Object-Centric Instrumentation with Pharo

Steven Costiou

February 5, 2019

Copyright 2017 by Steven Costiou.

The contents of this book are protected under the Creative Commons Attribution-ShareAlike 3.0 Unported license.

You are **free**:

- to **Share**: to copy, distribute and transmit the work,
- to **Remix**: to adapt the work,

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page:
http://creativecommons.org/licenses/by-sa/3.0/

Any of the above conditions can be waived if you get permission from the copyright holder. Nothing in this license impairs or restricts the author's moral rights.

Your fair dealing and other rights are in no way affected by the above. This is a human-readable summary of the Legal Code (the full license):
http://creativecommons.org/licenses/by-sa/3.0/legalcode

Layout and typography based on the `sbabook` LaTeX class by Damien Pollet.

Contents

Illustrations ii

1 What we are talking about 1

1.1 Illustration example 1

1.2 Evaluation criteria 2

1.3 Performance overhead evaluation 2

Bibliography 3

Illustrations

1-1 Trivial example for object-centric instrumentation 1

CHAPTER 1

What we are talking about

In the next chapters, we evaluate each technique following a three-fold evaluation. First, the studied technique is applied on a simple example of object-centric instrumentation. Second, the technique is evaluated against a set of desirable properties. Finally, performance overhead are evaluated. Only the raw solution is evaluated, without considering the possibility of enhancing the technique by building something on top.

1.1 Illustration example

Each solution is experimented on a trivial example of object-centric behavior instrumentation. This example is illustrated in script 1-1. Two instances of `OrderedCollection` are created, and to each of these instances is sent the `add:` message with a string as a parameter. The instrumentation must happen in-between. We want to instrument the `col1` object, so that when the `add:` message is received, the size of the collection and the added object (passed as parameter) are printed in the `Transcript`.

Listing 1-1 Trivial example for object-centric instrumentation

```
|col1 col2|
col1 := OrderedCollection new.
col2 := OrderedCollection new.

"...instrumentation must happen here..."

col1 add: 'Hello World'.
col2 add: 'Hello World'.
```

1.2 Evaluation criteria

Each solution is evaluated agains the following desirable properties.

Property	Definition
Manipulated entity	The unit of instrumentation (*e.g.* a class, a Trait, an object...)
Reusability	The entity can be reused to instrument different objects
Flexibility	Instrumentation does not put constraint on the source code or in the coding style
Granularity	The level of at which behavior can be instrumented (*e.g.* method, AST...)
Integration	Instrumentation does not break system features

1.3 Performance overhead evaluation

- source code with instrumentation example
- describe the used method
- describe the limitations of the method

Bibliography