Modelling Design Topics in Online Discussions

Calahan Janik-Jones,[1] Michalis Famelis[2]
[1] University of Toronto, [2] Université de Montréal

Motivations and Goals

When software designers are no longer aware of a system's design, it tends to erode.[1] Moreover, "design" is difficult to capture explicitly.[2]

Because developers rarely have access to the full state of a system's design, we aimed to develop a technique for **finding all design discussions related to a code snippet**, to help developers comprehend past decisions in the project.

Furthermore, this project attempts to address how design-relevant information can be **collected en masse** and **presented to designers** alongside existing code.

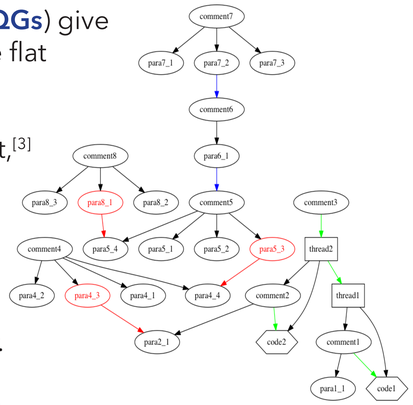
Fragment Quotation Graphs

Fragment Quotation Graphs (**FQGs**) give discourse structure to otherwise flat conversations.

Used in detecting disagreement,[3] topic segmentation,[4] and document summarization.[5]

Our FQG includes not only quotations, but mentions, code references, and comment order.

Treats snippets of **code artifacts** as a fully-fledged comment in the discussion.

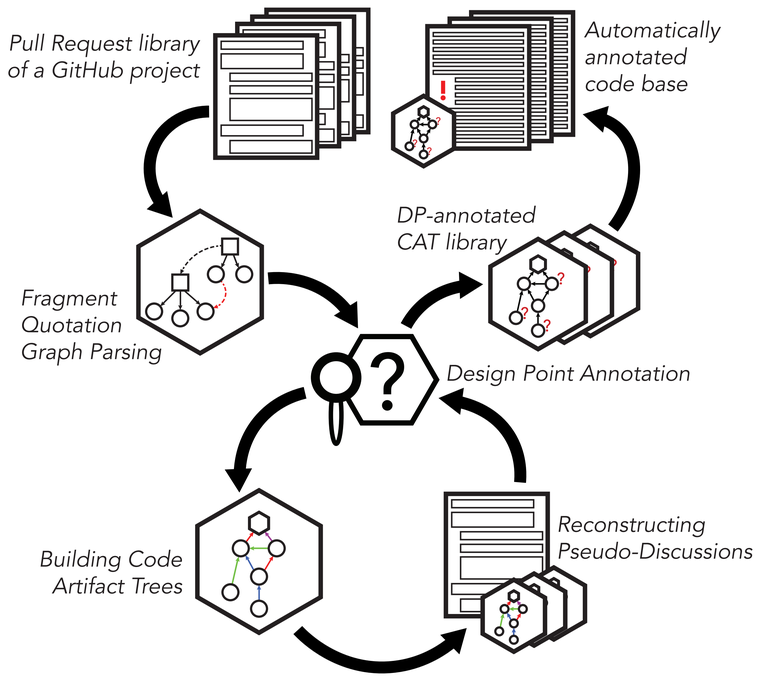


A subsection of an FQG including quotes (in red), @user mentions (in blue), and precedence (in green), visualized using GraphViz.

