

Integration of Real-Time Collaborative Coding with Git for Productivity Gains

Braden D. Licastro
Department of Computer Science
Allegheny College
licastb@allegheny.edu
<http://www.fullforceapps.com/>

February 21, 2013

Abstract

This paper describes a method of coding to be researched that has the potential to greatly improve efficiency of program development. By combining the effectiveness and flexibility of collaborative coding with a version control system it may be possible to reduce time wasted during the software development cycle and shorten turn around times.

1 Introduction

Software development is full of pitfalls that take an innumerable amount of time every year to overcome, especially when working with a group of developers. One major stumbling point every seasoned programmer will experience arises when using a revision control system. When projects are branched for feature work and merged at a later time collisions, otherwise known as conflicts will more than likely occur. The current method of resolving these conflicts is arduous at best, and can even lead to a code freeze until a solution is found. By integrating a version control system with a method of collaborative real-time coding it should be possible to greatly reduce, or even eliminate the chance of conflicts when a project is merged back to the master branch without losing the benefits of a source manages project.

Current implementations of revision control systems are designed to be as intuitive and straightforward as possible, but for the most part remain weak in their handling of errors. Frequent problems arise when two branches of a project must be merged back into one and the code on one conflicts with code in the other. As of the writing of this paper, there is no standardized method of handling these situations. Many times it is required that the developer go back and manually edit the code thus resolving any conflicts. In some instances this is a trivial task, but others it is tedious and time consuming. With real-time collaboration it is possible to have many developers actively working on the same code-base without the necessity of a code merge that may or may not work out of the box. This setup would allow all contributors to see the code as it currently stands on other computers and work together to create code that is conflict free on the fly. This gain in efficiency could add up to be quite considerable, and different implementations may produce significantly beneficial results.

2 Related Work

The topic of collaborative coding is a rather new research topic. Much of the current research resolves around the implementation of an effective environment. One research paper describes a web-based Java development environment that supports collaboration between programmers

names Collabode.[2] The system analyzed is web based and has no code management integrated. The implementation I have proposed would be non web based in implementation, and it would have integrated revision control. This paper also talks about the shortcomings of such technologies, while the proposed research would focus primarily on the efficiency gains of said implementation. Similar to this paper, the same authors published another article that discusses possible workload reductions by using the Collabode system to outsource small amounts of coding. [1]

Summarize the previously published papers and books that are related to your proposed research. Whenever possible, you should compare and contrast your approach with the ones that have been discussed in the past. As you describe your papers, please make sure that you cite them properly

3 Method of Approach

Use technical diagrams, equations, algorithms, and paragraphs of text to describe the research that you intend to complete.

4 Evaluation Strategy

Explain what steps you will take to evaluate your proposed method. If you intend to conduct experiments, then you must clearly define your evaluation metrics.

5 Research Schedule

Identify the main phases and tasks of your research project and set deadlines for when you will be able to complete each of these items.

6 Conclusion

Provide a summary of your proposed research and suggest the impact that it may have on the discipline of computer science. If possible, you may also suggest some areas for future research.

As mentioned in Section 3 ...

References

- [1] Max Goldman, Greg Little, and Robert C. Miller. Collabode: collaborative coding in the browser. In *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE '11, pages 65–68, New York, NY, USA, 2011. ACM.
- [2] Max Goldman, Greg Little, and Robert C. Miller. Real-time collaborative coding in a web ide. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 155–164, New York, NY, USA, 2011. ACM.