Artifacts Supporting the Paper

"Selection and Presentation Practices for Code Example Summaries"

in the Proceedings of the International Symposium on Foundations of Software Engineering, 2014

Annie T. T. Ying and Martin P. Robillard

November 6, 2014

1 How to Obtain the Artifact

`http://annieying.ca/fse2014/fse2014-artifacts.zip`

The assumption is that you have Bash, R, Perl, and the Latex tikz package installed.

2 How to Unpack the Artifacts

2.1 Getting the VM Up

Unzip the VM and start it in VirtualBox. The password is `annie123`.

2.2 Directory Structure

The directory containing the artifacts is `~/SummarizationArtifacts`. Each type of artifacts described in the study set-up (Section 3 of the paper) corresponds the following sub-directories:

- The code fragments for the summarization tasks (Sections 3.1 and 3.2) and the summaries generated by the participants (Section 3.1) are in the following directory:

  `~/SummarizationArtifacts/code-fragment-original-summary-pairs/`

- The context of the summarization tasks ("Query" and "Android API" in Figure 2 and Section 3.3) and the participant-to-task assignment (Section 3.4) are in the following file:

  `~/SummarizationArtifacts/task-participant-assignment.csv`

- The main result of this paper is the catalog of summarization practices we observed from the study. As we described in Section 4 of the paper, we have shown the evidence to the summarization practices using sparkline histograms, indicating the distribution of observations of a given practice for a participant (each bar) over the ten code fragments (the vertical axis). The data for generating the sparkline histograms are in the following directory:

  `~/SummarizationArtifacts/stats/`

- The Latex code to generate the histograms are in the following directory:

  `~/SummarizationArtifacts/sparklines/`

3 How to Use the Artifacts

We demonstrate how to construct the sparkline histograms, the major evidence supporting the summarization practices.

3.1 Extracting the Differences between Code Fragments and Summaries

First, we systematically extracted the textual differences between code fragments and the corresponding summaries. The result is in the following file:

```
~/SummarizationArtifacts/stats/selection-presentation-transformations.csv
```

Notable columns of the file are as follows:

- *user*: Participant ID
- *sid*: ID of the code fragment. Each code fragment was assigned to three participants.
- *transformation.type*: Type of low level selection or presentation transformation from the original code fragment to the summary
- *original.line.number*: Line number of the original code fragment the transformation is applied
- *summary.line.number*: Line number of the summary line number the transformation is applied

3.2 Aggregating the Differences into Practices

Presentation Practices (Except Formatting Practices)

We use a series of R scripts to aggregate the differences (the evidence) to form the practices in the paper. For presentation practices (Section 6, except 6.4), here is how the script can be run:

```
cd ~/SummarizationArtifacts/stats/presentation/
R CMD BATCH stats.R
```

This command generates the evidence data, `icon-data-<PRACTICE>*.csv`. Each of these files is the data used to generate the corresponding practice's histogram. For example, for the practice *shortening identifiers*, the histogram is  . The data is in `icon-data-shorten-identifiers.csv` whose content is as follows:

```
1 8
2 6
3 5
4 5
5 4
6 4
7 3
8 3
```

This information signifies that eight participants (the rows) showed evidence of the "shortening identifiers" practice in 8, 6, 5, 5, 4, 4, 3, and 3 code fragments respectively. Section 4 of the paper uses this example as well.

Formatting Presentation Practices

For formatting practices (Section 6.4), here is how the script can be run:

```
cd ~/SummarizationArtifacts/stats/formatting/
R CMD BATCH stats.R
```

Selection Practices

Analogously, for selection practices (Section 5), here is how the script can be run:

```
cd ~/SummarizationArtifacts/stats/selection/
R CMD BATCH stats.R
R CMD BATCH stats-methods-step1.R
R CMD BATCH stats-methods-step2.R
```

The two additional scripts compute the practices relating to method signatures and method bodies. Now copy the output `icon-data-<PRACTICE>.csv` files to the `sparklines` directory where we generate the actual histograms:

```
cd ~/SummarizationArtifacts/sparklines/
cp ../presentation-except-formatting/icon-data<PRACTICE>.csv .
cp ../formatting/icon-data<PRACTICE>.csv .
cp ../selection/icon-data<PRACTICE>.csv .
```

3.3 Generating the Histograms

The Latex files in the `sparklines` directory named `icon-<PRACTICE>.tex` generate the sparkline histograms. Each of these files takes the corresponding data file `icon-data-<PRACTICE>.csv` as the input to a Latex macro we wrote called `\picture`, which is defined in `icon.tex`. To generate all the sparkline histograms, run following script to generate histograms in PDF and SVG formats:

```
./generate-icons.bsh
```