Summary

- This paper's key contribution is about library or domain specific extensions to the IDE. While the idea for domain-specific extensions has been considered in an ad-hoc manner before (see examples below), this is the first formalization of the idea through a research publication as far as we know, and implements an Eclipse plugin to allow active code completion for arbitrary (i.e. user-defined) types.
    - We interpreted the contribution of the paper to be IDE-independent and extensible code completions for specific domains.
    - Examples of similar work in industry:
        - Color palettes in ReSharper
            - http://www.jetbrains.com/resharper/webhelp/Coding_Assistance__Color_Assistance.html
        - IntelliJ Regex Plugin
            - http://myregexp.com/ideaPlugin.html
            - http://blogs.jetbrains.com/idea/tag/regexp/

High-level Comments

- General Strengths
    - Large upfront survey results are very valuable (valuable for many different applications beyond code completion)
        - Very robust, good gauge of interest
    - The contribution itself
        - Formalizes and generalizes something already being used ad-hoc in industry
            - This is valuable, because it has not yet been addressed in research
        - Proposes (and includes example via an Eclipse plugin) of how to allow libraries to define their own autocomplete 'palettes'
- General Weaknesses
    - References to related work in code completion & extensions to the IDE
        - Most significant problem we found
    - We missed a quantitative analysis and interpretation for the results presented in the final user study
    - Transitions from design criteria to design of the tool seemed abrupt
        - How did the design criteria affect the design of your tool? We could draw some intuitive conclusions, but your reasoning was not explicitly explained.
    - Specific limitations of the tool (less important)
        - Are HTML pages really the most effective way to implement these code assistant tools?
            - Won't this make it difficult to match the look & feel of specific IDEs?
            - Since jQuery is loaded for each palette, doesn't this incur some unnecessary overhead?

Comments by Section

- Introduction
  - Clarify your contribution
    - We know that very similar code assistant tools exist in practice (e.g. Resharper color palettes and IntellIJ regex completion plugin). We suggest that you perhaps compare and contrast Graphite with these tools.
- Online Survey Results
  - Very impressive size for the survey, the results are interesting
- Design Criteria
  - It was unclear to us how these categories were selected from the survey results. We assume that it was done manually by the authors, but could/should there be more information about how these criteria were extracted from the results?
- Final User Study (Sections 6 & 7)
  - For this section, we wanted more details of tasks that users were asked to perform than the specific problems each user had. It would be good to release the task materials.
    - We saw an analysis of the feedback received on the tool, rather than information about the effect of the tool on the ease of completing the tasks assigned (and specific information about these tasks themselves)
  - What is the statistical significance of these results?
    - You say near the beginning of the section that the sample size is too small to get any generalizable results, but present all results in terms of percentages and try to discuss trends between the users.
  - Remove percentages from the user study
    - For the sake of consistency, and because of the size of the user study, it may be better to use raw numbers instead of percentages.
- Related Work
  - Biggest problem with this paper, and something that easily could be improved before ICSE camera-ready deadline is the related work section. Specifically, there has been closely related work in code completion:
    - Jungloid (from Ras Bodik's group at Berkeley) and a whole line of follow up papers by others addressed the issue of helping the programmers learn a library API by mining and recommending sample usage of the API
    - Work on code completion from Rob Miller's group at MIT
      - For example, *Code Completion from Abbreviated Input* (http://dspace.mit.edu/handle/1721.1/59377)
    - Other related links:
      - http://vimeo.com/36579366
      - http://code-recommenders.blogspot.com/
    - Similar examples for regex & color completion:
      - Color palettes in ReSharper
        - http://www.jetbrains.com/resharper/webhelp/Coding_Assistance__Color_Assistance.html
      - IntelliJ Regex Plugin
        - http://myregexp.com/ideaPlugin.html
        - http://blogs.jetbrains.com/idea/tag/regexp/