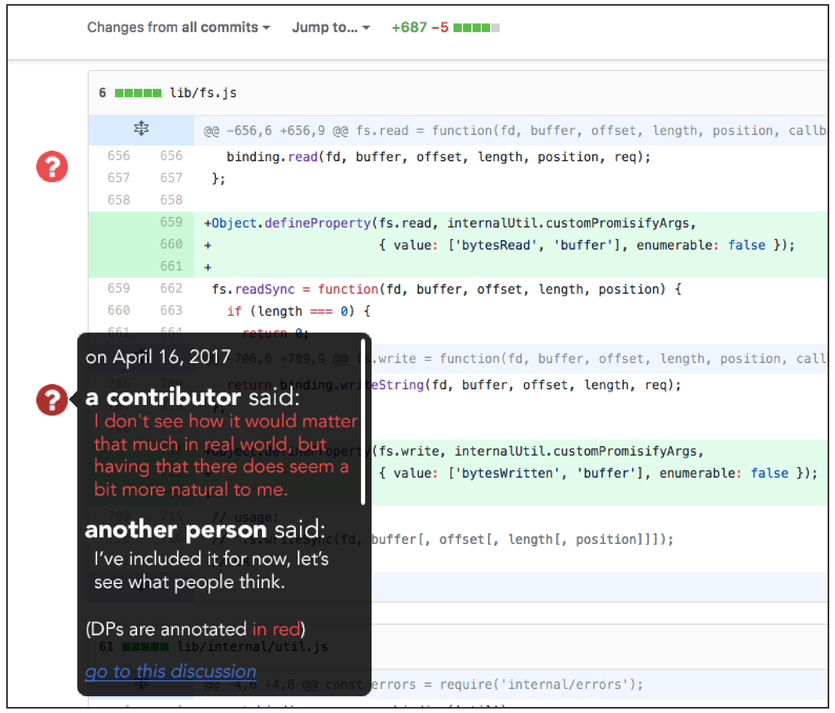
Code Artifact Trees

We transform the FQGs into CATs, which are snippets of the discussion directly addressing a code artifact.

These are directly linked to lines in a code artifact in the GitHub project. This will allow us to directly annotate while viewing code.



Our process for generating automatic code annotations. Code is dually annotated for design-relevant information, by analyzing both the original pull request and the reconstructed pseudo-discussions.



An early mockup for an annotation overlay on GitHub, which would summarize and link to design points in discussions about blocks of code, as well as note DPs identified from that discussion.

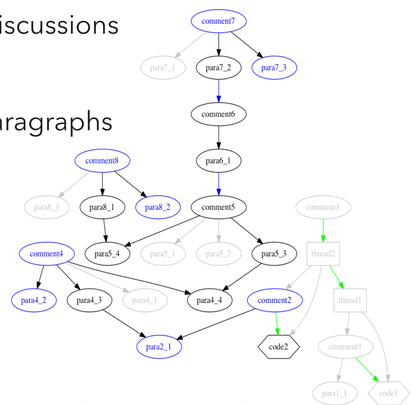
Design Discussion Annotation

From the FQG we extract sub-discussions containing design information.

We use a classifier to identify paragraphs that call on developers to make a decision.

With the FQG including code artifacts, connects design discussions to code.

Identifies information relevant in **design choices** that can **combine with FQG model**.



Annotating the FQG for design-relevant (in blue) lets us cut down to the information most relevant to tracking decisions made in the software design.

Pseudo-Discussion Parsing

We can take a forest of code artifact trees about a particular code artifact and construct a pseudo-discussion of all conversations about the artifact throughout the project.

RST-parsing and design-annotation this document gives us another layer of discourse information in a new context.

Context of **one change** with **many artifacts**
(original discussion annotation)
vs.
Context of **all changes** many for **one artifact**
(pseudo-discussion annotation)

Both give interesting information about the software design.

Code Annotation and Future Work

Currently the software scrapes full 10,000+ PR GitHub projects and construct libraries of

- Improve identification of design-relevant information
- Use topic labelling tools, such as ConVis,[4] for further annotation
- Combine with extractive summarization to communicate design
- Create an IDE or browser plugin

References

[1] Robillard, M. P., & Medvidović, N. (2016, May). Disseminating architectural knowledge on open-source projects: a case study of the book architecture of open-source applications. In Proceedings of the 38th International Conference on Software Engineering (pp. 476-487). ACM.

[2] Van Gurp, J., & Bosch, J. (2002). Design erosion: problems and causes. Journal of systems and software, 61(2), 105-119.

[3] Allen, K., Carenini, G., & Ng, R. (2014). Detecting disagreement in conversations using pseudo-monologic rhetorical structure. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1169-1180).

[4] Joty, S., Carenini, G., & Ng, R. T. (2013). Topic segmentation and labeling in asynchronous conversations. Journal of Artificial Intelligence Research, 47, 521-573.

[5] Hu, M., Sun, A., & Lim, E. P. (2008, July). Comments-oriented document summarization: understanding documents with readers' feedback. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (pp. 291-298). ACM.