Modelling Design Topics in Online Discussions

Calahan Janik-Jones,¹ Michalis Famelis²
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.

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.

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.

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.

References