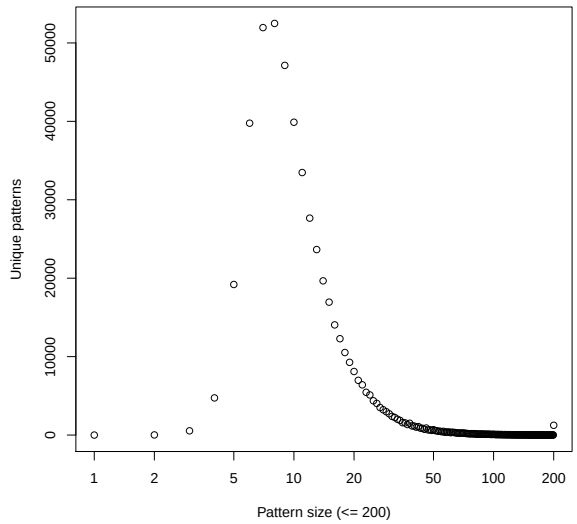

Fig. 3. How many unique patterns of a certain size are found. Patterns equal or larger than 200 are grouped to show that the long-tail does not contribute that much to the continuous growth of patterns observed.

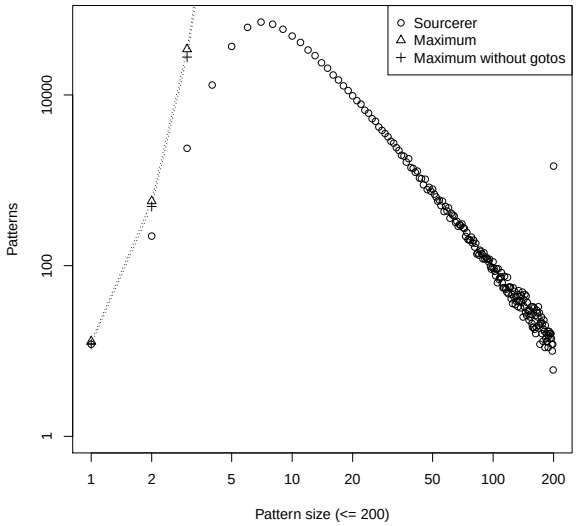

Fig. 4. How many different patterns of a certain size were observed and the maximum possible (only shown for $1-3$ due to the exponential growth). The "maximum without GOTOs" does not take into account the "structured" GOTOs in Java.

8) Could CFPs be used to fingerprint systems?
9) Why is there so much control flow in an OO language?

REFERENCES

[1] J. J. Vinju and M. W. Godfrey, "What does control flow really look like? eyeballing the cyclomatic complexity metric," in *Ninth IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE Computer Society, 2012.

[2] E. Linstead, S. K. Bajracharya, T. C. Ngo, P. Rigor, C. V. Lopes, and P. Baldi, "Sourcerer: mining and searching internet-scale software repositories," *Data Min. Knowl. Discov.*, vol. 18, no. 2, pp. 300–336, 2009.

[3] P. Klint, T. van der Storm, and J. J. Vinju, "RASCAL: A Domain Specific Language for Source Code Analysis and Manipulation," in *Proc. 9th IEEE International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE, September 2009, pp. 168–177.

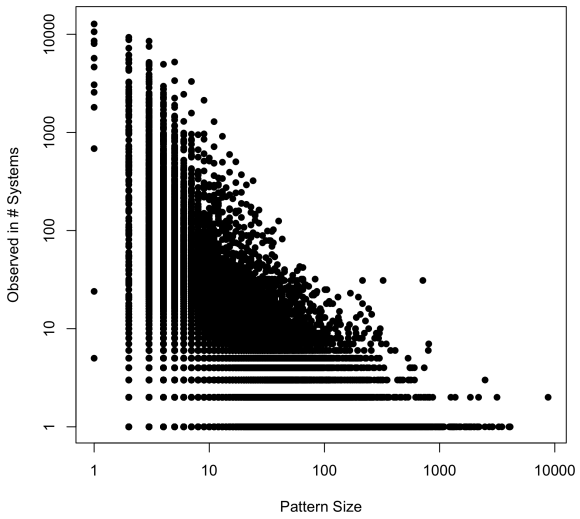

Fig. 5. A scatterplot of the pattern's sizes and in how many projects they occurred.