Code Story: Information Tracking for Program Comprehension

Felix Grund and Nick Bradley
Department of Computer Science
University of British Colombia
Vancouver, BC, Canada
ataraxie,nbrad11@cs.ubc.ca

ABSTRACT

Program comprehension impacts code quality, velocity and efficiency. However, it is a difficult task when only using code; many of the design and implementation decisions are lost when transferring the information into code. Unfortunately, capturing this information can be challenging, especially when it requires significant changes in developer workflow. Further, if the information is captured it must be linked to the section of code that it impacts and made available to reviewers in a way that enhances comprenshion without being obtrusive.

In this paper we describe a simple method for caputring developer reasoning during coding tasks. Our implementation, called CodeStory, integrates transparently into a developer's workflow by hooking into the standard copy-paste operation. In this iteration, when developers copy text or code snippets from StackOverflow, CodeStory will scrape the page capturing additional context and including it with the pasted content in their code. This information will later be seen by other developers giving them a better understanding of the author's reasoning.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

A program is the result of decisions made based on information obtained from a variety of sources. Knowledge of these decisions can quickly evaporate if there is no process for capturing it. In practice, documenting knowlegde is often considered a resource-intensive process without tangible, short-term gains, so often it is skipped or performed inadaquately[1, 2]. However, this information is useful in program comprehension tasks. In particular, we found there is strong industry support in code review.

approach **introduce the information as a code story**
The contributions of this paper are as follows:

- A simple approach for capturing text-based collaborative reasoning among developers. This can include communications using email, instant messaging, websites and references.
- A prototype tool that implements this approach. It captures contextual information surrounding snippets copied from StackOverflow and stores it in a database using a Google Chrome extension. An Atom[1] package lets developers paste the snippet with a hyperlink to the code story.

Section 2 presents related work, ...

2. RELATED WORK

related-work.tex

3. APPROACH

approach.tex

4. EVALUATION

evaluation.tex

5. IMPLEMENTATION

The implementation of CodeStory consists of three parts that interact with each other. There is (1) a Chrome extension responsible for collecting meta-information from StackOverflow when snippets are copied, (2) a backend receiving these information and persisting them in a database and (3) an Atom package that extends pasted snippets with a link to a page showing these meta information. This section provides a brief description of each part.

5.1 Chrome Extension

The CodeStory Chrome extension mainly consists of a Content Script that is executed whenever a user visits a

[1]https://atom.io

Table 1: Fields collected as meta-information from StackOverflow upon copy

Field Name	Description
copiedFrom	Whether the snippet was copied from a question or an answer.
type	Whether the copied snipped is code-only.
originalSelection	Exact text that was selected in the browser upon copy.
questionTitle	Title of the question of the current page.
questionUrl	URL of the question (i.e. the URL of the current page).
questionContent	Full content of the question as text.
questionContentWithHtml	Full content of the question preserving the HTML markup.
questionVotes	Number of votes the question received.
answerUrl*	URL of the answer.
answerContent*	Full content of the answer as text.
answerContentWithHtml*	Full content of the answer preserving the HTML markup.
answerVotes*	Number of votes the answer received.
accepted*	Whether the answer was accepted.
accessTime	Timestamp when the page was accessed.
fullCodeSnippet	The full code snippet (only available if the snippet is part of a code snippet).

* Only available if copied from an answer.

StackOverflow page. An event listener for the browser's *copy event* is added and henceforth called whenever content from the page is copied to the clipboard.

The listener first collects the required meta-information from the current page an saves them in an *storyData* object. The fields that are collected are shown in Table 1. To be able to identify a *storyData* object, a unique ID is created with a hash based on the current timestamp and the selected text. The created hash and the *storyData* object are now sent to the CodeStory backend with a POST request. Measures to ensure acceptance of these cross-origin requests are taken on the backend and the Chrome extension may thus send simple Ajax requests.

The missing link is now the one to the Atom package: the pasted snippet must be somehow associated with its meta-information. For that purpose, the clipboard content is enriched with the created hash which is achieved using the JavaScript clipboard API[2]. The Atom package may then extract this hash from the clipboard content upon paste and proceed accordingly.

5.2 Backend

XXX

5.3 Atom Package

XXX

6. DISCUSSION

discussion.tex

7. FUTURE WORK

future-work.tex

8. CONCLUSION

conclusion.tex

9. REFERENCES

[2]https://developer.mozilla.org/en-US/docs/Web/API/ClipboardEvent

[1] O. Zimmermann, U. Zdun, T. Gschwind, and F. leymann, "Combining pattern languages and reusable architectural decision models into a comprehensive and comprehensible design method," in *Proceedings of the Seventh Working IEEE/IFIP Conference on Software Architecture (WICSA 2008)*, WICSA '08, (Washington, DC, USA), pp. 157–166, IEEE Computer Society, 2008.

[2] N. B. Harrison, P. Avgeriou, and U. Zdun, "Using patterns to capture architectural decisions," *IEEE Softw.*, vol. 24, pp. 38–45, July 2007.